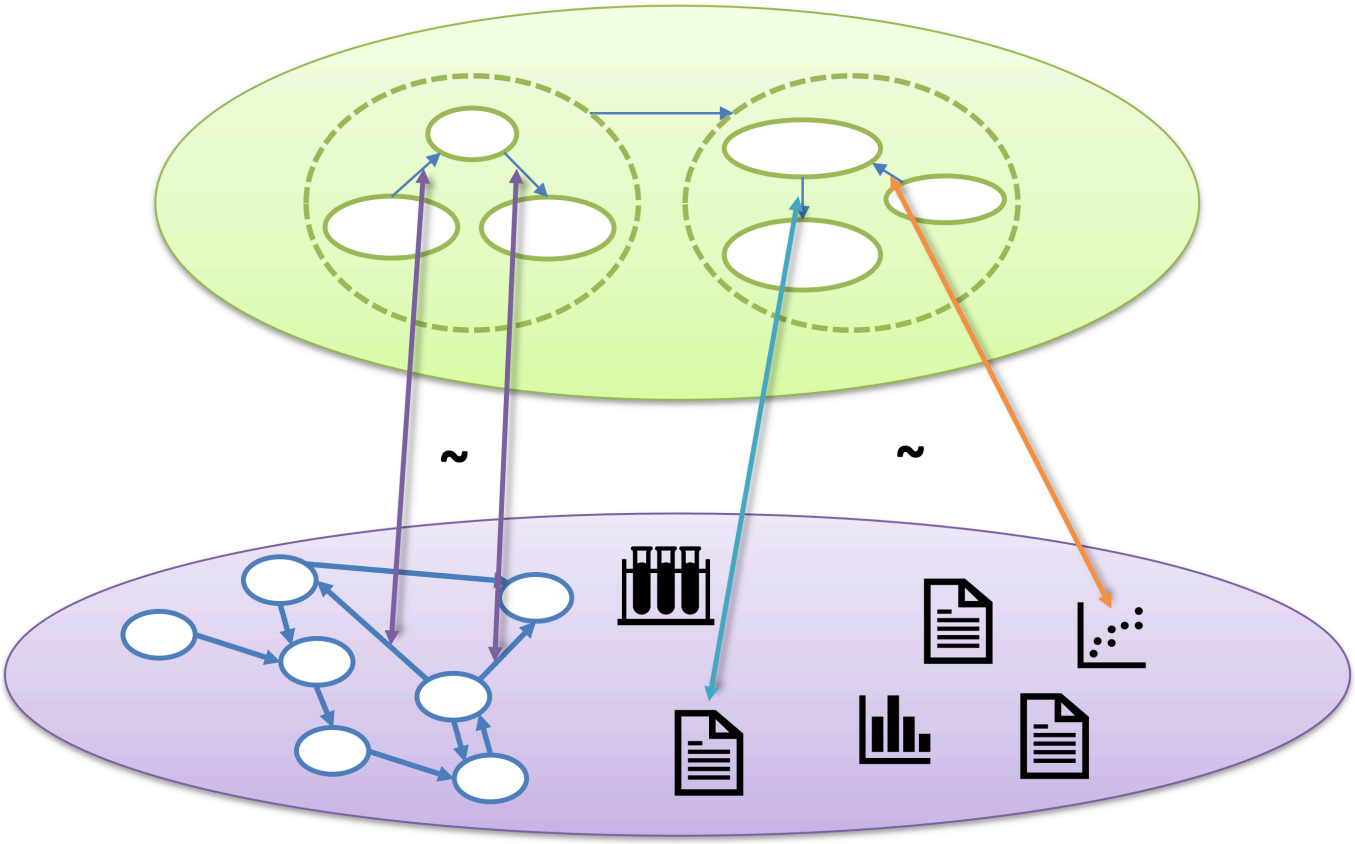
Technische Universität Braunschweig

ifis
Institut für Informationssysteme
Technische Universität Braunschweig



Narrative Information Access – A new Paradigm for Digital Libraries

Dissertation

Hermann Kroll

January 17, 2024

Narrative Information Access – A new Paradigm for Digital Libraries

Von der
Carl-Friedrich-Gauß-Fakultät
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation
(kumulative Arbeit)

von
Hermann Kroll
geboren am 7. August 1993
in Celle

Eingereicht am:	21. September 2023
Disputation am:	13. Dezember 2023
1. Referent:	Prof. Dr. Wolf-Tilo Balke
2. Referent:	Prof. Dr. Ralf Schenkel

2023

”Don’t Panic.”

Douglas Adams, The Hitchhiker’s Guide to the Galaxy

"Someone once told me that time was a predator that stalked us all our lives. I rather believe that time is a companion who goes with us on the journey and reminds us to cherish every moment, because it will never come again."

Jean-Luc Picard, Star Trek

Acknowledgments

Lieber Tilo, danke, dass du mich betreut hast und mein Mentor geworden bist. Du hast mich weiter gepusht, als ich jemals dachte, gehen zu können. Ich bin stolz auf das, was wir gemeinsam erreichen konnten. Ohne deine kritische Art und deine ehrlichen Worte wären wir heute nicht da, wo wir angekommen sind. Ich danke dir, dass du mich die wissenschaftliche Arbeit gelehrt hast.

Lieber Ralf, ich erinnere mich noch an das erste Treffen 2018. Du fragtest mich beim Abendessen, was Narrative mit Informationssystemen zu tun haben sollten. Ich hoffe, in dieser Arbeit findest du überzeugende Antworten. Ich danke dir für deine Begleitung während meiner Promotion und im Rahmen des FID. Du hast mir viele wertvolle Ratschläge gegeben und mir damit in meiner Forschung sehr geholfen. Danke, dass du mein Zweitprüfer bist.

Prof. Lipeck, Ihnen möchte ich ebenfalls danken. Ohne Ihre Unterstützung und vor allem ohne Ihre Betreuung im Rahmen meiner Masterarbeit wäre ich niemals so weit gekommen. Sie haben mich gefordert und mich die konzeptionelle Arbeit gelehrt.

Mein weiterer Dank gilt meinen Kollegen, sowohl den aktuellen als auch den ehemaligen. Danken möchte ich Usman, Mayukh, Janus und José. Weiterhin möchte ich dem AI-Guy Niklas, dem Narrative Boy Denis, dem LaTeX-Guru und DL-Co-Autoren Florian, meinem Datalog-Buddy Stephan sowie dem Kritiker Janni danken. Ihr alle habt maßgeblich zu dieser Arbeit beigetragen (ohne Florian hätte sie auch schlicht nicht kompiliert). Danken möchte ich auch Thomas. Du und deine Betreuung unserer Infrastruktur haben die Umsetzung und Implementierung dieser Arbeit ermöglicht.

Ebenfalls danke ich meinen Studierenden, die an meiner Forschung mitgewirkt und sie vorangetrieben haben – ohne euch wäre es nicht möglich gewesen, so weit zu kommen und vor allem den Narrativen Service umzusetzen. Namentlich hervorheben möchte ich dabei Johannes, Jan, Morris, Matze und Pascal.

Ein besonderer Dank gilt den Kolleginnin und Kollegen anderer wissenschaftlicher Einrichtungen: allen vorweg dir, Christin – deine kritsche Art und dein detailliertes Feedback waren prägend und lehrreich für meine wissenschaftliche Arbeit. Du hast mir gezeigt, wie eine präzise Argumentation geführt werden muss. Danke auch dir, Timo, für dein umfangreiches Feedback im Bereich des Information Retrieval. Thank you Mirjam for the nice collaboration in the area of digital libraries.

Danken möchte ich auch unseren Projektpartnern und -innen – vor allem Christina und Stefan. Ihr habt durch euer wertvolles Feedback und zahlreiche Erklärungen mein Wissen in der Pharmazie vertieft. Ohne euch wäre mir diese Domäne verschlossen geblieben. Ihr habt beide maßgeblich zu meiner Forschung und dem Narrativen Service

beigetragen. Weiterhin danken möchte ich auch den anderen aktuellen sowie ehemaligen FID Mitgliedern: Benjamin, Konrad, Robert, Denitsa, Kristof und Frau Stump.

Ein weiterer Gruß geht an meine Projektpartner und -innen, mit denen ich zusammen arbeiten durfte. Danke an den FID Politikwissenschaft für die Bereitstellung der in dieser Arbeit analysierten Daten sowie den anregenden Austausch und das hilfreiche Feedback. Danke auch an Julian für die intensiven und tiefgreifenden Gespräche über einen narrativen Informationszugriff in der Politikwissenschaft. Katharina und Lisa möchte ich auch danken – die sehr angenehme Zusammenarbeit im Rahmen des Long-COVID-Projekts hat die Relevanz meiner Forschung für mich bekräftigt.

Ein großer und außerordentlicher Dank gilt R. – Regine, du hast mein englisches Sprachverständnis sowie den Einsatz dieser Sprache entscheidend verbessert. Du hast ebenfalls meine Allgemeinbildung, insbesondere Kenntnisse in Film und Literatur, erweitert. Sehr gern erinnere ich mich an viele aufmunternde und unterhaltsame Kaffeepausen.

Danken möchte ich meinen Eltern. Ihr habt mich mein Leben lang begleitet und mir alles ermöglicht. Ohne eure Unterstützung wäre diese Reise nicht möglich gewesen.

Ein großer Dank gilt meinen Freunden (Olli, Jenny, Franzi, Leandra, dem AoE-Team, den Hots-People und vielen weiteren), die immer zu mir gehalten haben und mich mental unterstüzt haben. Besonders die Spieleabende am Freitag (danke Dominic) waren mir immer ein wöchentliches Highlight und eine willkommene Ablenkung.

Insbesondere möchte ich Jenny danken. Ohne deine Unterstützung wäre diese Arbeit nicht möglich gewesen. Du bist immer für mich da und trägst viel Freude in mein Leben.

Diese Forschung wurde gefördert durch die Deutsche Forschungsgemeinschaft (DFG): PubPharm – Fachinformationsdienst Pharmazie (Gepris 267140244). Vielen Dank.

Contributing Publications

This thesis is based on the following publications:

1. **Hermann Kroll**, Jan-Christoph Kalo, Denis Nagel, Stephan Mennicke, and Wolf-Tilo Balke. "Context-Compatible Information Fusion for Scientific Knowledge Graphs". International Conference on Theory and Practice of Digital Libraries (TPDL), Lyon, France, 2020, Springer. DOI: https://doi.org/10.1007/978-3-030-54956-5_3

2. **Hermann Kroll**, Jan Pirklbauer, and Wolf-Tilo Balke. "A Toolbox for the Nearly-Unsupervised Construction of Digital Library Knowledge Graphs". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Urbana-Champaign, IL, USA, 2021, IEEE. DOI: https://doi.org/10.1109/JCDL52503.2021.00014

3. **Hermann Kroll**, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. "Narrative Query Graphs for Entity-Interaction-Aware Document Retrieval". International Conference on Asian Digital Libraries (ICADL), Online, 2021, Springer. DOI: https://doi.org/10.1007/978-3-030-91669-5_7

4. **Hermann Kroll**, Florian Plötzky, Jan Pirklbauer, and Wolf-Tilo Balke. "What a Publication Tells You – Benefits of Narrative Information Access in Digital Libraries". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Cologne, Germany, 2022, ACM. DOI: https://doi.org/10.1145/3529372.3530928

5. **Hermann Kroll**, Jan Pirklbauer, Florian Plötzky, and Wolf-Tilo Balke. "A Library Perspective on Nearly-Unsupervised Information Extraction Workflows in Digital Libraries". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Cologne, Germany, 2022, ACM. DOI: https://doi.org/10.1145/3529372.3530924

6. **Hermann Kroll**, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. "A discovery system for narrative query graphs: entity-interaction-aware document retrieval". International Journal on Digital Libraries (IJDL) 2023. DOI: https://doi.org/10.1007/s00799-023-00356-3

7. **Hermann Kroll**, Jan Pirklbauer, Florian Plötzky, and Wolf-Tilo Balke. "A detailed library perspective on nearly unsupervised information extraction workflows in digital libraries". International Journal on Digital Libraries (IJDL) 2023. DOI: https://doi.org/10.1007/s00799-023-00368-z

8. **Hermann Kroll**, Christin Katharina Kreutz, Pascal Sackhoff, and Wolf-Tilo Balke. "Enriching Simple Keyword Queries for Domain-Aware Narrative Retrieval". ACM/IEEE Joint Conference on Digital Libraries (JCDL) Santa Fe, NM, USA, 2023, IEEE. DOI: https://doi.org/10.1109/JCDL57899.2023.00029 arXiv: https://doi.org/10.48550/arXiv.2304.07604

Abstract

Digital libraries usually allow users to access their collections through keyword-based retrieval. While such access paths come with moderate implementation and maintenance costs, they also come with limited expressiveness. This has two reasons: 1) expression of relations between keywords is challenging, and 2) exploratory search with variables is usually not supported at all. Inspired by the way humans exchange knowledge through oral or written narratives, this thesis proposes narrative information access to tackle those limitations. The central idea is that users formulate information needs as short narratives of interest, basically a graph pattern with relevant concepts and their interactions. Those patterns are then bound against a digital library's content, e.g., its document collection. In contrast to querying knowledge bases, narrative information access enforces a context-compatible information fusion to ensure valid results. This fusion only combines pieces of information whose validity refers to the same or similar settings. In this thesis, narrative information access has been realized in the use case of the pharmaceutical domain. Moreover, we propose, implement, and evaluate practical nearly-unsupervised information extraction workflows, novel implicit context models, and a full-fledged discovery system for narrative information access.

Zusammenfassung

Digitale Bibliotheken ermöglichen Nutzenden den Zugriff auf ihre Sammlungen in der Regel mittels schlüsselwortbasierter Anfragen. Solche Zugriffspfade sind mit moderaten Implementierungs- und Wartungskosten verbunden, haben aber auch eine begrenzte Aussagekraft. Dies hat zwei Gründe: 1) Der Ausdruck von Beziehungen zwischen Schlüsselwörtern ist schwierig, und 2) die explorative Suche mit Variablen wird in der Regel nicht unterstützt. Inspiriert durch die Art und Weise wie Menschen Wissen durch mündliche oder schriftliche Narrative austauschen, wird in dieser Arbeit ein narrativer Informationszugriff vorgeschlagen, um die obigen Einschränkungen zu überwinden. Die zentrale Idee besteht darin, dass Nutzende ihren Informationsbedarf in Form eines kurzen Narrativs formulieren, also als Graphmuster bestehend aus relevanten Konzepten und deren Interaktionen. Diese Muster werden dann an die Inhalte einer digitalen Bibliothek gebunden, z. B. an ihre Dokumentensammlung. Im Gegensatz zu Anfragen an Wissensbasen (Knowledge Bases) fordert der narrative Informationszugriff eine kontextkompatible Informationsfusion, um valide Ergebnisse sicherzustellen. Diese Fusion kombiniert nur Informationen, die unter gemeinsamen oder ähnlichen Bedingungen gültig sind. In dieser Arbeit wurde der narrative Informationszugriff im Anwendungsfall der pharmazeutischen Domäne realisiert. Darüber hinaus werden praktische, nahezu unüberwachte Informationsextraktionsworkflows, neuartige implizite Kontextmodelle und ein vollwertiges Discovery System für den narrativen Informationszugriff vorgeschlagen, implementiert und evaluiert.

Contents

1. Introduction

"Science […] is made up of mistakes, but they are mistakes which it is useful to make, because they lead little by little to the truth."

Jules Verne, A Journey to the Center of the Earth

Digital libraries are content providers for scientific knowledge: They maintain and curate extensive collections of data, e.g., images, videos, research data, and textual content such as articles and books. Today, keyword-based access paths are an established way to allow users to retrieve information from those collections. First, implementing such access comes with moderate costs for digital libraries; see [51] for indexing costs. Second, keyword queries are accepted by users because they are rather easy to use. In this way users can retrieve the library's content and determine what is actually *told* by the data.

1.1. Focus of this Thesis

Humans exchange and share knowledge following a narrative oral tradition, i.e., they *tell* stories and have structured debates and conversations [45]. Oral presentations are made persistent by writing up stories, comments, and discussions. During that writing process, the central way to encode knowledge is still to tell a story: A narrator relates what was observed, draws complex conclusions, and explains how they were derived from basic claims. We understand that process as *composing a narrative*, i.e., the narrator tells a pattern bound to real-world concepts to form rich lines of arguments [16]. We understand the process of *binding to real-world concepts* as connecting the narrative pattern to knowledge from the real-world, e.g., referring to concrete actors/entities, mechanisms, etc. Readers can follow these lines of arguments and may find the shared knowledge plausible.

We understand *narratives* as being *logical overlays* on top of *knowledge repositories* [35]; see Figure 1.1 for a visualization. In brief, a narrative conveys knowledge by combining *pieces* from different sources, e.g., knowledge graphs, databases, textual sources, and data sets. We then proposed narrative information systems [33] that focus on narratives as their first-class citizens. Based on this idea, this thesis proposes **narrative information access** as a new paradigm for digital libraries. Narrative information access turns the previous process around: A user formulates a narrative pattern, and a digital library has to *ground* the pattern by its contained knowledge, i.e., it must find evidence for the narrative's parts.

In detail, users formulate their information needs as **narrative patterns**, involving relevant concepts and their interactions. For instance, a user might search for *drugs* which *treat diabetes mellitus* in *adults* and which are *administered* as an *injection*. A system must then **bind** each of its parts against its underlying repository to find evidence. In our example, it must find evidence for the drug-disease treatment, the drug's applicability to adults, and the drug's administration as an injection. In addition to finding evidence, the system must also ensure that those bindings, i.e., the connections between the narrative pattern and the repository's knowledge, are **context-compatible**: Here, only drug administrations

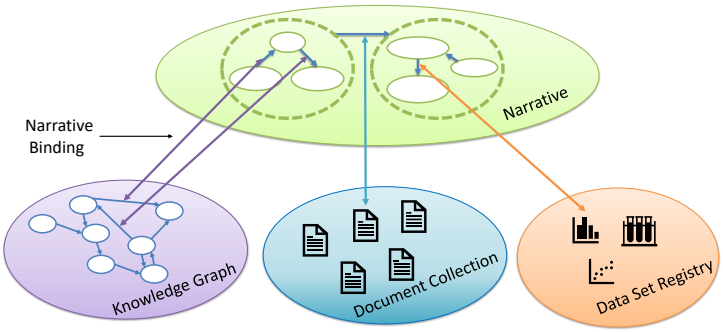


Figure 1.1.: Modeling narrative structures as logical overlays on top of knowledge repositories. This figure has been taken from our work [35].

are valid if, and only if, the drug can safely be used to treat diabetes in adults in that corresponding dosage form. In other words, a context-compatible information fusion ensures that the fused pieces belong together by matching their contexts and forming valid results.

We propose narrative information access as a new paradigm for digital libraries. However, its implementation requires a query paradigm, an efficient implementation for fast online retrieval, and suitable user interfaces so that real users accept and appreciate it. This thesis contributes answers by proposing practical extraction workflows, a novel implicit context model, and a complete retrieval system for narrative information access in the example of the pharmaceutical domain.

1.2. Contributions

This thesis' contributions were published in eight peer-reviewed papers in the field of digital libraries (4x Joint Conference on Digital Libraries, 1x Theory and Practices of Digital Libraries, and 2x International Journal on Digital Libraries, and 1x International Conference on Asia-Pacific Digital Libraries). Within the scope of this thesis, we focus on finding evidence in textual sources for narrative information access because they are common in digital libraries. The National Library of Medicine, for instance, curates about 36 million documents in their MEDLINE collection[1]. Binding narrative patterns against a library's texts requires to identify concepts and relations that are mentioned in the user's given narrative. We therefore proposed the following procedure to realize such a graph-based retrieval: First, we transformed texts into a graph representation. Then, we applied a graph pattern matching paradigm to compare the queried pattern and collection texts. Implementing a suitable extraction workflow, however, can come with high application costs and may be challenging for a digital library in practice because typical extraction workflows rely on the acquisition of training data; see Chapter 2 for a detailed discussion.

Information Extraction Workflows [36, 37, 39]: The first part of this thesis thus contributes practical, so-called **nearly-unsupervised**, workflows which bypass training data in the extraction phase, retain canonicalized triple-shaped representation, and come with acceptable costs, but at a lower quality compared to supervised methods.

[1] `https://www.nlm.nih.gov/medline/medline_overview.html`, Last accessed 10.09.2023

We demonstrate how these workflows can be deployed to transform texts into graph representations, at the example of the pharmaceutical domain. Next, narrative information access forces a context-compatible information fusion to ensure valid results in the end. In other words, the whole narrative pattern must be valid within one context. The related work suggests modeling context conditions explicitly, e.g., by harvesting n-ary relations [14], attaching qualifiers [23, 69], or using logic to specify rules on under which conditions knowledge can safely be fused [54]. In brief, designing contexts can be challenging for a digital library because every relevant context condition must be known in advance before harvesting statements from text. And even if known, extracting those conditions is challenging, e.g., higher-ary relations need to be defined and training data must be provided [14]. Explicitly modeling contexts is challenging and may become close to impossible in some domains. We refer the reader to Chapter 3 for a detailed discussion.

Context-Compatible Information Fusion [40, 43]: This thesis discusses why a digital library should retain contexts of statements. A context-compatible information fusion then ensures validity across result sets. Moreover, this thesis contributes a novel, practical **implicit context model**, for which no more is required than keeping a statement's source. In addition, we propose similarity measures based on texts or their metadata that allow to combine statements from different sources.

Our proposed information extraction workflows and the implicit context model allow a digital library to transform its textual collection into a graph representation and retain contexts for every statement. With that, we can implement narrative information access. The third part of this thesis tackles the conceptualization, a possible retrieval system design, its full-fledged implementation, and user evaluations. In particular, the query processing must be fast enough to allow online retrieval, and the user interface must be suitable and intuitive, so that users accept the system and gain benefits. For instance, variables in queries require a new way to visualize result lists for users. Moreover, the retrieval must be effective in the end, in terms of precise and exploratory searches.

Narrative Information Access [38, 41, 42, 43]: This thesis contributes narrative information access as a new paradigm for digital libraries. Moreover, the thesis proposes a possible system design and demonstrates its full-fledged implementation (service and user interface) involving query translation, retrieval, storage, indexing, and visualization. Our system has been implemented for the pharmaceutical domain (`www.narrative.pubpharm.de`). Beyond suitable user interfaces, we also proposed an effective method to deduce narrative patterns from keywords.

The related work for this thesis can roughly be structured into three categories: 1) knowledge representation and information extraction methods, 2) knowledge contextualization, and 3) information retrieval, involving graph-based retrieval, question answering, and keyword-based searches on structured data. The related work is distributed across the subsequent chapters. The thesis is structured as follows: Chapter 2 introduces nearly-unsupervised extraction workflows. Chapter 3 proposes an implicit context model and a context-compatible information fusion. Chapter 4 demonstrates the benefits and the implementation of narrative information access. We conclude our findings in Chapter 5.

2. Information Extraction Workflows

Information extraction has been a long-standing task in the field of Natural Language Processing (NLP). Its goal is to extract *structured* information from *unstructured* information sources, e.g., texts, images, and tables. Here, we focus on extracting structured statements from natural language texts. Digital libraries maintain collections of documents written in natural language. Some projects have already demonstrated the benefits of transforming a library's text collection into a structured representation, e.g., SemMedDB harvested from the MEDLINE collection [30] has been used for literature-based discovery [25] or predicting drug-drug interactions through semantic patterns [76].

One way to encode the extracted *knowledge* is to represent it with the Resource Description Framework (RDF) [53]. RDF proposes to store knowledge in the form of **statements**, also called facts, i.e., subject-predicate-object-shaped triples. For instance, (*Metformin*, *treats*, *diabetes mellitus*) encodes that *Metformin treats diabetes mellitus*. In RDF, subjects, predicates, and objects are called resources with unique resource identifiers (URI). In practice, those resources are also called **concepts**, a term we will use in this thesis. A predicate then puts two concepts into relation – forming a statement. A set of statements is called a **knowledge graph** (sometimes also referred to as a knowledge base). Knowledge graphs can be manually crafted in a collaborative fashion (e.g., Wikidata [69]), be harvested from text (e.g., SemMedDB [30]), or be extracted from semi-structured data (e.g., DBpedia [4]). The advantage of knowledge graphs is that they offer a canonicalized representation of knowledge, i.e., precise concepts and their relations. This representation then allows asking queries or perform reasoning with query languages like SPARQL.

2.1. Related Work

A comprehensive overview of the creation and curation of knowledge bases, especially methods to extract them from texts, can be found in [72]. The first step is usally to recognize named entities/concepts in texts, i.e., text spans that describe some relevant *thing*. Those entities/concepts can then be linked to knowledge bases. In this way, a text span is assigned to a precise concept. While this thesis implements dictionary-based concept linking, entity/concept recognition and linking is not our focus. Our focus lies on the statement extraction. Basically, the field can be structured into two main categories: 1) **Relation extraction** defines the set of known relations a-priori, and 2) **open information extraction** (Open IE) extracts based on the grammatical structure of sentences without knowing a schema with relations in advance. In the following, we quickly summarize the main research directions.

2.1.1. Relation Extraction

The relation extraction task aims to extract relations between two or more concepts from a text; see [72] for a comprehensive overview of methods. For relation extraction, the set of relations must be known a-priori. Usually supervised machine learning methods are used to gain the best possible accuracy. In the past, techniques like Support Vector Machines,

Random Forests, or neural networks were common for that task. The development of language models, such as BERT [11] or BioBERT [47], has led to improvements in terms of extraction accuracy. Language models are pre-trained on large text corpora and fine-tuned for a specific relation extraction task, e.g., extracting relations between drugs and diseases. The fine-tuning requires training data, i.e., labeled sentences. While different learning architectures mainly focus on improving accuracy, other methods focus on minimizing the amount of required training data: For instance, distantly-supervised relation extraction methods like Snorkel [60] create noisy labels for sentences by retrieving possible relation labels between concepts from existing knowledge bases. As opposed to creating extensive training data, few-shot relation extraction requires only *a few* training examples, and zero-shot requires no examples at all [49]. However, few/zero-shot extraction requires a suitable prompting strategy, i.e., an input that forces the language model to predict a relation with high accuracy. While language models have shown promising results on those tasks, the overall accuracy might strongly depend on selected prompts, and may in some cases not surpass random guessing [50].

Nevertheless, relation extraction, regardless of a specific subtask, still requires domain knowledge about what should be extracted from the texts. Gathering examples, knowledge bases, or suitable prompts can be challenging. In a practical digital library setting, they might simply not be available and too cost-intensive to acquire. In this thesis, we focus on open, unsupervised methods to 1) bypass the need for any training data and 2) explore what is being told in a digital library collection without knowing relations in advance.

2.1.2. Open Information Extraction

Open IE is the task of transforming a text into structured statements without knowing relations in advance; see [72] for a comprehensive overview of methods. Extractions are typically made based on the grammatical structure of a sentence. Usually, rules regarding when and what to extract, are hand-crafted or deduced by learning them from examples. Regardless of the method, Open IE can be applied directly and does not require domain-specific fine-tuning. However, extractions are not canonicalized, i.e., noun phrases (subjects/objects) are not linked/normalized to knowledge graphs/ontologies, and verb phrases are not linked to predicates. In brief, noun phrases might describe the same concept in different ways (synonyms). They might also be homonyms, or even contain multiple concepts, e.g., a single noun phrase describing a time and a location. For narrative information access it would be beneficial to extract precise concepts and relations. Users can then search for a specific concept/relation and get relevant text spans.

Much work has been done on evaluating and improving the Open IE task: Angeli et al. investigated how linguistic structure can be leveraged for open domain IE [3]. Other works have focused on the evaluation of Open IE on scientific texts [20], on crowd-sourced benchmarks [7], in a multilingual setting [31], or when performing multi-faceted fact-based extractions [17]. However, we focus on practical workflows with canonization.

Another direction of works aims to canonicalize the Open IE extractions, i.e., resolve synonymous noun phrases and verb phrases. CESI is a method that clusters those phrases to obtain a canonicalized version [67]. Therefore, CESI uses embedding strategies to create a semantic vector space representation for noun and verb phrases, i.e., similar phrases are embedded closely together. Then, clustering is performed to identify possibly synonymous phrases. In addition, the authors of [67] investigated how side information provided

by knowledge bases can be utilized to improve the clustering performance. There are also works that improve CESI further by utilizing language models in the process, e.g., see [10]. However, embedding phrases and clustering them comes with well-known challenges, e.g., finding a *good* vector representation for phrases, and interpreting clustered phrases, i.e., identifying which relation is hidden. Another challenge is to handle complex noun phrases that include more than a single concept. Semi-open relation extraction is another proposed option, which performs an Open IE and then filters for *relevant/interesting domain-specific* extractions [44]. However, relevancy must be defined by domain experts.

This thesis analyzes the application of two Open IE tools, namely the Stanford CoreNLP Open IE (2014) [52] and the language-model-based Open IE6 (2020) [32]. For a survey on other methods, we refer the reader to [58]. Our primary research question targets the utilization of these tools in a digital library setting, especially the canonicalization of the extractions. Moreover, we propose noun phrase filtering strategies with existing concept vocabularies, and a verb phrase canonicalization procedure that integrates domain experts into the process. As a comparison to our procedure, we reimplement a CESI-like method to canonicalize verb phrases and analyze its usefulness in a digital library scenario.

2.2. Nearly-Unsupervised Extraction Workflows

As a reminder, RDF represents structured information as triple-like statements, i.e., triples with subject-predicate-object shapes (s, p, o). For instance, (*Metformin*, *treats*, *diabetes mellitus*) states that Metformin treats diabetes mellitus. We have already seen that methods suitable for that task can be divided into two main categories: Either these methods are *closed* and trained towards those relations they should extract, or they are *open*, and the set of relations is not given a-priori. Closed methods rely on the curation of a schema, and often training data, which can become challenging in practice for a digital library (e.g., domain knowledge is required, data labeling costs money and time, and high-quality samples must carefully be picked to include all relevant relations, etc.). In contrast, open methods do not require training data but come with non-canonicalized extractions, i.e., extracted noun phrases (subjects and objects) and verb phrases do not refer to precise concepts and relations. In addition, these methods allow to explore what is told by a digital library collection. That is why we focused on these methods in the following.

Consider for example the following sentence: *The drug metformin is used to treat diabetes mellitus in patients.* Here, an open method might extract the noun phrases *the drug metformin* and *diabetes mellitus in patients.* Noun phrases may contain paraphrased versions of the same concept, e.g., *the drug metformin*, *drug metformin*, *metformin*, or even synonyms. These noun phrases are thus not canonicalized, i.e., they are not resolved to precise identifiers/concepts. The same applies to verb phrases: A plethora of different verb phrases might refer to the same relation. In summary, open methods without canonicalization do not offer canonicalized extractions, which could lead to useless extractions in practice.

That is why we proposed a method in between: **nearly-unsupervised extraction workflows**. The central idea is that digital libraries can reuse information they already have: well-curated vocabularies describing their domains. The previous example extraction can in this way be canonicalized by utilizing two different vocabularies. A concept vocabulary (e.g., for drugs and diseases) allows to filter noun phrases for domain-specific and relevant concepts. A relation vocabulary describes the set of domain-relevant relations

between these concepts, e.g., treats or induces between drugs and diseases. We therefore contributed a toolbox including dictionary-based concept linking, noun phrase filtering strategies, a canonicalization procedure for verb phrases, and cleaning constraints (e.g., treats can only be placed between drugs and diseases). The toolbox contains interfaces to Stanford CoreNLP Open IE [52] and Open IE6 [32]. In addition, we implemented a path-based extraction method called PathIE to have an adjustable, precision/recall-balanced extraction method as a comparison to Open IE. First, concepts need to be identified in the texts, e.g., through our dictionary-based concept linker. PathIE then extracts a statement between two concepts if they are connected through the grammatical structure of a sentence via a verb phrase or a special, pre-defined term. Domain experts can in this way define terms like *therapy*. If a drug and disease are then connected via the term *therapy*, an extraction (*drug, therapy, disease*) is made. Details can be found in our subsequent paper.

We demonstrated the toolbox's applicability in the biomedical domain and compared our workflows to established, supervised methods [36]. Moreover, our iterative predicate canonicalization procedure integrates domain experts into the process, which allows them to explore what is hidden in the collection. This iterative process can then be used to create the relation vocabulary from scratch. In addition, our procedure utilizes word embeddings to automatically infer more possible synonyms for a relation [56].

Our toolbox has been published in the following paper:

> [36] **Hermann Kroll**, Jan Pirklbauer, and Wolf-Tilo Balke. "A Toolbox for the Nearly-Unsupervised Construction of Digital Library Knowledge Graphs". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Urbana-Champaign, IL, USA, 2021, IEEE. DOI: https://doi.org/10.1109/JCDL52503.2021.00014

The toolbox code plus a comprehensive documentation was shared as open source on GitHub and the Software Heritage Project; see Appendix A.1.

In our previous paper, we evaluated the toolbox in the biomedical domain only. However, the biomedical domain has some practical advantages: It has well-curated vocabularies and ontologies, e.g., the Medical Subject Headings (MeSH)[1], the National Center for Biotechnology Information's gene vocabulary[2] or the BioOntology Portal[3]. Moreover, these vocabularies are curated, have precise terms, and are indeed used by authors, which eases the linking of concepts against them. In addition, concepts stand in very precise relations to each other, e.g., a drug might *treat* a disease or *induce* a disease. For instance, a domain like political science might not have well-described vocabularies, and may come with various relations that could be placed between concepts: Think about possible relations between persons, e.g., they could be in a family relation (spouse, father, mother, sibling, relative, etc.), in a work relation (employer, employee, student, professor, co-author, etc.), or do some action (dance with, play in a team with, like/dislike, etc.). The toolbox's application to the biomedical domain was thus a special use case.

Digital libraries, however, curate knowledge for various domains. That is why we focused on our toolbox's generalizability in the following paper. In brief, we investigated how well our methods generalize to other domains, namely, political science and encyclo-

[1] https://meshb.nlm.nih.gov, Last accessed: 10.09.2023
[2] https://www.ncbi.nlm.nih.gov/gene/, Last accessed: 10.09.2023
[3] https://bioportal.bioontology.org, Last accessed: 10.09.2023

pedic texts (in the example of Wikipedia). The specialized information service for political science (Pollux) provided example texts and assisted us in our proceeding. We formulated research questions that can be structured into three areas: application costs (what is necessary/prerequisites to apply the toolbox; how much effort and expertise is required), generalizability (how well do methods work beyond biomedical texts), and limitations (what is missing/lost in the extraction workflows). We also measured runtimes to estimate how well the applied methods scale to a real, extensive digital library collection.

Our findings have been reported in [39].

[39] **Hermann Kroll**, Jan Pirklbauer, Florian Plötzky, and Wolf-Tilo Balke. "A Library Perspective on Nearly-Unsupervised Information Extraction Workflows in Digital Libraries". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Cologne, Germany, 2022, ACM. DOI: https://doi.org/10.1145/3529372.3530924

The previous work focused on qualitative aspects and best practices, i.e., when methods work and fail. One of the observed issues was that the extracted noun phrase *complexity* was quite high, i.e., many noun phrases contained more than a single concept, which caused problems in the filtering step. That is why we extended our work by 1) providing more user study details, 2) quantifying the complexity of extracted noun phrases through a set of different metrics, and 3) analyzing a second open extraction method to generalize our findings. The verb phrase canonicalization was problematic too: Crafting a suitable relation vocabulary in domains such as political science was difficult due to various relations and ambiguous verb phrases like *use*. Thus, we implemented and tested a second, clustering-based verb phrase canonicalization procedure proposed in the related work CESI [67]. Beyond that, digital libraries may contain content in languages other than English. That is why we also investigated how our toolbox can handle non-English texts, mainly through automated machine translation. Again, code and scripts to reproduce our findings, as well as used data and produced results were made available in the toolbox repositories. We published our extended work in [37].

[37] **Hermann Kroll**, Jan Pirklbauer, Florian Plötzky, and Wolf-Tilo Balke. "A detailed library perspective on nearly unsupervised information extraction workflows in digital libraries". International Journal on Digital Libraries (IJDL) 2023. DOI: https://doi.org/10.1007/s00799-023-00368-z

In conclusion, nearly-unsupervised information extraction workflows allow digital libraries a new way to structure their textual collections. They bypass the need for training data for the concept linking and extraction phase completely, but require the design of suitable vocabularies. However, it must be noted, that those workflows can require extensive filtering in practice. The contribution is thus a practical workflow with moderate application costs (requirements, computation costs, required domain knowledge, etc.) that can be implemented in a digital library today. We contributed a novel verb-phrase canonicalization algorithm with an expert feedback loop and automated construction through word embeddings. Moreover, this thesis demonstrated how a digital library, here with the example of the pharmaceutical domain, can successfully deploy such a workflow.

3. Context-Compatible Information Fusion

The previous chapter focused on methods for the information extraction task and, more precisely, proposed nearly-unsupervised workflows to extract structured statements from natural language texts. Please consider the following example sentences:

(1) *Metformin is a first-line therapy for adults with diabetes mellitus.* (2) *This study purposed to explore the effects of nanoparticles combined with Metformin in a rat model.*

As a short recap, we understand a statement as a subject-predicate-object-shaped triple, e.g., (*Metformin*, *treats*, *diabetes mellitus*). However, statements are in this way restricted to a rather basic knowledge representation (triple representation). An information extraction might yield the following: From (1): (*Metformin*, *treats*, *diabetes mellitus*) and (*Metformin*, *treats*, *adults*). From (2): (*Metformin*, *administered*, *nanoparticle*).

This has two consequences. First, the coherence between information can be lost: A complex statement, e.g., *Metformin is a first-line therapy for adults with diabetes mellitus*, must be broken into two triples, e.g., *Metformin treats adults* and *Metformin treats diabetes mellitus*. The connection between *adults* and *diabetes mellitus* is then simply lost. Second, if we extract statements from texts, we usually tear them apart from their contexts, e.g., the administration of Metformin as a nanoparticle was only tested in a rat model. Especially in the scientific discourse, in which authors arrange statements carefully in their lines of arguments, tearing those statements apart can severely threaten the statements' *validity*.

A query could ask whether Metformin can be administered as a nanoparticle for adults with diabetes mellitus. A subsequent information fusion would now combine both statements to construct its answer, i.e., performing graph pattern matching between the query and the knowledge graph, which can be derived from the statements. The pure matching on the graph structure would result in a *Yes*-answer. However, this would not be valid: Metformin has just been tested as a nanoparticle for rats, but not for adults. Although the statement extraction was accurate, the fused statements' contexts do not match and thus the result is not valid. In this sense, we call the result of an information fusion **valid** if the used statements can safely be fused, i.e., their contexts match.

This property is hence not a consequence of an inaccurate information extraction. It is caused by the representation of complex circumstances as triples. The context is usually lost in precisely this transformation, as shown in our examples. In brief, two correctly extracted statements could still be fused and form an invalid result in the end because their *contexts* are not *compatible*. We understand any condition required for a statement to become valid as its **context**. We divide contexts into two categories: constraining contexts (every condition that must be known for the statement) and corresponding contexts (retaining the connection between statements). We call two statements **context-compatible** if we can safely fuse them to produce a valid result. Subsequently, we discuss how a digital library can implement a context-compatible information fusion in practice.

3.1. Related Work

There are various methods to enrich triple-shaped knowledge graphs, which allow to store contexts explicitly. Named graphs store quadruples [9], i.e., the statement plus its provenance. Provenance is a broad term usually understood as any kind of information that may validate some statement's quality or origin [73], i.e., how a statement has been derived or who crafted it. The PROV-O (provenance Ontology) [46] is an established standard to represent provenance information for knowledge graphs. PROV-O basically allows us to model a provenance graph for each statement, i.e., a whole provenance graph is crafted to describe a statement's origin (source, who crafted it, when it was defined, etc.).

An alternative is to use reification. Reification allows to make statements about statements, i.e., additional information about statements can be represented. Wikidata, for instance, uses so-called qualifiers (property-value pairs) to enrich its knowledge [23]. Another option is to represent the context for statements in some logic, e.g., McCarthy proposed to store contexts based on first-order predicate logic back in 1993 [54]. Explicit context models can then be used to contextualize a knowledge bases, e.g., see [63].

An alternative to enriching statements is to store information in n-ary relations directly [14]. While relational databases did that for a long time, extracting binary relations has been a topic of wide research (see Sect. 2.1). Ernst et al. argued to move beyond binary relations and directly extract n-ary relations from texts to store additional context [14]. However, n-ary relations require prior domain knowledge about each relation, i.e., additional and relevant context information must be known and defined for each relation in advance. Even worse, their method requires training data which can be challenging if not all information is contained within one sentence. Suchanek argued to move beyond triples, i.e., use more advanced knowledge representations [64]. However, *moving beyond triples* requires the design of new methods to extract, store, and query knowledge. In contrast, we proposed a practical context model for established triple-shaped representations.

We discussed these explicit models in our work [43], e.g., n-ary relations (context is kept inside a relation) [14], a first-order logic model (context and rules are defined by hand) [54], property-value pairs (like qualifiers in Wikidata) [23, 69], and provenance techniques [46].

[43] **Hermann Kroll**, Florian Plötzky, Jan Pirklbauer, and Wolf-Tilo Balke. "What a Publication Tells You – Benefits of Narrative Information Access in Digital Libraries". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Cologne, Germany, 2022, ACM. DOI: https://doi.org/10.1145/3529372.3530928

In brief, we understand provenance, methods like reification, and logic models as possible implementations to face the context problem. They tell us *how* to store information about statements, but not *how to get* and *work* with the contexts, e.g., what are the context conditions for some domain and when are two contexts compatible. Unfortunately, explicit models require domain knowledge to formulate the context conditions (what is relevant for a statement) and rules on *how* and *when* to combine different contexts.

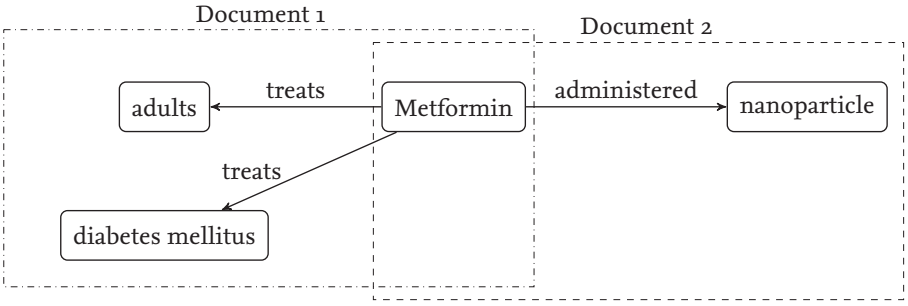


Figure 3.1.: Implicit context representation: The metformin therapy for adults with diabetes mellitus was extracted from Document 1, and its administration as a nanoparticle from Document 2. Both documents span the contexts surrounding their statements.

3.2. Implicit Contexts

We have seen that modeling context conditions explicitly is a tedious and challenging task: Applying an explicit context model requires domain experts to know and manually model every single condition in advance. And above all, rules are then required to state when two contexts are compatible. These rules, however, are challenging to craft, especially if the number of different conditions increases. That is why we introduced a novel kind of context. Our motivation for retaining contexts was caused by the rather simple triple-based statement representation that cannot keep up with complex circumstances.

As previously discussed in Chapter 1, humans share their knowledge as narratives. In the scientific discourse, they arrange statements in concise lines of arguments and publish them in a written form, e.g., articles and books. Digital libraries then maintain these documents in extensive collections. We argue that we should keep the statements' source documents when implementing extraction pipelines that harvest statements from texts, e.g., see SemMedDB harvested from the textual MEDLINE collection [30]. The idea of an **implicit context** is based on how documents are crafted: We assume their authors include relevant context conditions in their documents (e.g., articles, publications), especially in the scientific discourse (at best they should do). In other words: If they arrange statements together within their concise line of arguments, these statements should have the same context. Thus, simply keeping the reference of a statement to its source article is enough to retain its implicit context. Our model reuses their connection inside a textual document. It *spans* a scope around statements extracted from the same document, i.e., these statements belong together and can thus safely be fused later on. Our model bypasses the need to model context conditions explicitly and, hence, comes with less costs in practice, in terms of requirements to design, apply and extract contexts. Please note that this is, however, an approximation. For short abstracts contexts should be *stable*, i.e., statements from the same abstract could safely be fused. We use this assumption to implement narrative information access in the pharmaceutical domain. Retaining an implicit context is in this way as easy as keeping a reference to the statement's source. An example is shown in Figure 3.1: Two documents span the contexts around statements extracted from each. Please note that a long document, however, might contain several different contexts. Here, a more fine-grained implicit context should be stored, e.g., keeping a reference to the corresponding section of a document.

3.3. Context-Compatibility

In the following, we call the source of an implicit context a *document*, regardless of whether it is an article, a book, etc.. Implicit models allow us to retain the statements' contexts. Next, rules are required that define when statements can safely be fused. First, we proposed the **strict implicit context** rule, which restricts the information fusion to the document level, i.e., only statements extracted from the same document can be combined. This rule is unfortunately very restrictive: While we may retain high quality in the information fusion because we only fuse statements that an author has put together initially, we lose the capability to fuse statements from different sources. In science, discovering new knowledge is often based on the fusion of existing findings over documents' borders, which the strict implicit context rule prohibits by design. We therefore designed measures to estimate whether two different documents are context-compatible. They can be categorized as follows: The first set of measures is based on textual similarity measures. In brief, statements are context-compatible if their corresponding documents have similar texts, e.g., titles or abstracts. The second set is based on metadata of documents, e.g., authors, keywords or specialized metadata like chemical annotations in the biomedical domain. Here, we pursue ideas like the following: A group of authors from a scientific lab might work on similar topics with similar context conditions, i.e., if documents share the same set of authors or have overlapping ones, their contexts are compatible. Or, if documents utilize the same/similar chemicals in biomedicine, their contexts are compatible. Our measures allow us to weigh up between precision and recall through adjustable similarity thresholds, e.g., how similar texts or metadata should be for context-compatibility.

We published our implicit context model and context-compatibility measures in [40]. In addition, we quantified the consequences of an uncontrolled information fusion in the biomedical domain. Moreover, we also demonstrated the applicability of our implicit context model plus context-compatibility measures.

[40] **Hermann Kroll**, Jan-Christoph Kalo, Denis Nagel, Stephan Mennicke, and Wolf-Tilo Balke. "Context-Compatible Information Fusion for Scientific Knowledge Graphs". International Conference on Theory and Practice of Digital Libraries (TPDL), Lyon, France, 2020, Springer. DOI: https://doi.org/10.1007/978-3-030-54956-5_3

In conclusion, we introduced the relevance of context and a context-compatible information fusion to digital libraries. Moreover, we showed and discussed how contexts can be retained in practice, whether in the form of explicit or implicit context models. Since explicit contexts are challenging to define and maintain, we proposed implicit contexts for digital libraries. Retaining an implicit context is as easy as keeping a reference to the extracted statements' source documents. With implicit models and practical measures, we contributed a novel approach to how digital libraries can retain contexts and ensure a context-compatible information fusion in the end.

4. Narrative Information Access

In general, information retrieval is the task of finding relevant content with regard to some user query [51]. This thesis focuses on textual document retrieval for digital libraries. We discuss differences to our narrative information access in the following.

4.1. Related Work

Digital libraries can implement Boolean retrieval, i.e., documents are relevant if they contain the queried terms based on logical AND or OR expressions. In addition to Boolean retrieval, relevancy is used to create a ranked list of documents [51], e.g., via vector space models, tf-idf-based metrics, or BM25 rankings. Semantic retrieval and query expansions allow a more sophisticated retrieval, e.g., see works for medical [65] or cross-lingual medical text retrieval [61]. Ideas are, for example, to derive synonyms or paraphrases for specific terms from the corpus and rewrite queries so that synonyms are also included. Deep learning boosts retrieval further, e.g., by using neural information retrieval with query expansion [48], or by precise biomedical retrieval [78]. Here, relevancy is learned through training data. Another way is forcing users to express their information needs through linguistic patterns, e.g., see the SPIKE system [62] and its biomedical search function [66]. Here, a user may search for a certain expression like *Metformin treats DISEASE*, which should be answered by sentences that include this linguistic expression. In contrast, our extraction pipeline canonicalizes different linguistic patterns to obtain precise relations, i.e., users can directly search for those relations and do not have to state different linguistic patterns. Alternatively, we could force users to learn SQL for text retrieval [21].

While narrative information access still asks users to learn a new query paradigm, it minimizes the overhead for them: They formulate their information need as graph patterns that can be entered through a query builder. With that, no query language like SQL must be learned. Moreover, narrative information access does not require learning relevancy between queries and documents. Instead, we proposed a Boolean graph matching paradigm, i.e., documents that *match* the narrative query are equally relevant.

4.1.1. Graph-based Retrieval

Graph-based retrieval is a special form of information retrieval. Works in that area propose to utilize knowledge graphs to boost the retrieval quality [12], e.g., by utilizing entity information like synonyms to rewrite queries. Another work proposes to utilize Open IE to extract statements from texts and then use the extracted statements to improve the retrieval [29]. While these works are relevant to this thesis, they rather propose first ideas and metrics that could be deployed. In contrast, we implemented and evaluated a complete graph-based retrieval system in the pharmaceutical domain.

A second group of works proposes to transform literature into a knowledge graph, e.g., GrapAL [5], literature graphs [2], open research knowledge graph [26], Microsoft academic knowledge graph [15], and the most recent OpenAlex project [59]. While some works like OpenAlex are still continued as of September 2023, others have already been discontinued,

e.g., GraphAL and the Microsoft Academic knowledge graph. In brief, these works have in common that they propose to transform metadata and textual content into a knowledge graph representation. Then, users may explore the resulting graph with suitable interfaces (see the open research knowledge graph's interfaces [68]) or use query endpoints to pose queries in SPARQL. While this thesis also proposes to transform a textual collection into a graph representation, it differs in two ways: We proposed and implemented a workflow to automatically transform a whole biomedical collection into a graph representation. On top of that, narrative information access does not answer queries on a single knowledge graph level. Instead, every document in the collection spans its own graph, and queries are answered on the document level. In this way, a context-compatible information fusion is implemented to ensure the results' validity. Moreover, users still get the source documents so that they can estimate their relevance by and for themselves.

4.1.2. Question Answering

The goal of question answering (QA) is that a system must answer a user's posed question by retrieving and utilizing relevant content to generate its answer [28]. For instance, QKBFly [57] constructs a knowledge base on-the-fly to answer queries, by pre-processing and extracting statements from texts. Another option is to store textual content as edges in knowledge graphs and apply retrieval on them [77]. Recently, language models in combination with knowledge graphs have shown good results [74]. Although these methods sound promising, especially biomedical QA is far from being solved [28]: In brief, the central challenges involve the expensive dataset collection for training, the utilization of domain knowledge, the explanation of answers, as well as fairness and bias concerns.

The central difference between question answering and narrative information access is that the latter focuses again on a digital library's documents. More precisely, queries are answered on the document level, so that users can evaluate a document's quality. Especially in the scientific discourse, authors have to carefully check evidence and estimate whether the given context fits their own work. Here, a language model's hallucinated/fabricated answer without any evidence might simply be useless; see [27] for a discussion on hallucinations. In contrast, narrative information access returns bindings against documents, which provides evidence and excludes hallucinations by design.

4.1.3. Keyword Search on Structured Data

Even if a digital library collection can be transformed into a structured representation, e.g., a knowledge graph or a relational database, querying requires users to pose questions with query languages like SPARQL and SQL. Related work proposes to relieve users from learning a new query language by using keywords to search in those repositories. BANKS is one method to do so: Users pose keyword queries to search and browse in a relational databases [6]. Another work, BLINKS, proposes a search on a graph structure, including a ranking of matching graph fragments [22]. Further works propose to implement keyword search on RDF graphs [13, 8]. Diversity can then be integrated to explore the graph more efficiently by offering users diverse result sets [8]. Another group of works proposes to use keywords or natural language to derive a possible SQL or SPARQL query [19, 75]. For instance, [75] suggests to use keyword search for an incremental semantic query construction, i.e., users pose keywords and the system proposes possible queries.

An overview of natural language interfaces for databases can be found in [1] and a comprehensive benchmarking of text2SQL systems was investigated in [18]. However, such a procedure still requires users to understand the query language to select the intended query. Even if the query is executed directly, without showing it to the users first, users have no information about the correctness of the translated query or at least no control of the translation process. And even worse, those translation systems usually rely on training data, i.e., examples of text and corresponding SQL/SPARQL queries, which are typically unavailable in a digital library. For narrative information access, we investigated a keywords-to-graph translation. Still, in contrast to the previous works, we put our focus on an unsupervised translation algorithm, a different data model (millions of small graphs instead of a single, big knowledge graph), and user aspects, i.e., if the interface and query representation/visualization can be understood and used in our domain.

4.2. Demonstrating the Benefits of Narrative Information Access

Keyword-based access paths are common in digital libraries, e.g., the PubMed[1] search engine for the biomedical MEDLINE collection. Herskovic et al. performed an extensive query log analysis of PubMed [24] in 2007 and reported the following: 1) Users state 4.3 queries per session on average. A user-specific log analysis revealed that they include specific information about the keywords' intended semantic relations to navigate through the collection, e.g., *myocardial infarction AND aspirin* may be refined to *myocardial infarction prevention AND aspirin*. Hence, they indeed searched for relations between keywords. Please note that the authors did not have user sessions logs available and so they classified queries by time-intervals and similar topics as belonging to the same user. 2) Result set sizes can become quite challenging since they range between 1 and 4,844,731 documents. On average, they reach (rather unmanageable) 14,050 documents (median 68) with a standard deviation of 145,074. These findings strengthen our argument that keyword-based access comes with the following limitations: First, stating the relations between keywords can become exhausting when the search for relations like *prevention* could be paraphrased in various ways. Second, exploratory searches are not well supported because users cannot use variables in their queries to structure result lists, e.g., search for all diseases that can be treated with *Metformin*. Instead, they would have to leave out some keywords (e.g., the specific disease) and may have to browse through extensive result lists (e.g., by just searching for *Metformin* or *Metformin therapies*).

That is why we propose narrative information access: Formally, we define a narrative pattern as a directed, node- and edge-labeled graph with nodes being concepts and edges being interactions between them. We call an edge between two concepts a statement. A narrative query is then a narrative pattern in which nodes can be replaced by variables. Given a narrative query, a system must 1) bind all of the pattern's statements against data (to find evidence) and 2) ensure that these bindings are context-compatible so that the overall pattern becomes valid. If the query contains variables, the variables must be adequately substituted by concepts within the binding process. Here, structural compatibility must be ensured, i.e., some variable must be substituted for all query statements that include it by the same concept. This paradigm is similar to the variable substitution in SPARQL. We introduced narrative information access and its benefits in our work [43].

[1] `http://pubmed.ncbi.nlm.nih.gov`. Last Accessed: 10.09.2023

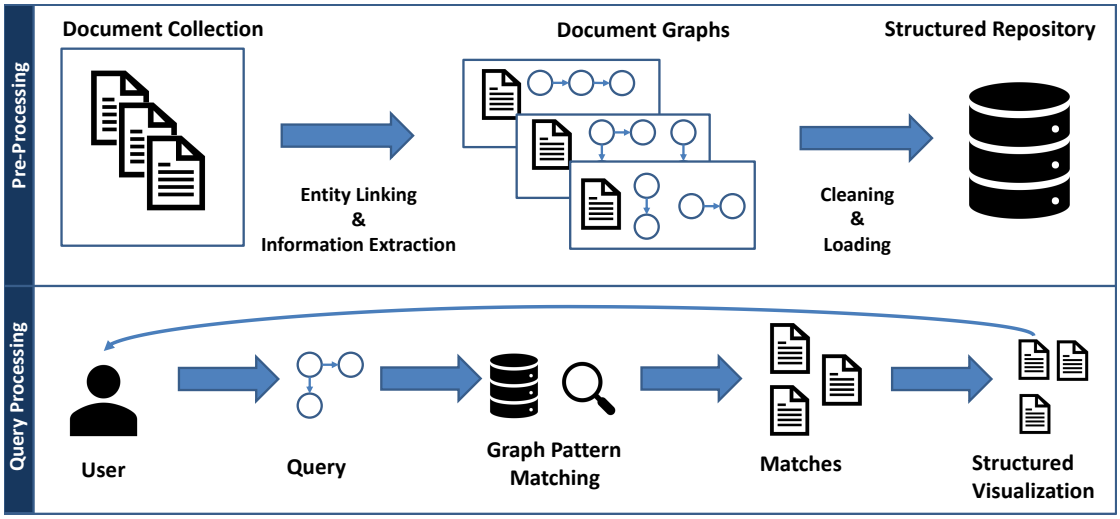


Figure 4.1.: Systematic overview of our system (taken from our work [42]): A pre-processing step transforms the textual documents into document graphs and stores them in a structured repository. Users pose queries as graph patterns. The system then performs a graph pattern matching to compute, return and visualize relevant documents.

[43] **Hermann Kroll**, Florian Plötzky, Jan Pirklbauer, and Wolf-Tilo Balke. "What a Publication Tells You – Benefits of Narrative Information Access in Digital Libraries". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Cologne, Germany, 2022, ACM. DOI: https://doi.org/10.1145/3529372.3530928

The key difference to knowledge graph querying, unless not explicitly modeled and queried, is the enforced context-compatibility. In the case of SPARQL with basic graph patterns, contexts are not used at all, and the matching may produce invalid results in the end; see our introductory example in Chapter 3, or our work [40] for consequences. Our previous work mainly focused on introducing narrative information access, discussing suitable context models and overall benefits for digital libraries by exemplifying the access in pharmacy and political science. Next, we show how a digital library can implement it.

We have already introduced nearly-unsupervised information extraction workflows to transform natural language texts into a graph representation. Our strict implicit context model allows us to enforce a practical context-compatible information fusion by retaining references to the statement's sources. If we restrict contexts to scientific abstracts, the assumption that a context is stable within an abstract should be applicable. This model then allows us to bypass an extensive, domain-specific modeling of explicit contexts conditions. Our next work focused on implementing narrative information access [42]. We tackled it from two sides: First, we discussed the implementation from a technical point of view, i.e., designing a domain-specific extraction pipeline, a query matching paradigm, fast query computation through inverted indexes, and in-memory hash-based joins, the translation of entered user inputs to precise concepts, and query expansions through ontologies. Second, we performed user studies to verify the suitability and acceptance of our overall approach, e.g., an user interface with helpful visualization strategies for queries with variables. A systematic overview of our system is shown in Figure 4.1.

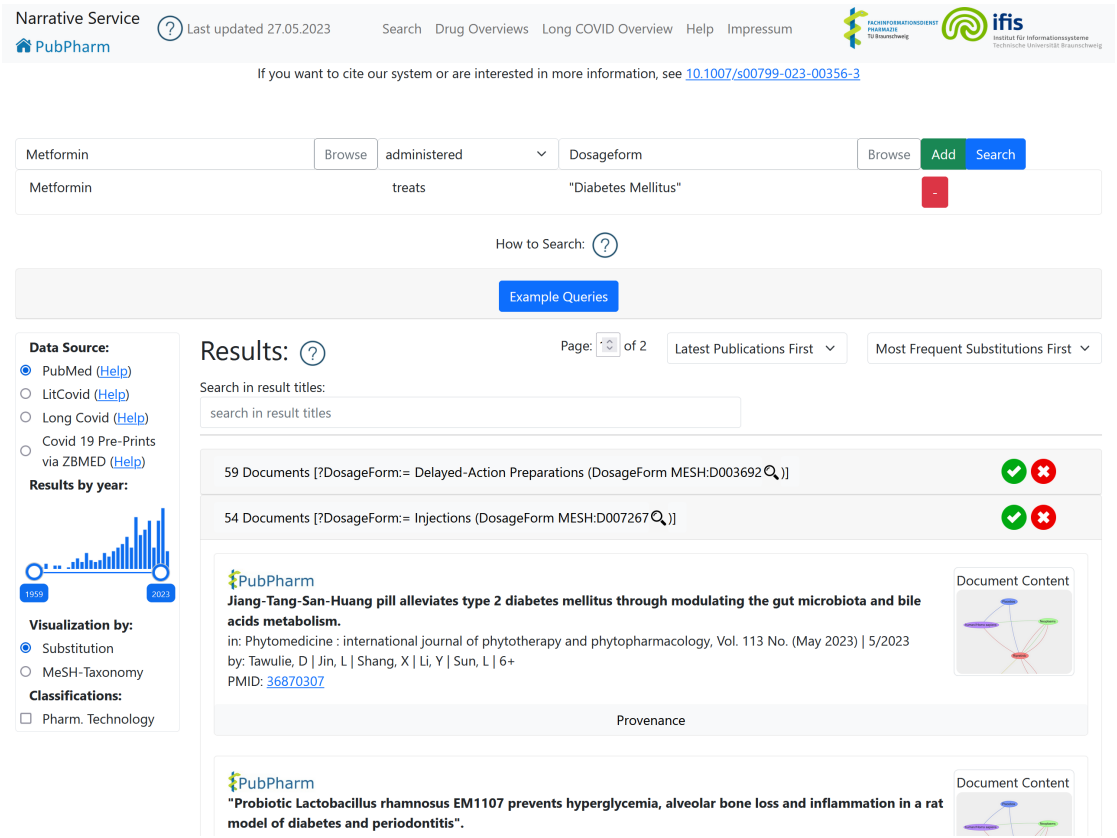


Figure 4.2.: A screenshot of the Narrative Service (`www.narrative.pubpharm.de`) shows a search for Metformin administrations (variable) to treat diabetes mellitus.

[42] **Hermann Kroll**, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. "Narrative Query Graphs for Entity-Interaction-Aware Document Retrieval". International Conference on Asian Digital Libraries (ICADL), Online, 2021, Springer. DOI: `https://doi.org/10.1007/978-3-030-91669-5_7`

This paper has led to the Narrative Service (`www.narrative.pubpharm.de`), which has been developed in the scope of this thesis and has been hosted by the specialized information service for Pharmacy (PubPharm). Figure 4.2 shows a screenshot of the system with an exemplary search. The service provides users with a provenance functionality, i.e., users can click on a document match and get an explanation of the reason why it matches. More precisely, the sentence(s) are shown from which the matching statements have been extracted. In this way, users can quickly estimate the document's relevance. This feature was claimed to be helpful in our user study [42].

To detect concepts in texts, we used external annotation services like PubTator [70, 71] or derived concepts vocabularies from knowledge bases like ChEMBL [55], and Wikidata [69]. We utilized our self-developed path-based method (PathIE) for the actual statement extraction. We compared PathIE to Open IE methods in our work [36]. In [42], we evaluated how well our proposed method performed in the biomedical retrieval setting by com-

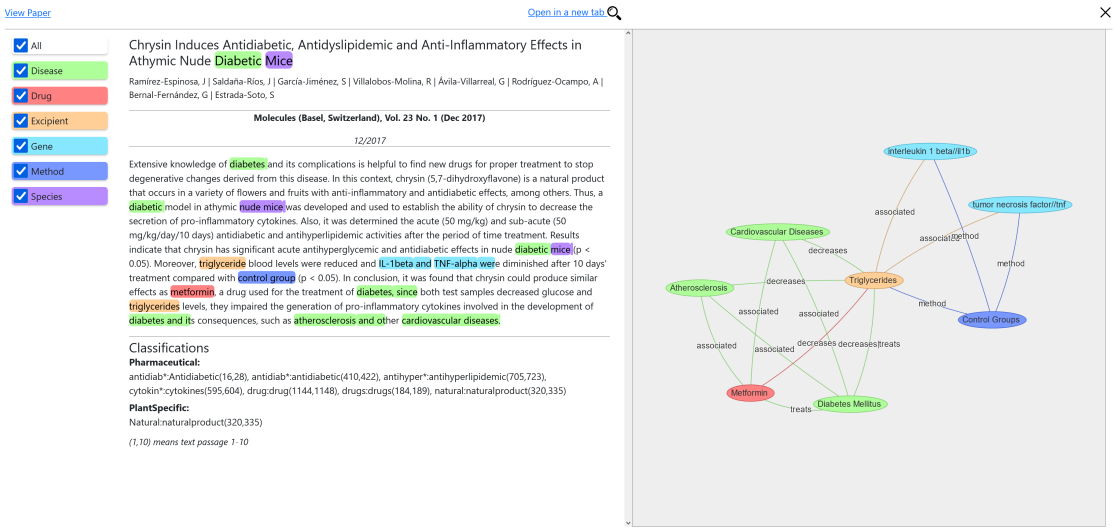


Figure 4.3.: The document graph visualization is shown. The left side highlights detected concepts in the text. The right side shows the extracted interactions in a graph visualization.

paring it to a keyword-based search with PubMed and a PubMed MeSH-term search. As a reminder, PubMed is a keyword-based search engine for biomedical literature. It also supports the search for Medical Subject Headings (MeSH) annotated in articles. In brief, our retrieval system was more precise than a PubMed term and comparable to a MeSH search if articles had MeSH annotations given. Our user study verified that our system was helpful and that our variable visualization strategies were understandable for our users.

We extended our work and published a more descriptive article [38]. We analyzed a second extraction method for the retrieval workflow: Stanford CoreNLP Open IE [52]. While CoreNLP achieved a higher precision, its recall was clearly lagging behind PathIE. In summary, PathIE had a better trade-off between precision and recall (F1 score). For our retrieval service, we preferred a better F1 because users can quickly check results through the offered provenance information. In addition to that, we published details about our database schema, used indexes, runtimes, and space requirements for the actual implementation. We also included more details of our query computation, especially how user-entered strings are translated to query objects. Here, a string could refer to multiple homonymous concepts. In addition to that, a concept like *diabetes mellitus* should, at best, be expanded to all of its subclass concepts, e.g., *diabetes mellitus type 1* and *type 2*.

[38] **Hermann Kroll**, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. "A discovery system for narrative query graphs: entity-interaction-aware document retrieval". International Journal on Digital Libraries (IJDL) 2023. DOI: https://doi.org/10.1007/s00799-023-00356-3

In [38], we also introduced a document visualization – called the document graph – to show what our system has extracted from some document's text; see Figure 4.3. The visualization highlights detected concepts directly in the text and depicts the extracted interactions as a colored, directed and labeled graph.

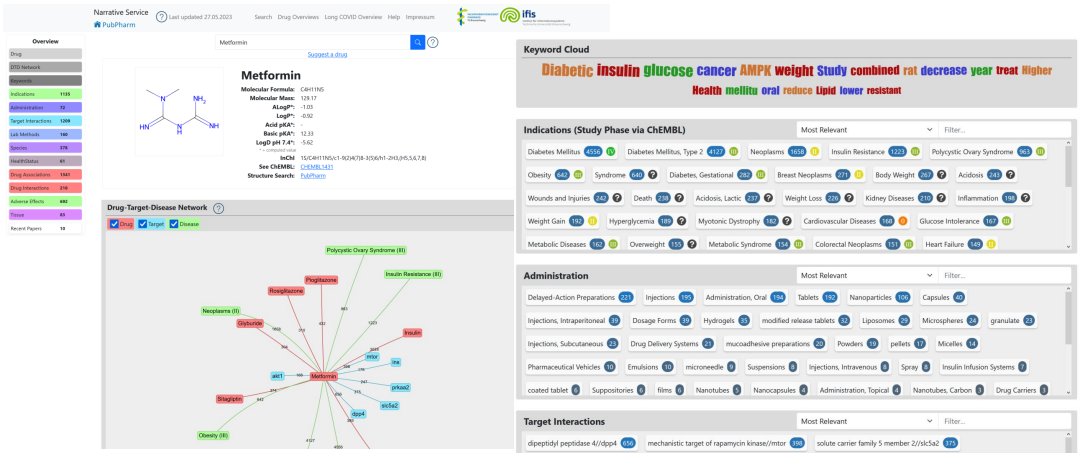


Figure 4.4.: A screenshot of the Drug Overview Service is shown. The left side shows information about the drug searched for (Metformin) in an info box and as a graph view. The right side depicts interactions structured in categories like indications or administrations.

4.3. Simplifying Narrative Information Access for Users

Further user studies for our Narrative Service revealed that users find it challenging to (1) explore the literature by using variables in the query and (2) formulate complex interaction patterns, i.e., search for more than a single statement. A query log analysis from 2021 and 2022 revealed that only 440 of 7268 queries contained more than one single statement.

We tackled (1) by introducing the so-called Drug Overviews (`www.narrative.pubpharm.de/drug_overview`). The Drug Overview Service summarizes information about drugs in one place; see Figure 4.4. First, the users enter a substance of interest. A set of predefined narrative queries is then executed, and the results are visualized in an info box, as a graph view and through structured substitution lists (basically sorted lists categorized into treatment options, administrations, interactions, etc.). The Drug Overview Service has briefly been described in [38]. The service allowed our users to quickly gain an overview of literature on a drug, e.g., by exploring therapy options, drug-drug interactions, target interactions, treated species, patient target groups, and administered dosage forms in one place. A click on some interaction then forwards the users to a corresponding search in the Narrative Service, which provides them with provenance (matching text snippets).

For (2), we developed an algorithm that assists users in formulating their queries. It takes a set of keywords as its input and returns a set of possible narrative queries that could be deduced from those keywords.

In contrast to the related work (e.g., natural language to SQL or SPARQL systems), our system had a different data model: A relational database or knowledge graph stores each tuple (here statement) once. In our scenario, a statement can be extracted from various different documents, i.e., we had a support criterion available for how many documents support a certain statement. This support criterion is beneficial for the query generation as it can be used to deduce patterns that are frequently mentioned in documents. Still, deducing narrative queries from keywords may not be ambiguous: For instance, a keyword could refer to different concepts (homonyms). In addition, we might have to decide between different relations: either distinguish on a similar level of detail (*treats* vs. *induces*), or select rather general relations like *associated* before specialized ones like *treats*.

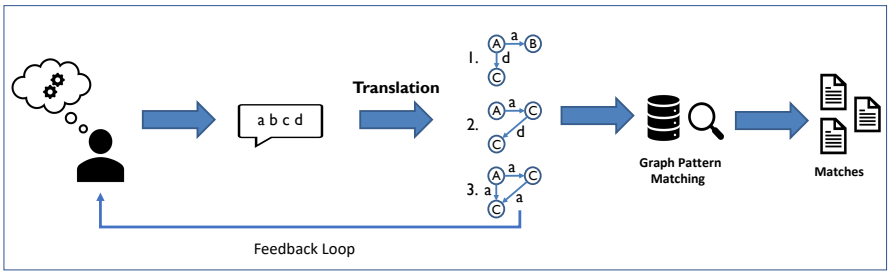


Figure 4.5.: Systematic overview about our keyword-to-graph system: Users formulate their information need as keywords. The system translates the keywords and offers a set of possible deduced narrative queries. Users select one query and start their search.

We tackled the ambiguity by integrating a feedback loop: Users state keywords and the system replies with a set of possible narrative queries. The narrative queries are visualized for the user in a suitable representation, and finally, users can select their intended query and start their search. A systematic overview is shown in Figure 4.5.

In our work [41], we focused on three aspects: 1) The design and implementation of the translation algorithm (keywords-to-graph), 2) a suitable query representation for the user feedback loop and 3) the effectiveness of our query model and the translation by testing different strategies on established biomedical information retrieval benchmarks.

[41] **Hermann Kroll**, Christin Katharina Kreutz, Pascal Sackhoff, and Wolf-Tilo Balke. "Enriching Simple Keyword Queries for Domain-Aware Narrative Retrieval". ACM/ IEEE Joint Conference on Digital Libraries (JCDL) Santa Fe, NM, USA, 2023, IEEE. DOI: https://doi.org/10.1109/JCDL57899.2023.00029 arXiv: https://doi.org/10.48550/arXiv.2304.07604

In brief, our work concluded that a graph representation was most suitable for our pharmaceutical users. They claimed they were familiar with graph representations because they formulate interaction mechanisms between drugs and targets as graphs in their research. Next, our query model outperformed a term-based search in terms of precision, recall and F1. Indeed, we demonstrated that our set of proposed translation strategies can find many of the best possible queries regarding one of these scores. With that, our strategies can provide users effectively with precision-oriented or F1-oriented query patterns.

In conclusion, as far as we know, we are the first who proposed narrative information access, demonstrated its usefulness, and implemented a complete retrieval system in the pharmaceutical domain. Therefore, we contributed a pure graph-based retrieval system for biomedical document retrieval. The system supports online retrieval for about 36 million biomedical documents (as of September 2023). We offered solutions on how to implement a fast and effective online retrieval as well as a suitable and accepted user interface. Moreover, we successfully integrated extensions in the form of novel interfaces like the Drug Overviews or eased access paths like keywords-to-graph into our system. Finally, we shared our Narrative Service, our Drug Overviews and our pharmaceutical extraction pipeline as open source (see Appendix A.1), so that digital libraries have an example implementation for narrative information access.

5. Conclusion and Outlook

This thesis contributes narrative information access to digital libraries by proposing a conceptualization, discussing its benefits, implementing novel, suitable extraction workflows and a full-fledged discovery system for the pharmaceutical domain. Narrative information access extends pure knowledge base querying by enforcing a context-compatible information fusion. This step ensures that we only combine knowledge that belongs together – in the sense of compatible context conditions to produce valid results in the end. Ensuring contexts is not an easy task: Explicit context models require to manually model context conditions and manually design rules on how to combine them. This step is especially challenging for a digital library because experts need to describe a domain in detail and in advance – before the extractions are made. That is why we contributed a novel, implicit context model that requires no more than keeping a reference to the source of an extracted statement. Context-compatibility realized through textual or metadata-based similarity measures also allows a digital library to go beyond fusing information from the same source, boosting the capability to discover new knowledge.

To actually implement narrative information access, a retrieval system must be designed: Such a system must be capable of realizing fast online retrieval and must come with intuitive interfaces/access paths to use it. Without them, the system will not be accepted by real users. This thesis tackles both challenges and proposes possible solutions for each. We proposed to transform a document collection into a graph representation for fast online retrieval. Therefore, we designed nearly-unsupervised information extraction workflows that allow us to transform textual collections into a graph representation, more precisely, textual documents into document graphs. Our workflows come with acceptable costs and do not require training data for the extraction phase. We demonstrated how to establish a complete pharmaceutical extraction workflow within this thesis' scope. Finally, we designed the retrieval system: We demonstrated effective solutions involving query translations, graph storage, retrieval with inverted graph indexes, and suitable user interfaces verified in user studies. Moreover, we also proposed Drug Overviews and keyword-to-narrative query translations to ease the access for the system's user. Our user studies have verified that a feedback loop and the graph representation were considered helpful and intuitive when translating keyword queries. And beyond that, we demonstrated the effectiveness of our query model and translation strategies on biomedical information retrieval benchmarks.

In conclusion, this thesis contributes narrative information access and demonstrates its benefits. Narrative information access allows precise searches through stating interactions between concepts directly, as well as structured searches through variables to explore the content of a digital library. It is an extension to knowledge base querying by ensuring the validity of results through a context-compatible information fusion. Moreover, we implemented and evaluated a full-fledged discovery system for the pharmaceutical domain and thus demonstrated how it can be realized by a digital library today. While we focused mainly on the pharmaceutical domain with its concept-centric knowledge, we also investigated extraction workflows and narrative information access in political science to generalize our findings.

Limitations. It must clearly be stated that although our extraction workflows can be transferred to other domains, they might not always be the best option. The biomedical domain has well-curated ontologies and precise relations that are used in practice. This property does not generalize to other domains, e.g., see our work in the political science where vocabularies were not available, and relations were hard to define [37]. Especially with the advent of language models, other ways to realize the extraction, e.g., as few/zero-shot approaches, might be an option that should not be ignored. These models minimize the amount of required training data and still retain an acceptable accuracy, which could be a better trade-off for digital libraries. However, they still require extensive computation power and, on top of that, require effective prompting strategies, whereas our extraction workflows come with lower application costs.

Another limitation is the evaluation of our system in the pharmaceutical domain only: Pharmacists are familiar with searching for biomedical concepts and especially their interactions. That is why we assume that formulating graph patterns might be easier for them than, for example, for experts in the political science domain. To generalize our findings to a certain extent, we discussed our access with political science experts and verified the usefulness of narrative information access in their domain; see [43].

Outlook. Our implicit context model comes with strong assumptions, e.g., stable contexts within abstracts and context-compatibility approximations through texts or metadata like authors. Future work should investigate 1) the handling of long, full-text documents that include multiple contexts, e.g., by detecting topic drifts to approximate context boundaries, and 2) more sophisticated context-compatibility measures, e.g., story-based (argumentation plus contextual setting) similarity measures between documents.

Our proposed discovery system applies Boolean retrieval and does not create a ranked document result list yet, i.e., all documents that contain the searched pattern are similarly relevant and are sorted by their publication date at the moment. Here, future work could design ranking methods that utilize the graph structure of documents. For instance, a document that puts the pattern in its center could be more relevant than documents that mention the pattern only as a side note. Such mechanisms could especially be relevant if narrative information access is implemented for full-text retrieval. In addition, a relaxed query mechanism could be designed, i.e., documents that match the pattern completely could be placed before documents that only contain parts of the query.

Another direction would be to investigate the notion of *plausibility*. We argued that a pattern is plausible if it is completely bound against some knowledge repositories. In practice, however, it might be much more complex. Sources come with trustworthiness (e.g., peer-reviewed articles vs. preprints), and bindings may come with some confidence (e.g., extraction confidence). We already discussed possible dimensions for plausibility in our work [34], but a comprehensive, practical solution remains open.

"There is no real ending. It's just the place where you stop the story."

Frank Herbert

”So long, and thanks for all the fish.”

Douglas Adams, The Hitchhiker’s Guide to the Galaxy

References

[1] Katrin Affolter, Kurt Stockinger, and Abraham Bernstein. "A comparative survey of recent natural language interfaces for databases". In: *VLDB J.* 28.5 (2019), pp. 793–819. doi: 10.1007/s00778-019-00567-8.

[2] Waleed Ammar et al. "Construction of the Literature Graph in Semantic Scholar". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. Association for Computational Linguistics, June 2018, pp. 84–91. doi: 10.18653/v1/N18-3011.

[3] Gabor Angeli, Melvin J. Johnson Premkumar, and Christopher D. Manning. "Leveraging Linguistic Structure For Open Domain Information Extraction". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. The Association for Computer Linguistics, 2015, pp. 344–354. doi: 10.3115/v1/p15-1034.

[4] Sören Auer et al. "DBpedia: A Nucleus for a Web of Open Data". In: *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*. Vol. 4825. Lecture Notes in Computer Science. Springer, 2007, pp. 722–735. doi: 10.1007/978-3-540-76298-0_52.

[5] Christine Betts, Joanna Power, and Waleed Ammar. "GrapAL: Connecting the Dots in Scientific Literature". In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*. Association for Computational Linguistics, 2019, pp. 147–152. doi: 10.18653/v1/p19-3025.

[6] Gaurav Bhalotia et al. "Keyword Searching and Browsing in Databases using BANKS". In: *Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, USA, February 26 - March 1, 2002*. IEEE Computer Society, 2002, pp. 431–440. doi: 10.1109/ICDE.2002.994756.

[7] Sangnie Bhardwaj, Samarth Aggarwal, and Mausam. "CaRB: A Crowdsourced Benchmark for Open IE". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics, 2019, pp. 6261–6266. doi: 10.18653/v1/D19-1651.

[8] Nikos Bikakis et al. "RDivF: Diversifying Keyword Search on RDF Graphs". In: *Research and Advanced Technology for Digital Libraries - International Conference on Theory and Practice of Digital Libraries, TPDL 2013, Valletta, Malta, September 22-26, 2013. Proceedings*. Vol. 8092. Lecture Notes in Computer Science. Springer, 2013, pp. 413–416. doi: 10.1007/978-3-642-40501-3_49.

[9] Jeremy J. Carroll et al. “Named graphs, provenance and trust”. In: *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005*. ACM, 2005, pp. 613–622. doi: 10.1145/1060745.1060835.

[10] Sarthak Dash et al. *Joint Entity and Relation Canonicalization in Open Knowledge Graphs using Variational Autoencoders*. 2020. arXiv: 2012.04780.

[11] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186. doi: 10.18653/v1/n19-1423.

[12] Laura Dietz, Alexander Kotov, and Edgar Meij. “Utilizing Knowledge Graphs for Text-Centric Information Retrieval”. In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. ACM, 2018, pp. 1387–1390. doi: 10.1145/3209978.3210187.

[13] Shady Elbassuoni and Roi Blanco. “Keyword Search over RDF Graphs”. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. CIKM ’11. Glasgow, Scotland, UK: Association for Computing Machinery, 2011, pp. 237–242. isbn: 9781450307178. doi: 10.1145/2063576.2063615.

[14] Patrick Ernst, Amy Siu, and Gerhard Weikum. “HighLife: Higher-Arity Fact Harvesting”. In: *Proceedings of the 2018 World Wide Web Conference*. WWW ’18. International World Wide Web Conferences Steering Committee, 2018, pp. 1013–1022. isbn: 9781450356398. doi: 10.1145/3178876.3186000.

[15] Michael Färber. “The Microsoft Academic Knowledge Graph: A Linked Data Source with 8 Billion Triples of Scholarly Data”. In: *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*. Vol. 11779. Lecture Notes in Computer Science. Springer, 2019, pp. 113–129. doi: 10.1007/978-3-030-30796-7_8.

[16] James B Freeman. *Argument Structure:: Representation and Theory*. Vol. 18. Berlin/Heidelberg, Germany: Springer Science & Business Media, 2011.

[17] Kiril Gashteovski et al. “BenchIE: A Framework for Multi-Faceted Fact-Based Open Information Extraction Evaluation”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*. Association for Computational Linguistics, 2022, pp. 4472–4490. doi: 10.18653/v1/2022.acl-long.307.

[18] Orest Gkini et al. “An In-Depth Benchmarking of Text-to-SQL Systems”. In: *Proceedings of the 2021 International Conference on Management of Data*. SIGMOD ’21. Virtual Event, China: Association for Computing Machinery, 2021, pp. 632–644. isbn: 9781450383431. doi: 10.1145/3448016.3452836.

[19] Katerina Gkirtzou et al. “Keywords-To-SPARQL Translation for RDF Data Search and Exploration”. In: *Research and Advanced Technology for Digital Libraries - 19th International Conference on Theory and Practice of Digital Libraries, TPDL 2015, Poznań, Poland, September 14-18, 2015. Proceedings*. Vol. 9316. Lecture Notes in Computer Science. Springer, 2015, pp. 111–123. doi: 10.1007/978-3-319-24592-8_9.

[20] Paul Groth et al. “Open Information Extraction on Scientific Text: An Evaluation”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 3414–3423. url: https://aclanthology.org/C18-1289.

[21] Benjamin Hättasch et al. “WannaDB: Ad-hoc SQL Queries over Text Collections”. In: *Datenbanksysteme für Business, Technologie und Web (BTW 2023), 20. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme” (DBIS), 06.-10, März 2023, Dresden, Germany, Proceedings*. Vol. P-331. LNI. Gesellschaft für Informatik, 2023, pp. 157–181. doi: 10.18420/BTW2023-08.

[22] Hao He et al. “BLINKS: ranked keyword searches on graphs”. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, Beijing, China, June 12-14, 2007*. ACM, 2007, pp. 305–316. doi: 10.1145/1247480.1247516.

[23] Daniel Hernández, Aidan Hogan, and Markus Krötzsch. “Reifying RDF: What Works Well With Wikidata?” In: *Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems co-located with 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 11, 2015*. Vol. 1457. CEUR Workshop Proceedings. CEUR-WS.org, 2015, pp. 32–47. url: https://ceur-ws.org/Vol-1457/SSWS2015%5C_paper3.pdf.

[24] Jorge R. Herskovic et al. “A Day in the Life of PubMed: Analysis of a Typical Day’s Query Log”. In: *Journal of the American Medical Informatics Association* 14.2 (Mar. 2007), pp. 212–220. issn: 1067-5027.

[25] Dimitar Hristovski et al. “Constructing a Graph Database for Semantic Literature-Based Discovery”. In: *Studies in health technology and informatics* 216 (2015), p. 1094. issn: 0926-9630. url: http://europepmc.org/abstract/MED/26262393.

[26] Mohamad Yaser Jaradeh et al. “Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge”. In: *Proceedings of the 10th International Conference on Knowledge Capture, K-CAP 2019, Marina Del Rey, CA, USA, November 19-21, 2019*. ACM, 2019, pp. 243–246. doi: 10.1145/3360901.3364435.

[27] Ziwei Ji et al. “Survey of Hallucination in Natural Language Generation”. In: *ACM Comput. Surv.* 55.12 (Mar. 2023). issn: 0360-0300. doi: 10.1145/3571730.

[28] Qiao Jin et al. “Biomedical Question Answering: A Survey of Approaches and Challenges”. In: *ACM Comput. Surv.* 55.2 (Jan. 2022). issn: 0360-0300. doi: 10.1145/3490238.

[29] Amina Kadry and Laura Dietz. “Open Relation Extraction for Support Passage Retrieval: Merit and Open Issues”. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. ACM, 2017, pp. 1149–1152. doi: 10.1145/3077136.3080744.

[30] Halil Kilicoglu et al. “SemMedDB: a PubMed-scale repository of biomedical semantic predications”. In: *Bioinformatics* 28.23 (Oct. 2012), pp. 3158–3160. issn: 1367-4803. doi: 10.1093/bioinformatics/bts591.

[31] Keshav Kolluru et al. “Alignment-Augmented Consistent Translation for Multilingual Open Information Extraction”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2502–2517. doi: 10.18653/v1/2022.acl-long.179.

[32] Keshav Kolluru et al. “OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Association for Computational Linguistics, 2020, pp. 3748–3761. doi: 10.18653/v1/2020.emnlp-main.306.

[33] Hermann Kroll and Wolf-Tilo Balke. “On Design Principles for Narrative Information Systems”. In: *Proceedings of the Workshop on Semantic Techniques for Narrative-Based Understanding co-located with 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence (IJCAI-ECAI 2022), Vienna, Austria, July 24, 2022*. Vol. 3322. CEUR Workshop Proceedings. CEUR-WS.org, 2022, pp. 11–18. url: https://ceur-ws.org/Vol-3322/short3.pdf.

[34] Hermann Kroll, Niklas Mainzer, and Wolf-Tilo Balke. “On Dimensions of Plausibility for Narrative Information Access to Digital Libraries”. In: *Linking Theory and Practice of Digital Libraries - 26th International Conference on Theory and Practice of Digital Libraries, TPDL 2022, Padua, Italy, September 20-23, 2022, Proceedings*. Vol. 13541. Lecture Notes in Computer Science. Springer, 2022, pp. 433–441. doi: 10.1007/978-3-031-16802-4_43.

[35] Hermann Kroll, Denis Nagel, and Wolf-Tilo Balke. “Modeling Narrative Structures in Logical Overlays on Top of Knowledge Repositories”. In: *Conceptual Modeling - 39th International Conference, ER 2020, Vienna, Austria, November 3-6, 2020, Proceedings*. Vol. 12400. Lecture Notes in Computer Science. Springer, 2020, pp. 250–260. doi: 10.1007/978-3-030-62522-1_18.

[36] Hermann Kroll, Jan Pirklbauer, and Wolf-Tilo Balke. “A Toolbox for the Nearly-Unsupervised Construction of Digital Library Knowledge Graphs”. In: *ACM/IEEE Joint Conference on Digital Libraries, JCDL 2021, Champaign, IL, USA, September 27-30, 2021*. IEEE, 2021, pp. 21–30. doi: 10.1109/JCDL52503.2021.00014.

[37] Hermann Kroll et al. “A detailed library perspective on nearly unsupervised information extraction workflows in digital libraries”. In: *International Journal on Digital Libraries* (June 2023). issn: 1432-1300. doi: 10.1007/s00799-023-00368-z.

[38] Hermann Kroll et al. “A discovery system for narrative query graphs: entity-interaction-aware document retrieval”. In: *International Journal on Digital Libraries* (Apr. 2023). issn: 1432-1300. doi: 10.1007/s00799-023-00356-3.

[39] Hermann Kroll et al. “A Library Perspective on Nearly-Unsupervised Information Extraction Workflows in Digital Libraries”. In: *JCDL ’22: The ACM/IEEE Joint Conference on Digital Libraries in 2022, Cologne, Germany, June 20 - 24, 2022*. ACM, 2022, p. 35. doi: 10.1145/3529372.3530924.

[40] Hermann Kroll et al. “Context-Compatible Information Fusion for Scientific Knowledge Graphs”. In: *Digital Libraries for Open Knowledge - 24th International Conference on Theory and Practice of Digital Libraries, TPDL 2020, Lyon, France, August 25-27, 2020, Proceedings*. Vol. 12246. Lecture Notes in Computer Science. Springer, 2020, pp. 33–47. doi: 10.1007/978-3-030-54956-5_3.

[41] Hermann Kroll et al. “Enriching Simple Keyword Queries for Domain-Aware Narrative Retrieval”. In: *JCDL ’23: The ACM/IEEE Joint Conference on Digital Libraries in 2023, Santa Fe, NM, USA, June 26 - 30, 2023*. IEEE, 2023. doi: 10.48550/arXiv.2304.07604.

[42] Hermann Kroll et al. “Narrative Query Graphs for Entity-Interaction-Aware Document Retrieval”. In: *Towards Open and Trustworthy Digital Societies - 23rd International Conference on Asia-Pacific Digital Libraries, ICADL 2021, Virtual Event, December 1-3, 2021, Proceedings*. Vol. 13133. Lecture Notes in Computer Science. Springer, 2021, pp. 80–95. doi: 10.1007/978-3-030-91669-5_7.

[43] Hermann Kroll et al. “What a Publication Tells You—Benefits of Narrative Information Access in Digital Libraries”. In: *JCDL ’22: The ACM/IEEE Joint Conference on Digital Libraries in 2022, Cologne, Germany, June 20 - 24, 2022*. ACM, 2022, p. 9. doi: 10.1145/3529372.3530928.

[44] Ruben Kruiper et al. “In Layman’s Terms: Semi-Open Relation Extraction from Scientific Texts”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 2020, pp. 1489–1500. doi: 10.18653/v1/2020.acl-main.137.

[45] János László. *The science of stories: An introduction to narrative psychology*. Oxfordshire, England, UK: Routledge, 2008.

[46] T. Lebo, S. Sahoo, and D. McGuinness. *PROV-O: The PROV Ontology*. https://www.w3.org/TR/prov-o/. 2013.

[47] Jinhyuk Lee et al. “BioBERT: a pre-trained biomedical language representation model for biomedical text mining”. In: *Bioinform.* 36.4 (2020), pp. 1234–1240. doi: 10.1093/bioinformatics/btz682.

[48] Xiangsheng Li et al. “A Cooperative Neural Information Retrieval Pipeline with Knowledge Enhanced Automatic Query Reformulation”. In: *WSDM ’22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*. ACM, 2022, pp. 553–561. doi: 10.1145/3488560.3498516.

[49] Pengfei Liu et al. “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing”. In: *ACM Comput. Surv.* 55.9 (2023), 195:1–195:35. doi: 10.1145/3560815.

[50] Yao Lu et al. “Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8086–8098. doi: 10.18653/v1/2022.acl-long.556.

[51] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008. isbn: 978-0-521-86571-5. doi: 10.1017/CBO9780511809071.

[52] Christopher D. Manning et al. “The Stanford CoreNLP Natural Language Processing Toolkit”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*. The Association for Computer Linguistics, 2014, pp. 55–60. doi: 10.3115/v1/p14-5010.

[53] Frank Manola, Eric Miller, Brian McBride, et al. “RDF primer”. In: *W3C recommendation* 10.1-107 (2004), p. 6.

[54] John McCarthy. “Notes on Formalizing Context”. In: *Proceedings of the 13th International Joint Conference on Artificial Intelligence. Chambéry, France, August 28 - September 3, 1993*. Morgan Kaufmann, 1993, pp. 555–562. url: http://www-formal.stanford.edu/jmc/context3/context3.html.

[55] David Mendez et al. “ChEMBL: towards direct deposition of bioassay data”. In: *Nucleic Acids Research* 47.D1 (Nov. 2018), pp. D930–D940. issn: 0305-1048. doi: 10.1093/nar/gky1075.

[56] Tomás Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. In: *1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings*. 2013. url: http://arxiv.org/abs/1301.3781.

[57] Dat Ba Nguyen et al. “Query-Driven On-The-Fly Knowledge Base Construction”. In: *Proc. VLDB Endow.* 11.1 (2017), pp. 66–79. doi: 10.14778/3151113.3151119.

[58] Christina Niklaus et al. “A Survey on Open Information Extraction”. In: *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*. Association for Computational Linguistics, 2018, pp. 3866–3878. url: https://aclanthology.org/C18-1326/.

[59] Jason Priem, Heather Piwowar, and Richard Orr. *OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts*. 2022. doi: 10.48550/ARXIV.2205.01833.

[60] Alexander Ratner et al. “Snorkel: rapid training data creation with weak supervision”. In: *VLDB J.* 29.2-3 (2020), pp. 709–730. doi: 10.1007/s00778-019-00552-1.

[61] Shadi Saleh and Pavel Pecina. “Term Selection for Query Expansion in Medical Cross-Lingual Information Retrieval”. In: *Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14-18, 2019, Proceedings, Part I*. Vol. 11437. Lecture Notes in Computer Science. Springer, 2019, pp. 507–522. doi: 10.1007/978-3-030-15712-8_33.

[62] Micah Shlain et al. “Syntactic Search by Example”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 2020, pp. 17–23. doi: 10.18653/v1/2020.acl-demos.3.

[63] Heiko Stoermer et al. “Contextualization of a RDF Knowledge Base in the VIKEF Project”. In: *Digital Libraries: Achievements, Challenges and Opportunities*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 101–110. isbn: 978-3-540-49377-8.

[64] Fabian M. Suchanek. "The Need to Move beyond Triples". In: *Proceedings of Text2Story - Third Workshop on Narrative Extraction From Texts co-located with 42nd European Conference on Information Retrieval, Text2Story@ECIR 2020, Lisbon, Portugal, April 14th, 2020 [online only]*. Vol. 2593. CEUR Workshop Proceedings. CEUR-WS.org, 2020, pp. 95–104. url: https://ceur-ws.org/Vol-2593/paper12.pdf.

[65] Lynda Tamine and Lorraine Goeuriot. "Semantic Information Retrieval on Medical Texts: Research Challenges, Survey, and Open Issues". In: *ACM Comput. Surv.* 54.7 (2022), 146:1–146:38. doi: 10.1145/3462476.

[66] Hillel Taub-Tabib et al. "Interactive Extractive Search over Biomedical Corpora". In: *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing, BioNLP 2020, Online, July 9, 2020*. Association for Computational Linguistics, 2020, pp. 28–37. doi: 10.18653/v1/2020.bionlp-1.3.

[67] Shikhar Vashishth, Prince Jain, and Partha P. Talukdar. "CESI: Canonicalizing Open Knowledge Bases using Embeddings and Side Information". In: *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*. ACM, 2018, pp. 1317–1327. doi: 10.1145/3178876.3186030.

[68] Lars Vogt et al. *Webinar: Introduction to the Open Research Knowledge Graph (ORKG)*. Technische Informationsbibliothek (TIB). 2021. doi: 10.5446/52956.

[69] Denny Vrandecic and Markus Krötzsch. "Wikidata: a free collaborative knowledgebase". In: *Commun. ACM* 57.10 (2014), pp. 78–85. doi: 10.1145/2629489.

[70] Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. "PubTator: a web-based text mining tool for assisting biocuration". In: *Nucleic Acids Res.* 41.Webserver-Issue (2013), pp. 518–522. doi: 10.1093/nar/gkt441.

[71] Chih-Hsuan Wei et al. "PubTator central: automated concept annotation for biomedical full text articles". In: *Nucleic Acids Res.* 47.Webserver-Issue (2019), W587–W593. doi: 10.1093/nar/gkz389.

[72] Gerhard Weikum et al. "Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases". In: *Found. Trends Databases* 10.2-4 (2021), pp. 108–490. doi: 10.1561/1900000064.

[73] M. Wylot et al. "Storing, Tracking, and Querying Provenance in Linked Data". In: *IEEE Transactions on Knowledge and Data Engineering* 29.8 (2017), pp. 1751–1764.

[74] Michihiro Yasunaga et al. "Deep Bidirectional Language-Knowledge Graph Pretraining". In: *NeurIPS*. 2022. url: https://proceedings.neurips.cc/paper_files/paper/2022/file/f224f056694bcfe465c5d84579785761-Paper-Conference.pdf.

[75] Gideon Zenz et al. "From Keywords to Semantic Queries-Incremental Query Construction on the Semantic Web". In: *Web Semant.* 7.3 (Sept. 2009), pp. 166–176. issn: 1570-8268. doi: 10.1016/j.websem.2009.07.005.

[76] Rui Zhang et al. "Using semantic predications to uncover drug–drug interactions in clinical data". In: *Journal of Biomedical Informatics* 49 (2014), pp. 134–147. issn: 1532-0464. doi: 10.1016/j.jbi.2014.01.004.

[77] Chen Zhao et al. “Complex Factoid Question Answering with a Free-Text Knowledge Graph”. In: *WWW ’20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. ACM / IW3C2, 2020, pp. 1205–1216. url: https://doi.org/10.1145/3366423.3380197.

[78] Sendong Zhao et al. “GRAPHENE: A Precise Biomedical Literature Retrieval Engine with Graph Augmented Deep Learning and External Knowledge Empowerment”. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019*. ACM, 2019, pp. 149–158. doi: 10.1145/3357384.3358038.

A. Appendix

A.1. Code and Data

Context-compatible information fusion:

1. **GitHub:** `https://github.com/HermannKroll/ContextInformationFusion`
2. **Software Heritage:** `https://archive.softwareheritage.org/swh:1:dir:d46e07ff51a41d2770ba26d8cb736a3179d423db`

Nearly-unsupervised information extraction toolbox (KGExtractionToolbox):

1. **GitHub:** `https://github.com/HermannKroll/KGExtractionToolbox`
2. **Software Heritage:** `https://archive.softwareheritage.org/swh:1:dir:f748da901c1a5cc3e31769557ed14234423d2687`

Pharmaceutical extraction pipeline fork (NarrativeAnnotation):

1. **GitHub:** `https://github.com/HermannKroll/NarrativeAnnotation`
2. **Software Heritage:** `https://archive.softwareheritage.org/swh:1:dir:7b4a8f9245d33fc6fcca7c9ff099743ecda92876`

Narrative Service and Drug Overviews (NarrativeIntelligence):

1. **GitHub:** `https://github.com/HermannKroll/NarrativeIntelligence`
2. **Software Heritage:** `https://archive.softwareheritage.org/swh:1:dir:903c8f7ea46032959ab1230c1b1ac1472cfd6068`

A.2. Publication List of Hermann Kroll

1. **Hermann Kroll**, Denis Nagel, and Wolf-Tilo Balke. "BAFREC: Balancing Frequency and Rarity for Entity Characterization in Open Linked Data". 1st International Workshop on Entity REtrieval (EYRE) at the ACM International Conference on Information and Knowledge Management (CIKM), Turin, Italy, 2018. URL: http://ws.nju.edu.cn/conf/eyre2018/paper_14.pdf

2. Stephan Mennicke, Jan-Christoph Kalo, Denis Nagel, **Hermann Kroll**, and Wolf-Tilo Balke. "Fast Dual Simulation Processing of Graph Database Queries". IEEE International Conference on Data Engineering (ICDE), Macau, China, 2019, IEEE. DOI: https://doi.org/10.1109/ICDE.2019.00030

3. Kristof Keßler, **Hermann Kroll**, Janus Wawrzinek, Christina Draheim, Stefan Wulle, Katrin Stump, and Wolf-Tilo Balke. "PubPharm – Gemeinsam von der informationswissenschaftlichen Grundlagenforschung zum nachhaltigen Service". ABI Technik 2019. DOI: https://doi.org/10.1515/abitech-2019-4005

4. Katharina Ostaszewski, Philip Heinisch, Ingo Richter, **Hermann Kroll**, Wolf-Tilo Balke, Diego Fraga, and Karl-Heinz Glaßmeier. "Pattern recognition in time series for space missions: A rosetta magnetic field case study". Acta Astronautica 2020. DOI: https://doi.org/10.1016/j.actaastro.2019.11.037

5. **Hermann Kroll**, Jan Pirklbauer, Johannes Ruthmann, and Wolf-Tilo Balke. "A Semantically Enriched Dataset based on Biomedical NER for the COVID19 Open Research Dataset Challenge", 2020. arXiv: https://doi.org/10.48550/arXiv.2005.08823

6. **Hermann Kroll**, Jan-Christoph Kalo, Denis Nagel, Stephan Mennicke, and Wolf-Tilo Balke. "Context-Compatible Information Fusion for Scientific Knowledge Graphs". International Conference on Theory and Practice of Digital Libraries (TPDL), Lyon, France, 2020, Springer. DOI: https://doi.org/10.1007/978-3-030-54956-5_3

7. **Hermann Kroll**, Denis Nagel, and Wolf-Tilo Balke. "Modeling Narrative Structures in Logical Overlays on top of Knowledge Repositories". International Conference on Conceptual Modeling (ER), Vienna, Austria, 2020, Springer. DOI: https://doi.org/10.1007/978-3-030-62522-1_18

8. **Hermann Kroll**, Denis Nagel, Morris Kunz, and Wolf-Tilo Balke. "Demonstrating Narrative Bindings: Linking Discourses to Knowledge Repositories". Workshop on Narrative Extraction From Texts (Text2Story) at the European Conference on Information Retrieval (ECIR), Lucca, Italy, 2021, CEUR-WS. DOI: https://ceur-ws.org/Vol-2860/paper7.pdf

9. **Hermann Kroll**, Jan Pirklbauer, and Wolf-Tilo Balke. "A Toolbox for the Nearly-Unsupervised Construction of Digital Library Knowledge Graphs". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Urbana-Champaign, IL, USA, 2021, IEEE. DOI: https://doi.org/10.1109/JCDL52503.2021.00014

10. **Hermann Kroll**, Judy Al-Chaar, and Wolf-Tilo Balke. "Open Information Extraction in Digital Libraries: Current Challenges and Open Research Questions". Workshop on Digital Infrastructures for Scholarly Content Objects (DISCO) at the ACM/IEEE Joint Conference on Digital Libraries (JCDL), Urbana-Champaign, IL, USA, 2021, CEUR-WS. DOI: http://ceur-ws.org/Vol-2976/short-1.pdf

11. **Hermann Kroll**, and Christina Draheim. "Narrative Information Access for a Precise and Structured Literature Search". O-Bib. Das Offene Bibliotheksjournal 2021. DOI: https://doi.org/10.5282/o-bib/5730

12. **Hermann Kroll**, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. "Narrative Query Graphs for Entity-Interaction-Aware Document Retrieval". International Conference on Asian Digital Libraries (ICADL), Online, 2021, Springer. DOI: https://doi.org/10.1007/978-3-030-91669-5_7

13. **Hermann Kroll**, Florian Plötzky, Jan Pirklbauer, and Wolf-Tilo Balke. "What a Publication Tells You — Benefits of Narrative Information Access in Digital Libraries". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Cologne, Germany, 2022, ACM. DOI: https://doi.org/10.1145/3529372.3530928

14. **Hermann Kroll**, Jan Pirklbauer, Florian Plötzky, and Wolf-Tilo Balke. "A Library Perspective on Nearly-Unsupervised Information Extraction Workflows in Digital Libraries". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Cologne, Germany, 2022, ACM. DOI: https://doi.org/10.1145/3529372.3530924

15. **Hermann Kroll**, and Wolf-Tilo Balke. "On Design Principles for Narrative Information Systems". Workshop on Semantic Techniques for Narrative-Based Understanding (SEM4NBU) at the International Joint Conference on Artificial Intelligence and the European Conference on Artificial Intelligence (IJCAI-ECAI), Vienna, Austria, 2022, CEUR-WS. DOI: https://ceur-ws.org/Vol-3322/short3.pdf

16. **Hermann Kroll**, Niklas Mainzer and, Wolf-Tilo Balke. "On Dimensions of Plausibility for Narrative Information Access to Digital Libraries". International Conference on Theory and Practice of Digital Libraries (TPDL), Padua, Italy, 2022, Springer. DOI: https://doi.org/10.1007/978-3-031-16802-4_43

17. Christina Draheim, **Hermann Kroll**, and Stefan Wulle. "Neue PubPharm-Tools - Gezielter suchen, Überblick gewinnen". Krankenhauspharmazie 2023. URL: https://www.krankenhauspharmazie.de/heftarchiv/2023/01/neue-pubpharm-tools-gezielter-suchen-ueberblick-gewinnen-1.html

18. Niklas Kiehne, **Hermann Kroll**, and Wolf-Tilo Balke. "Contextualizing Language Models for Norms Diverging from Social Majority". Findings of the Association for Computational Linguistics: EMNLP, Abu Dhabi, United Arab Emirates, 2022, ACL. DOI: http://dx.doi.org/10.18653/v1/2022.findings-emnlp.339

19. **Hermann Kroll**, and Wolf-Tilo Balke. "Are Qualifiers Enough? Context-Compatible Information Fusion for Wikimedia Data". Wiki Workshop 2023, Online. DOI: https://wikiworkshop.org/2023/papers/WikiWorkshop2023_paper_26.pdf

20. **Hermann Kroll**, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. "A discovery system for narrative query graphs: entity-interaction-aware document retrieval". International Journal on Digital Libraries (IJDL) 2023. DOI: https://doi.org/10.1007/s00799-023-00356-3

21. **Hermann Kroll**, Jan Pirklbauer, Florian Plötzky, and Wolf-Tilo Balke. "A detailed library perspective on nearly unsupervised information extraction workflows in digital libraries". International Journal on Digital Libraries (IJDL) 2023. DOI: https://doi.org/10.1007/s00799-023-00368-z

22. **Hermann Kroll**, Christin Katharina Kreutz, Pascal Sackhoff, and Wolf-Tilo Balke. "Enriching Simple Keyword Queries for Domain-Aware Narrative Retrieval". ACM/IEEE Joint Conference on Digital Libraries (JCDL) Santa Fe, NM, USA, 2023, IEEE. DOI: https://doi.org/10.1109/JCDL57899.2023.00029 arXiv: https://doi.org/10.48550/arXiv.2304.07604

23. **Hermann Kroll**, Christin Katharina Kreutz, Mirjam Cuper, Bill Matthias Thang, and Wolf-Tilo Balke. "Aspect-Driven Structuring of Historical Dutch Newspaper Archives". International Conference on Theory and Practice of Digital Libraries (TPDL), Zadar, Croatia, 2023, Springer. DOI: https://doi.org/10.1007/978-3-031-43849-3_4 arXiv: https://doi.org/10.48550/arXiv.2307.09203

24. **Hermann Kroll**, Julian Schenke, Florian Plötzky, and Wolf-Tilo Balke. "Narrativer Informationszugriff Interdisziplinär – Chancen und Herausforderungen für Fachinformationsdienste". O-Bib. Das Offene Bibliotheksjournal, 2023. DOI: https://doi.org/10.5282/o-bib/5962

25. **Hermann Kroll**, Katharina Heldt, and Lisa Kühnel. "Innovative Recherchetools für das Screening von Literatur zu Long COVID: Eine kooperative Zusammenarbeit zwischen RKI, ZB MED und PubPharm". GMS Medizin - Bibliothek - Information, 2023. DOI: https://doi.org/10.3205/mbi000558

B. Full-texts of Publications

The full-texts of the contributing papers are attached below. They are ordered by their publication date in ascending order. The order is:

[40] **Hermann Kroll**, Jan-Christoph Kalo, Denis Nagel, Stephan Mennicke, and Wolf-Tilo Balke. "Context-Compatible Information Fusion for Scientific Knowledge Graphs". International Conference on Theory and Practice of Digital Libraries (TPDL), Lyon, France, 2020, Springer. DOI: https://doi.org/10.1007/978-3-030-54956-5_3

[36] **Hermann Kroll**, Jan Pirklbauer, and Wolf-Tilo Balke. "A Toolbox for the Nearly-Unsupervised Construction of Digital Library Knowledge Graphs". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Urbana-Champaign, IL, USA, 2021, IEEE. DOI: https://doi.org/10.1109/JCDL52503.2021.00014

[42] **Hermann Kroll**, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. "Narrative Query Graphs for Entity-Interaction-Aware Document Retrieval". International Conference on Asian Digital Libraries (ICADL), Online, 2021, Springer. DOI: https://doi.org/10.1007/978-3-030-91669-5_7

[43] **Hermann Kroll**, Florian Plötzky, Jan Pirklbauer, and Wolf-Tilo Balke. "What a Publication Tells You – Benefits of Narrative Information Access in Digital Libraries". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Cologne, Germany, 2022, ACM. DOI: https://doi.org/10.1145/3529372.3530928

[39] **Hermann Kroll**, Jan Pirklbauer, Florian Plötzky, and Wolf-Tilo Balke. "A Library Perspective on Nearly-Unsupervised Information Extraction Workflows in Digital Libraries". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Cologne, Germany, 2022, ACM. DOI: https://doi.org/10.1145/3529372.3530924

[38] **Hermann Kroll**, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. "A discovery system for narrative query graphs: entity-interaction-aware document retrieval". International Journal on Digital Libraries (IJDL) 2023. DOI: https://doi.org/10.1007/s00799-023-00356-3

[37] **Hermann Kroll**, Jan Pirklbauer, Florian Plötzky, and Wolf-Tilo Balke. "A detailed library perspective on nearly unsupervised information extraction workflows in digital libraries". International Journal on Digital Libraries (IJDL) 2023. DOI: https://doi.org/10.1007/s00799-023-00368-z

[41] **Hermann Kroll**, Christin Katharina Kreutz, Pascal Sackhoff, and Wolf-Tilo Balke. "Enriching Simple Keyword Queries for Domain-Aware Narrative Retrieval". ACM/IEEE Joint Conference on Digital Libraries (JCDL) Santa Fe, NM, USA, 2023, IEEE. DOI: https://doi.org/10.1109/JCDL57899.2023.00029 arXiv: https://doi.org/10.48550/arXiv.2304.07604

B.1. TPDL 2020: Context-Compatible Information Fusion for Scientific Knowledge Graphs

TPDL'20

Hermann Kroll, Jan-Christoph Kalo, Denis Nagel, Stephan Mennicke, and Wolf-Tilo Balke. "Context-Compatible Information Fusion for Scientific Knowledge Graphs". International Conference on Theory and Practice of Digital Libraries (TPDL), Lyon, France, 2020, Springer. DOI: https://doi.org/10.1007/978-3-030-54956-5_3

Context-Compatible Information Fusion for Scientific Knowledge Graphs

Hermann Kroll, Jan-Christoph Kalo, Denis Nagel, Stephan Mennicke, and Wolf-Tilo Balke

Institute for Information Systems, TU Braunschweig, Braunschweig, Germany
{kroll,kalo,mennicke,nagel,balke}@ifis.cs.tu-bs.de

Abstract. Currently, a trend to augment document collections with entity-centric knowledge provided by knowledge graphs is clearly visible, especially in scientific digital libraries. Entity facts are either manually curated, or for higher scalability automatically harvested from large volumes of text documents. The often claimed benefit is that a collection-wide fact extraction combines information from huge numbers of documents into one single database. However, even if the extraction process would be 100% correct, the promise of pervasive information fusion within retrieval tasks poses serious threats with respect to the results' validity. This is because important contextual information provided by each document is often lost in the process and cannot be readily restored at retrieval time. In this paper, we quantify the consequences of uncontrolled knowledge graph evolution in real-world scientific libraries using NLM's PubMed corpus vs. the SemMedDB knowledge base. Moreover, we operationalise the notion of *implicit context* as a viable solution to gain a sense of *context compatibility* for all extracted facts based on the pair-wise coherence of all documents used for extraction: Our derived measures for context compatibility determine which facts are relatively safe to combine. Moreover, they allow to balance between precision and recall. Our practical experiments extensively evaluate context compatibility based on implicit contexts for typical digital library tasks. The results show that our implicit notion of context compatibility is superior to existing methods in terms of both, simplicity and retrieval quality.

Keywords: Implicit Context · Knowledge Graph · Digital Libraries

1 Introduction

Knowledge graphs have revolutionised the access to entity-centric information on the Web, with *Google's knowledge graph*[1] and the *Wikidata knowledge base* [19] being prime examples. One reason is that the old 'Web of Documents' is more and more turning into a 'Web of Linked Data', which needs new access methods beyond IR-style keyword search: entity-centric information needs to be structured, disambiguated, and semantically enriched by information from various

[1] https://developers.google.com/knowledge-graph/

sources. Thus, also in the well-curated domains of digital libraries, a trend to augment document collections to semantically enriched content bases is clearly visible. Especially in scientific libraries *Big Scholarly Data* in heterogeneous form (see [21] for a good overview) is exploited for value-adding services, such as related work recommendation, expert search, or information enhancement using specialised entity-centric databases, like *DrugBank*[2] or *UniProt*[3]. The ultimate vision currently is to extract facts from complete digital collections into one comprehensive knowledge graph for science, supporting complex information needs and offering a variety of additional services, see e. g. [1, 7, 18].

Yet, the question whether a document collection may still offer more than a collection of extracted facts was already raised at an early stage. An obvious problem concerns the *trustworthiness* of sources: there is a long-standing discussion about the actual truth or plausibility of extracted facts and how well they match with facts extracted from other sources [14]. Thus, keeping lineage or provenance information and respective reputation scores as metadata for each fact is vital [2]. A second class of problems is created by errors in the *algorithmic processes* necessary for fact extraction from natural language texts, covering entity recognition, disambiguation and linking, as well as reliable relation extraction, see e. g. [15]. In fact, all tasks in this process are still error-prone, and even small errors may quickly spoil the overall quality in knowledge graphs [10].

However, even if all these problems were solved, there would be still a major, yet rarely discussed issue: the general *validity* of facts. With respect to general fact validity, current knowledge graphs on the Web vastly differ from those used in scientific digital libraries. Whereas entity-centric data in typical Linked Open Data sources on the Web may or may not be correct, it still tends to be *generally* valid, as e. g. the *birthdate of a person* or *which actors played in some movie*. In contrast, entity-centric data reported in scientific digital collections is often more problematic. Consider for instance different medical treatment options with some active ingredient. They depend on many caveats: general concerns, unresolved discourses in the community, the specific disposition of an actual patient, etc. Another prime examples are clinical trials: even if they are methodically sound, their results can only be considered valid *within the limited context* investigated by each trial. Thus, given the problems to properly control studies currently the generalisability of facts extracted from clinical trials is difficult to assess.

Assume we extract the fact (`simvastatin`, `causes`, `rhabdomyolysis`) from some document reporting on a simultaneous treatment of patients with simvastatin and amiodarone. As the resulting interaction indeed may lead to rhabdomyolysis as a side effect, the information is correct. In the same fashion, we may correctly extract the fact (`simvastatin`, `treats`, `arteriosclerosis`) from some other document on treatment options for arteriosclerosis. But if we now use the combined knowledge graph to query the side effects of *simvastatin in treating arteriosclerosis*, we run into trouble: the fact that *simvastatin causes rhabdomyolysis* is not valid *in general*. It is only valid *within the context of si-*

[2] https://www.drugbank.ca
[3] https://www.uniprot.org

multaneous treatment with simvastatin and amiodarone. Thus, without having facts restricted by their exact context, a free combination with other facts from the knowledge graph may at least be questionable, if not plain false. Yet, current extraction procedures do exactly this: after long years of standardisation, knowledge graphs typically store facts as simple RDF-triples [3]. This way, tearing facts out of documents and putting them into a knowledge graph means losing all contextual information. If such knowledge graphs are later used for tasks like knowledge discovery, question answering and querying, serious errors can be foreseen. The central question in designing knowledge graphs for digital libraries is thus: *How can knowledge graphs maintain a sense of context for their individual collection of facts?* And concerning later applications: *How can we combine individual facts or even completely merge fact collections while still maintaining their contexts?*

When working with RDF-triples, the *technical* solution for adding context information mostly relies on reification of triples. But how is the correct context for each fact determined? To overcome this problem, two approaches are common: 1. In the community project Wikidata, uploaders are also responsible for supplying all necessary contextual information as additional triples, called qualifiers [19]. 2. In cases where clear-cut contexts can a-priori be determined for some field, the direct modelling and extraction of n-ary relations from document collection are possible [6].

Yet, in both cases, the context needs to be modelled *explicitly*. In this paper, we harness valuable work in the digital library community on standardising provenance and bibliographic metadata (such as authors or keywords) to derive a novel *implicit*, i. e. document-based context model for knowledge graphs. Documents like scientific papers interweave facts in complex contexts and can be assumed to be intrinsically coherent, e. g. by describing all relevant assumptions, methods, observations and conclusions. Thus, for all facts our model takes advantage of the respective extraction documents' characteristics and uses them as an implicit context for facts. Such implicit contexts ensure that given a retrieval problem, only facts from a coherent group of documents can be combined to produce a valid result. Indeed, our experiments show that restricting the information fusion process of knowledge graphs to (restricted) document contexts has a high impact on the number and quality of possible candidates. In addition to structural requirements (graph matching), we consider the context approximated by documents sharing different characteristics to produce valid answers to a query. To improve the result quality for any given query, we operationalise and analyse metrics to find documents having **compatible** contexts. A context compatible set of documents can then be used to obtain better results in terms of validity for tasks like knowledge discovery and querying. We analyse our document-based implicit context model in Sect. 3 and provide a detailed experimental analysis in Sect. 4. Our contributions are:

1. We design and discuss a novel implicit context model suitable for digital libraries. We demonstrate the superiority of implicitly capturing contexts for a real-world knowledge graph in the medical domain.

2. Further, we introduce the concept of context compatibility, i. e. we extend strict document contexts to compatible contexts, increasing the recall for practical applications.
3. We publish all of our scripts as well as evaluation data and results in a publicly available GitHub repository[4] for reproducibility.

2 Related Work

Literature-based Discovery is a well-known and highly discussed topic, i. e. inferring new knowledge based on the current state of literature [16]. In this work, we focus on the application of scientific knowledge graphs for digital libraries. Contextualisation of data can be realised by adding additional contextual information to an individual statement or fact. Regarding RDF, this means to incorporate triples into the knowledge graphs that capture information about a specific triple already existent in the data. Ideas on how to represent contextual information in RDF are provided in [13]. This process is called reification of RDF data [8]. It is realised by introducing a new resource, referencing the reified triple in other statements.

Qualifiers for Contextualising Knowledge Wikidata, the most extensive open knowledge base on the Web, tries to reify pure RDF facts by using so-called *qualifiers* [19]. Qualifiers add information to a fact by appending a property-value pair directly to it. An example fact (`simvastatin`, `causes`, `rhabdomyolysis`) may further be described by an additional qualifier, namely `when simultaneously used with` along with the respective value `amiodarone`. The qualifiers claim that *simvastatin causes rhabdomyolysis* only, in a simultaneous treatment with *simvastatin* and *amiodarone*. Thus, qualifiers may be used to add additional provenance and sometimes contextual information to simple RDF facts [9]. Even though Wikidata comprises around 30 million qualifier statements (10-2018), they are hardly used to express context for scientific facts, i. e. drug-disease treatments. Even more, only about 5% of all statements are qualifiers (573 million statements). Qualifiers are often restricting the statement they are referring to in a temporal manner, e. g. using the `start time` qualifier. Besides, they may add some provenance information such as references or citations to the statements. In other cases they state information that has no impact on the validity of the fact in question, e. g. the `determination method` is simply used with qualifier values like *chronometry* or *questionnaire* without affecting the validity of its fact. Using qualifiers in joining facts has no precise semantics, e.g. how can we decide whether two qualifiers describe the same context? The curation of explicit contexts is a huge task and moreover, working with explicit context models in practice is unclear.

N-ary Fact Extraction An extension of extracting binary facts is to harvest n-ary facts [6]. In a large-scale experiment, the authors prove that n-ary facts are more

[4] https://github.com/HermannKroll/ContextInformationFusion

precise than just using binary facts [6]. Thereby, it is possible to explicitly extract and store the context of relations in a higher level relation. For our previous drug and side effect scenario, we may easily design a ternary relation capturing drug, the cause as well as the interacting drug: $causes \subseteq drug \times sideeffect \times interacting\ drug$. However, how good is n-ary fact extraction in practice? Ernst et al. extracted the relation *AthleteWonAward* from a news corpus consisting of 2.8 million documents with about 112 million sentences [6]. They mined 3804 binary, 1089 ternary, 224 4-ary, 23 5-ary and two 6-ary instances of this relation with their best configuration regarding precision. Even though n-ary facts are a promising idea to capture the context of facts, obtaining such n-ary facts is a difficult task, because it requires manually defining the context for every single relation by defining its arity, its domains and its semantics upfront. This is a very strong restriction because considering any possible context of some relation a priori is close to impossible.

Provenance Another understanding of contexts is provenance, which mainly focuses on storing information attached to the actual fact [17]. The scope of provenance thereby ranges from storing only the explicit source document over additionally storing information related to its creation process such as the author or release date [20]. Provenance can then help to argue about the quality and trustworthiness of the statement in question. Provenance can be integrated into knowledge graphs by using Named Graphs [5]. These are linked to individual facts by extending RDF triples to form N-Quads [4]. In the last years, much work was spent on developing the so-called Prov-O Ontology Description [12]. Prov-O enables knowledge graph designers to encode and store arbitrary information, such as context, for knowledge graph facts. Unfortunately, Prov-O requires users to spend much work on manually providing this additional information, i. e. Prov-O comes with a similar problem as qualifiers in Wikidata. There is yet no solution to automatically reuse context information in the fusion process of knowledge graphs. As far as we know, there exists no practical evaluation of using contexts in typical knowledge graph tasks. With the introduction of our document-based implicit context model and evaluation on a real-world scenario, we extend the current state of literature by giving a practical solution to retain context for digital libraries. Therefore, already applied techniques like Prov-O, Named Graphs, as well as reification, may simply be used as an implementation providing the necessary context in the form of document references for our implicit context model.

3 Implicit Context

Instead of modelling contexts explicitly, textual documents (i. e. research papers) serve as contexts for knowledge graph facts. A scientific publication interweaves facts in assumptions, methods, observations and conclusions. Thus, the argumentative story of a scientific document provides all relevant context variables implicitly, validating its contained facts. We assume scientific documents to come

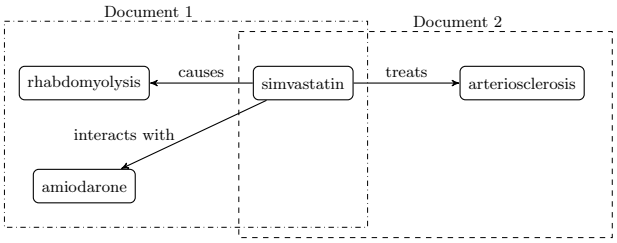


Fig. 1: Implicit Context Representation for a Knowledge Graph

with a single context, e. g. clinical trials analyse drugs under stable conditions. Indeed, surveys and scientific papers might include several contexts, e. g. describing related work. For this paper, we assume that scientific knowledge graphs should be built by extracting facts out of the paper's main argumentation, i. e. skipping sections such as related work in the extraction process. For our running example, the document provides vital information that simvastatin only causes rhabdomyolysis, when the person is simultaneously treated with amiodarone. Here, the document itself implicitly defines and, thereby, determines the context of interest, because we assume the extracted facts to participate in the main argumentation of the paper. If we mine facts from a single document, then all extracted facts from this document naturally share the same context. The information fusion process by combining/joining facts from the same document to answer a query automatically leads to valid facts because they stem from the same context. In the scientific domain, this context often boils down to conclusions being observed under the same experimental conditions. Therefore, returning to our running example, we define the implicit context of a fact as the document it stems from, see Fig. 1 as an example.

When using a **strict implicit context**, we restrict the combination of facts to those facts within the same **context**, i. e. to facts extracted from the exact same document. Applied to our example, we obtain either that simvastatin treats arteriosclerosis, or that simvastatin causes rhabdomyolysis. We would not obtain the wrong side effect rhabdomyolysis in an arteriosclerosis treatment because there is not a single document validating it.

3.1 Context Compatibility

Obviously, restricting the fusion process of knowledge graphs to strict implicit context will have a substantial impact on the number of obtained results, because we combine facts stemming from the same document only. In addition to strict implicit contexts, we may assume that two scientific documents on simvastatin share the same context, e. g. they describe clinical trials analysing an arteriosclerosis treatment using simvastatin. Since both papers are clinical trials with the same experimental conditions, it seems promising that a combination of facts from both documents leads to valid query results. Hence, inferring new

knowledge between different documents may also be possible. Our idea extends the restriction on pure document contexts to context compatibility ranging over sets of documents. This will lead to broader contexts and allows for a less restrictive combination of facts. Two documents d_1 and d_2, sharing the same context in the above-mentioned sense, will be denoted as **context compatible**: $d_1 \sim d_2$. Thereby, we require $\sim$ to be a reflexive binary relation over the document collection, i. e. one document is always compatible with itself. Combining facts from different but context compatible documents shall yield valid query results.

Comparing the contexts spanned by two or more documents directly is a tedious and time-consuming task that requires a deep understanding of documents' domains. Here, we use different metrics to approximate the context compatibility of documents. In digital libraries, a collection of documents typically provides valuable metadata information. Subsequently, we design two different kinds of similarity metrics to assess the context compatibility of documents: 1. metrics, which directly work on metadata information like authors and curated keywords, and 2. metrics, which build upon textual similarities for titles and abstracts. We choose a threshold-based classification approach to estimate whether two documents are context compatible or not. If the similarity value, computed by a metric, between two documents is above a threshold t, we assume the documents to have a compatible context. Thus, we can safely fuse the facts of two context compatible documents to form a valid answer.

Definition 1. *Let sim be a similarity metric between documents and $t \in \mathcal{R}$ a threshold value. Two documents d_1 and d_2 are context compatible, denoted by $d_1 \sim d_2$, if $sim(d_1, d_2) \geq t$.*

Metadata-based Similarity Metrics In scientific contexts, researchers typically work on a specific research field, e. g. a group of medical experts are researching *drug interactions with simvastatin*. They might write several publications about their findings based on similar assumptions like *experimental conditions*. Thus, we assume papers, written by the same authors, to have compatible contexts. We formulate the first metric sim_{author} to estimate context compatibility by using the Jaccard coefficient between the authors of documents. Since contexts of facts should be compatible, if they comprise similar assumptions or experimental designs, we try to capture this intuition by relying on the valuable manually curated metadata available for medical documents. In PubMed, documents are annotated with manual curated mesh headings and chemicals. A mesh heading is a mesh term describing medical entities, actors, processes and concepts like *humans*, *pain*, *trial* and *simvastatin*. The mesh headings, therefore, might capture the context that is given by a document. The second metric sim_{mesh} is defined as the Jaccard coefficient of the documents' mesh headings. Similarly to the mesh terms, we use the chemicals annotated to documents as an approximation for context compatibility. Therefore, $sim_{chemical}$ is defined as the Jaccard coefficient of the documents' chemicals.

Text-based Similarity Metrics In addition to the metadata-based approaches, we also try to capture the context compatibility by measuring textual similarities

among the documents' texts. Here, sim_{title} is defined as the Jaccard coefficient between the titles of two documents to estimate the text-similarity between documents. The previous similarity metrics can only be applied to pairs of documents for determining context compatibility. To further extend fact fusions to more than a pair of documents, we suggest to also directly determine the compatibility between multiple documents by clustering documents into **context compatible sets** such that all documents inside such a set are pairwise context compatible. Given the respective documents the facts in the knowledge graph stem from, we use a clustering method to produce groups of documents with compatible contexts. Here, we use textual information, i. e. titles and abstracts of documents. We select a common method to cluster documents to understand whether compatible document sets are helpful: 1. We extract the titles and abstracts of documents. Thereby, we remove stop words and apply stemming. 2. We compute the TF-IDF matrix upon the texts. Words which occur very frequently or words which occur very rarely are removed. 3. Clustering documents with various texts requires much computational power. Thus, we use a principal component analysis (PCA) to reduce the number of dimensions to 300. 4. Finally, we apply a k-means++ clustering on the reduced matrix with different k values.

4 Analysis on SemMedDB

In the following experiments, we evaluate whether restricting fact combinations to their document contexts is capable of producing valid facts for typical medical queries. We perform a comparison to querying a knowledge graph without contextual information, allowing us to join arbitrary facts. In our expectations, using implicit context should increase the quality of query results substantially, while reducing the overall number of results. For the evaluation, we compare the number and quality of results for typical queries on a large medical knowledge graph called *SemMedDB* by using no context as a baseline and our implicit context models.

SemMedDB is a fact-based database consisting of medical entities and relations between them [11]. A fact mining process automatically extracted all facts from abstracts and titles of documents in PubMed. For each extracted fact in *SemMedDB*, a reference to its source document is retained. Hence, *SemMedDB* provides provenance information. We use *SemMedDB* 2019[5] in version *semmedVER40R*. This version comprises 20,124,700 distinct facts extracted 97,972,561 times. We design three experiments to compare the usage of *SemMedDB* as a knowledge graph without context on the one hand and with implicit context on the other. The experiments are built on three scientific queries, and are also depicted in Fig. 2: 1. Knowledge discovery via querying using the `causes` relation, 2. Predicting drug-drug interactions via a gene (like already performed by domain experts [22]) and 3. Predicting drug-drug interactions via a biological function (like already performed by domain experts [22]).

[5] https://skr3.nlm.nih.gov/SemMedDB/

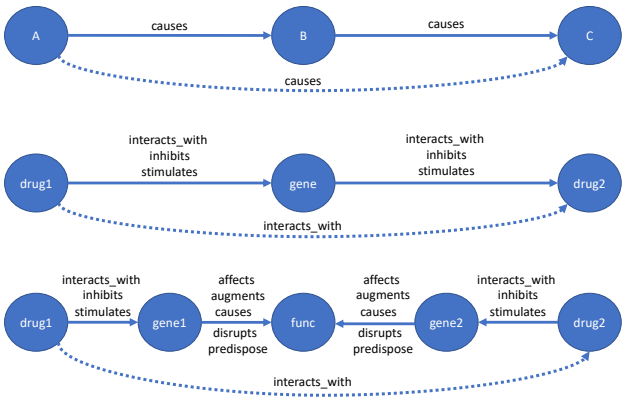


Fig. 2: Graph Patterns to Derive New Facts in SemMedDB. The Dotted Edge Depicts the New Derived Fact

Transitive Causal Relation (Causes) `Causes` is used to express a relation between a cause and an effect of medical concepts, e. g. a drug and a disease. Since this relation is usually assumed to be transitive, the goal in this knowledge discovery task is to query for new facts by joining two existing causal facts from the knowledge graph. As an example, the facts (`simvastatin`, `causes`, `risk of heart disease`) and (`risk of heart disease`, `causes`, `heart failure`) may be joined to obtain the new fact `(simvastatin, causes, heart failure)`. To increase the quality of these facts, we select only facts appearing in at least three documents, yielding 153,024 distinct facts extracted 1,584,676 times from documents.

Predicting drug-drug interactions (DDI) In a second experiment, we rely on a known approach for finding drug-drug interactions using *SemMedDB* [22]. Such an interaction may cause several side effects in a patient's treatment. Thus, finding these new interactions is a relevant task for medical experts that can be easily supported by knowledge graphs. Drug-drug interactions are discovered using two queries as described in [22]. We call these interactions *DDI-G*, a drug-drug interaction via a gene and DDI-F, a drug-drug interaction via a function.

Estimating the Result Quality To be able to perform the evaluation, we take *SemMedDB* as the gold standard of medical knowledge and assume that it is 100% correct and also complete. As far as we know, there is no medical source comprising more medical domain knowledge than *SemMedDB*. *SemMedDB* contains a dedicated `causes` predicate and `interacts with` predicate between drugs. Thus, we count how many derived facts are contained in *SemMedDB* already and how many of them are correct. To estimate the recall, we take the number of query answers on the knowledge graph without restricting fact combinations as an overestimation of the number of all correct results. Thereby, we

Table 1: Number and Quality of Newly Distinct Obtained Facts by Querying a Knowledge Graph without Context and with Strict Implicit Context

Graph	#Obtained Facts	#Correct	Precision	Recall
Knowledge Graph (Causes)	7,978,099	95,037	1.19%	**100%**
Strict Implicit Context (Causes)	11,478	5,544	**48.3%**	5.83%
Knowledge Graph (DDI-G)	753,899	55,370	7.34%	**100%**
Strict Implicit Context (DDI-G)	1,311	909	**69.3%**	1.64%
Knowledge Graph (DDI-F)	18,685,416	148,346	0.79%	**100%**
Strict Implicit Context (DDI-F)	2,138	1,352	**63.2%**	0.9%

overestimate the recall of the knowledge graph as being 100% and compare the remaining approach to that number. We underestimate the precision, because there may exist correctly derived facts, which are not included in our ground truth (the knowledge graph itself).

4.1 Strict Implicit Context

For the knowledge graph query experiments, we have no restrictions when joining facts and just perform a simple pattern matching from the query to the knowledge graph. In contrast, when using strict implicit context, we restrict fact combinations to the document contexts, i. e. combinations of facts are only possible within the context of a document. The number and quality of obtained results by using no context in comparison to using strict implicit context for all three tasks (causes, DDI-G and DDI-F) are listed in Table 1. The number of facts obtained from the baseline, a knowledge graph without context, differs by orders of magnitude compared to the knowledge graph with strict implicit context in all three experiments. However, the results only come with a precision of 1.19% (causes), 7.34% (DDI-G) and 0.79% (DDI-F) by using no context and 48.3% (causes), 69.3% (DDI-G) and 63.2% (DDI-F) by using strict implicit context. The recall decreases from 100% to 5.83% (causes), 1.64% (DDI-G) and 0.9% (DDI-F).

Discussion In sum, using strict implicit document-based contexts outperforms the plain knowledge graph (no context) approach for all three experiments with regard to the precision. However, strict implicit context restricts the derivation process of facts to single document contexts, and thus a considerable amount of incorrect, but also some correct results are not returned. This leads to a lower recall in comparison to joining arbitrary facts. When querying a knowledge graph, a high degree of correctness is often needed. Particularly if medical experts need to verify drug-drug interactions in studies, high-quality results are desired.

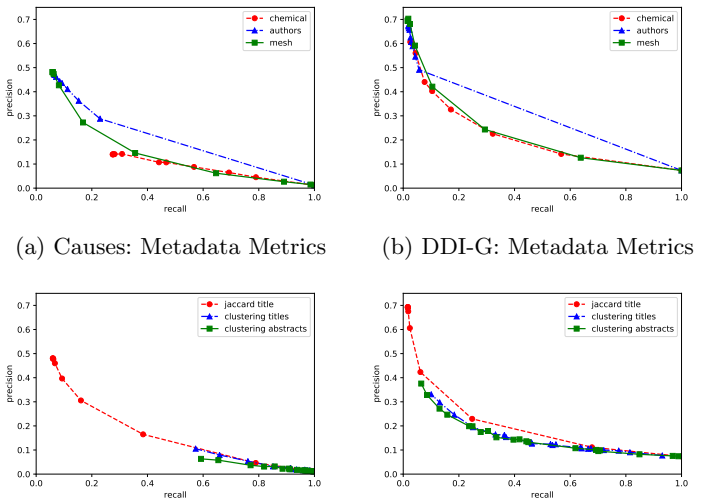
4.2 Context Compatibility

We design context compatibility to increase the recall for different tasks in comparison to strict implicit context by allowing the fusion of facts stemming from compatible document contexts. Our evaluation comprises six different approaches for context compatibility on two different medical queries. Three of the approaches work purely on the metadata (i. e. chemical, mesh headings and authors) and three approaches work with textual measures (i. e. Jaccard coefficient between titles, clustering of titles and abstracts). The two queries are the causes query from Fig. 2 at the top and the DDI-G query depicted in Fig. 2 in the middle. Unfortunately, we have to skip the third experiment (DDI-F) here due to performance issues. In the DDI-F experiment, the knowledge graph produces around 18 million facts. Checking the context compatibility between documents, validating a fact derivation, leads to too many different combinations. For all our experiments, we evaluate different thresholds and k-values to report our findings as precision-recall curves. We check different thresholds (0 to 1.0 by a step size of 0.1) and 20 different k values ranging from 2 to 100,000. Additionally to the results presented in this paper, more experimental results can be found on our GitHub repository. To perform our experiments, we have accessed the metadata and texts of PubMed documents by downloading the latest version of the PubMed Medline 2019 as an XML dump[6], which provides title, abstracts and valuable metadata.

Causes Experiment Fig. 3 (a) depicts the precision-recall curve for the cause experiment using metadata similarity metrics. Note that selecting a threshold of 0.0 leads to the same result as using the knowledge graph approach without contextual restrictions and 1.0 leads to similar results as using strict implicit context. We achieve the best possible precision of about 48% with a recall of about 6% by using a threshold of 1.0 for sim_{mesh} and $sim_{authors}$. A higher recall is achieved when using $sim_{chemicals}$ because 53% of all documents provide curated chemicals, whereas the other metadata is less common. We obtain the best F1-Score of 25.5% (28.8% precision and 23% recall) for $sim_{authors}$ with a threshold of 0.1. Although sim_{author} outperforms the other metrics regarding precision and recall, sim_{author} provides only a small recall range. 9 of 10 thresholds for sim_{author} yield a recall below 23% and the last threshold yields 100% recall. Computing more fine-grained thresholds would not help here, because most of the papers have only a few authors yielding a small range of different Jaccard coefficients.

The results of our text-based approaches for context compatibility are depicted in Fig. 3 (c). Here, the clustering methods on titles and abstracts share a similar shape; hence they have a comparable performance. Variations of the number of clusters can cover a range of recall values between 0.6 and 1.0 while keeping an acceptable precision of around 10%. Hence, the methods can boost the precision of the knowledge graph 10-fold, while only sacrificing around 40%

[6] https://www.nlm.nih.gov/databases/download/pubmed_medline.html



(a) Causes: Metadata Metrics (b) DDI-G: Metadata Metrics

(c) Causes: Text-based Metrics (d) DDI-G: Text-based Metrics

Fig. 3: Precision-Recall Curve of the Experiments (Causes and DDI-G) by using Different Metrics to Estimate the Context Compatibility Between Documents

of recall. In contrast, the Jaccard-based similarity sim_{title} outperforms the clustering methods (denoted as jaccard title in the plot). The approach achieves a comparable precision for high recall values. Besides, it is possible to achieve even higher precision, for sacrificing some correct results at lower recall values by achieving a precision of almost 50% at a recall of 10%.

Overall, we can summarise that sim_{author} and sim_{title} achieve the best results for the causes experiment. While sim_{author} performs better regarding precision, sim_{title} offers to select a broader range of recall values.

DDI Gene Experiment Fig. 3 (b) depicts the precision-recall curve for the DDI-G experiment using metadata similarity metrics. Again, $sim_{authors}$ outperforms the other metrics, e. g. selecting a threshold of 0.1 yields a precision of 49% and a recall of 6%. Compared to strict implicit context, the precision decreases from 69% to 49%, while the recall increases from 1.6% to 6%. Thereby, 9 of 10 thresholds for $sim_{authors}$ yield a recall below 6%. In this experiment, $sim_{chemical}$ performs better than in the causes experiment. We obtain the best F1-Score of 26.5% (22.6% precision and 32.1% recall) for $sim_{chemicals}$ with a threshold of 0.2. We assume that a chemical-based similarity fits best for a drug-based query.

We depict the precision-recall curve for the DDI-G experiment using text-based similarities in Fig. 3 (d). Again, the clustering methods on titles and abstracts share a similar shape. In comparison to the causes experiment, the clustering approaches provide a broader range of recall values with higher precision. The Jaccard-based similarity sim_{title} outperforms the clustering methods.

Similar to our previous experiments, all approaches boost the precision of the knowledge graph, which was around 7%, while keeping good recall values. Overall, for the DDI-G experiment, we can summarise that sim_{author} and sim_{title} achieve best results.

Discussion All techniques for context compatibility can boost the poor quality of query answers on knowledge graphs by at least one order of magnitude while being able to retain high recall. Furthermore, the techniques offer much more flexibility than the knowledge graph without context and with strict implicit context alone by providing the possibility of choosing between precision and recall, depending on the application.

5 Conclusion

In this paper, we highlighted the importance of retaining document contexts for supporting typical knowledge graph tasks for digital libraries. Indeed, document context proves crucial for proving the validity of facts, especially, in scientific domains such as biomedicine or pharmacy. Moreover, we introduced *implicit contexts* using documents as an approximation of contexts and evaluated them in combination with compatible contexts for different tasks. Our experiments show the applicability and feasibility of document-driven contextualisation for tasks like knowledge discovery and querying in practice. Approximating contexts at the document-level offers an easy-to-use and, likewise, high-quality opportunity to maintain context in knowledge graphs. Storing techniques like Prov-O, Named Graphs and N-Quads are already ready-to-use and established fact mining processes may easily be extended by maintaining a reference for each fact to its source document, but nothing more. Providing context compatibility between documents might be as simple as designing metrics for already available metadata in digital libraries. This technique leads to an apparent increase of recall when using implicit contexts, but would not deny the valuable context given by librarian documents.

As future work, we would like to investigate measures for *story-based* similarity between documents and to evaluate their usefulness for context compatibility. The *story* of a document is related to its argumentation plus their contextual settings. We believe that a story-based similarity measure would improve the previously described similarity metrics in different tasks.

References

1. Auer, S., Kovtun, V., Prinz, M., Kasprzik, A., Stocker, M., Vidal, M.E.: Towards a knowledge graph for science. In: Proc. of the 8th Int. Conf. on Web Intelligence, Mining and Semantics. WIMS '18, ACM (2018)
2. Bechhofer, S., Buchan, I., De Roure, D., Missier, P., Ainsworth, J., Bhagat, J., Couch, P., Cruickshank, D., Delderfield, M., Dunlop, I., et al.: Why linked data is not enough for scientists. Future Generation Computer Systems **29**(2), 599–611 (2013)

3. Candan, K.S., Liu, H., Suvarna, R.: Resource Description Framework : Metadata and Its Applications. SIGKDD Explorations **3**(1), 6–19 (2001)
4. Carothers, G.: RDF 1.1 N-Quads. https://www.w3.org/TR/n-quads/ (2014)
5. Carroll, J.J., Bizer, C., Hayes, P., Stickler, P.: Named graphs, provenance and trust. In: Proc. of the 14th Int. Conf. on WWW. p. 613–622. WWW '05, ACM (2005)
6. Ernst, P., Siu, A., Weikum, G.: Highlife: Higher-arity fact harvesting. In: Proc. of the 2018 World Wide Web Conf. p. 1013–1022. WWW '18, Int. World Wide Web Conf. Steering Committee (2018)
7. Fathalla, S., Vahdati, S., Auer, S., Lange, C.: Towards a Knowledge Graph Representing Research Findings by Semantifying Survey Articles. In: Int. Conf. on Theory and Practice of Digital Libraries. pp. 315–327. Springer (2017)
8. Hayes, P.J., Patel-Schneider, P.F.: RDF 1.1 Semantics. https://www.w3.org/TR/rdf11-mt/##whatnot (2014)
9. Hernández, D., Hogan, A., Krötzsch, M.: Reifying RDF: what works well with wikidata? In: Proc. of the 11th Int. Work. on Scalable Semantic Web Knowledge Base Systems. CEUR Work. Proc., vol. 1457, pp. 32–47. CEUR-WS.org (2015)
10. Kalo, J.C., Homoceanu, S., Rose, J., Balke, W.T.: Avoiding Chinese Whispers: Controlling End-to-End Join Quality in Linked Open Data Stores. In: Proc. of the ACM Web Science Conf. pp. 5:1—-5:10. WebSci '15, ACM (2015)
11. Kilicoglu, H., Shin, D., Fiszman, M., Rosemblat, G., Rindflesch, T.C.: SemMedDB: A PubMed-scale repository of biomedical semantic predications. Bioinformatics **28**(23), 3158–3160 (2012)
12. Lebo, T., Sahoo, S., McGuinness, D.: PROV-O: The PROV Ontology. https://www.w3.org/TR/prov-o/ (2013)
13. Patel-Schneider, P.: Contextualization via qualifiers. In: Workshop on Contextualized Knowledge Graphs co-located with 17th Int. Semantic Web Conf., CKG@ISWC 2018 (2018), http://wiki.knoesis.org/index.php/CKG2018
14. Pinto, J.M.G., Balke, W.T.: Can plausibility help to support high quality content in digital libraries? In: Int. Conf. on Theory and Practice of Digital Libraries. pp. 169–180. Springer (2017)
15. Shen, W., Wang, J., Han, J.: Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. IEEE Transactions on Knowledge and Data Engineering **27**(2), 443–460 (2015)
16. Swanson, D.R.: Complementary structures in disjoint science literatures. In: Proc. of the 14th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval. p. 280–289. SIGIR '91, ACM (1991)
17. Tan, W.C.: Provenance in databases: Past, current, and future. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering (2007)
18. Vahdati, S., Palma, G., Nath, R.J., Lange, C., Auer, S., Vidal, M.E.: Unveiling scholarly communities over knowledge graphs. In: Int. Conf. on Theory and Practice of Digital Libraries. pp. 103–115. Springer (2018)
19. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)
20. Wylot, M., Cudré-Mauroux, P., Hauswirth, M., Groth, P.: Storing, tracking, and querying provenance in linked data. IEEE Transactions on Knowledge and Data Engineering **29**(8), 1751–1764 (2017)
21. Xia, F., Wang, W., Bekele, T.M., Liu, H.: Big Scholarly Data: A Survey. IEEE Transactions on Big Data **3**(1), 18–35 (2017)
22. Zhang, R., Cairelli, M.J., Fiszman, M., Rosemblat, G., Kilicoglu, H., Rindflesch, T.C., Pakhomov, S.V., Melton, G.B.: Using semantic predications to uncover drug–drug interactions in clinical data. J. of biomedical informatics **49**, 134–147 (2014)

B.2. JCDL 2021: A Toolbox for the Nearly-Unsupervised Construction of Digital Library Knowledge Graphs

JCDL'21

Hermann Kroll, Jan Pirklbauer, and Wolf-Tilo Balke. "A Toolbox for the Nearly-Unsupervised Construction of Digital Library Knowledge Graphs". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Urbana-Champaign, IL, USA, 2021, IEEE. DOI: https://doi.org/10.1109/JCDL52503.2021.00014

A Toolbox for the Nearly-Unsupervised Construction of Digital Library Knowledge Graphs

Hermann Kroll*, Jan Pirklbauer*, Wolf-Tilo Balke*
*Institute for Information Systems, TU Braunschweig, Braunschweig, Germany
{kroll, balke}@ifis.cs.tu-bs.de and j.pirklbauer@tu-bs.de

***Abstract*—Knowledge graphs are essential for digital libraries to store entity-centric knowledge. The applications of knowledge graphs range from summarizing entity information over answering complex queries to inferring new knowledge. Yet, building knowledge graphs means either relying on manual curation or designing supervised extraction processes to harvest knowledge from unstructured text. Obviously, both approaches are cost-intensive. Yet, the question is whether we can minimize the efforts to build a knowledge graph. And indeed, we propose a toolbox that provides methods to extract knowledge from arbitrary text. Our toolkit bypasses the need for supervision nearly completely and includes a novel algorithm to close the missing gaps. As a practical demonstration, we analyze our toolbox on established biomedical benchmarks. As far as we know, we are the first who propose, analyze and share a nearly unsupervised and complete toolbox for building knowledge graphs from text.**

***Index Terms*—Knowledge Graph, Information Extraction, Digital Library**

I. Introduction

Knowledge graphs are essential for digital libraries to structure textual collections in an entity-centric way. They open up a variety of applications for all kinds of information needs, such as finding detailed descriptions of cultural heritage objects in the Europeana [1], exploiting drug-disease treatments harvested from PubMed in the SemMedDB [2], semantically querying relationships and properties of Linked Data in Wikidata [3], and many more. But crafting such knowledge graphs for all kinds of domains is time-consuming and expensive. This is because many of today's practical knowledge graphs are built completely manually, such as Wikidata [3] or the Europeana [1], or at best semi-automatically (given that the textual information is sufficiently structured, e.g., harvesting Wikipedia infoboxes in DBpedia [4]).

Yet, why is automatically building knowledge graphs so difficult? On the one hand, the content curated by digital libraries may be too heterogeneous to create good quality knowledge graphs by rule-based approaches. For example, the creators of SemMedDB were quite experienced with medical language. They used a variety of grammatical patterns to extract medical relations from PubMed [2]. The challenges are clear: Neither do the rules adapt to paraphrased pieces of information, nor are they easily transferable to other domains or disciplines. On the other hand, artificial intelligence and machine learning techniques that would cater for this heterogeneity rely on supervision; See [5] for a good overview. For the training of reliable extraction algorithms, tens of thousand training examples are necessary, which in turn are usually again handcrafted. Moreover, this kind of training is needed for each specific entity type, relation, etc.

Although the process of harvesting knowledge from unstructured texts is challenging, novel developments in the area of Open Information Extraction (OpenIE) promise to change the game: OpenIE tools are designed to extract as much information as possible without the need for supervision [5]–[7]. While this would account for the applicability across domains and the excessive need for training data, OpenIE tools still have practical limitations. Since these methods are designed to work on all kinds of information, their extractions within topically focused digital libraries tend to be far too general to result in a concrete graph structure describing the respective domain sufficiently well. Moreover, more complex natural language processing tasks like resolving synonyms or disambiguating homonyms still need domain experts' explicit input and data modeling.

The question is whether these limitations can be bypassed in practical digital library projects? Probably, we need a minimum of supervision. This paper focuses precisely on this gap: We develop a toolbox that converts a collection of unstructured text from arbitrary domains into a structured knowledge graph using as little supervision as possible.

Subsequently, our requirements for our nearly unsupervised toolbox are obvious: It must be capable of processing millions of documents for real-world scenarios, and the resulting knowledge graph should retain good quality. We analyze the necessary steps to build a knowledge graph, including entity linking and information extraction. Entity linking detects concepts of pre-known vocabularies in texts, and information extraction extracts relations between them [5]. Our findings will show that we need practical algorithms to transform general OpenIE outputs into a domain-specific knowledge graph using as little supervision as possible. Here, we develop a novel iterative semi-supervised cleaning algorithm with expert feedback. In addition, we develop a novel extraction technique called PathIE that reuses entity information in the extraction phase. PathIE is more flexible, faster, and has a better recall than established OpenIE tools, but suffers in precision.

In this paper, we will analyze the missing gap for constructing knowledge graphs in digital libraries: *Can we bypass the need for supervision completely? And how reliable and well will tools perform for practical applications in digital libraries?* As far as we know, we are the first who develop

a practical and nearly unsupervised extraction toolbox for digital libraries, see Sect. III. Our toolbox is not domain-specific and bypasses the extensive need for supervision when possible; See our discussion in Sect. V. We have applied and analyzed our toolbox in the biomedical domain; See Sect. IV. However, our evaluation will show that the toolbox will suffer in performance compared to established supervised methods. Although the quality might be lacking, the toolbox offers a nearly unsupervised way to build knowledge graphs in digital libraries. Further, we share our toolbox on GitHub[1] to make it reusable for other researchers. The code is written in Python and is published under the MIT license.

The contributions of our work are:

1) We design an unsupervised, fast, and easy-to-use information extraction method PathIE. PathIE is capable of finding subject-predicate-object facts as well as support the extraction of important keywords.
2) We develop a novel semi-supervised iterative predicate cleaning algorithm utilizing word embeddings and expert feedback.
3) We design a nearly unsupervised toolbox covering entity linking, information extraction, cleaning, and storage. We analyze the quality of our toolbox on established biomedical benchmarks.

II. Related Work

This section gives an overview of related work for the essential components to build knowledge graphs: Entity linking and relation extraction. Besides, we report on work about the canonicalization of open information extraction outputs.

a) Entity Linking: is the task to link text spans to pre-known entities [5]. Many algorithms and frameworks exist to perform entity linking in practice, such as the ConceptMapper. Funk et al. performed a large-scale evaluation of available annotation tools in the biomedical domain [8]. Their findings show that parameters should carefully be chosen for different ontologies to achieve good quality. Dictionary-based algorithms take a vocabulary and a text as an input and perform a direct string-matching, i.e., if an entity term is mentioned in the text, a mapping between the vocabulary entry and the text is produced. An advantage of dictionary-based approaches is their performance, i.e., a single iteration over the text with dictionary-based lookups is enough to produce the annotations. Suffering performance to be more error-tolerant may be done by searching via string similarities, i.e., slight derivations of vocabulary entries are allowed. If entity terms have ambiguous meanings (homonyms), then the context of the entity terms in the text must be considered. Here, more complex approaches are needed to resolve homonymous terms correctly. For example, short abbreviations in the biomedical domain refer to several diseases, genes, and drugs. Tools such as TaggerOne and GNormPlus are designed to consider the context of the words [9], [10]. Typically, these tools are supervised [5], i.e., they are trained with training data to learn the appropriate contexts [9], [10]. There was a long discussion about the complexity of tagging models in [11]. The authors argue that it might help train a language model like BERT to maximize the annotation quality. However, simpler models like classical decision trees perform slightly worse but are trained much faster. In summary, the decision is up to a specific domain and use case. Supervised models offer the best performance, but in practice, dictionary-based approaches might already be sufficient.

b) Relation Extraction: Supervised relation extraction supports the construction of knowledge graphs from text [5], [12]. Collecting training data for supervised methods means compiling tens of thousand example extractions. These examples are then used to train a relation extraction for a single relation. Modern relation extraction even builds upon pre-trained language models like BioBERT [13]. Further, relation extraction tools may build upon distant supervision, i.e., a training procedure does not require explicit sentences and their contained facts [14]. A ground truth of valid facts is sufficient, but no text evidence for them must be provided. A learning procedure then extracts facts from texts to learn which grammatical structures lead to correct extractions. Tools such as Snorkel [15] support the automatic generation of training data by formulating hints on which sentences would be good candidates for a relation. Although the quality might be promising, training relation extraction models means giving examples for every relation, i.e., these models cannot be transferred to another domain. And moreover, having such a ground truth for distant supervision is not always the case. So indeed, although methods exist that try to boil down the need for supervision, here, as far as we know, supervision cannot be bypassed completely. Hence, we design our toolbox to bypass the need for training data in the extraction phase completely.

c) Canonicalizing OpenIE Extractions: Research has already been done on canonicalizing OpenIE extractions [16], [17]. For example, CESI uses word embeddings and side-information to canonicalize open knowledge bases [16]. An open knowledge base may be understood as the output of an open information extraction process. The authors suggest clustering subjects, predicates, and objects in a high-dimensional vector space. They use side-information like additional databases and embeddings to embed a subject, predicate, or object into a high-dimensional vector space. A small part of all subjects and objects must be linked to some existing entity vocabulary. Then, a clustering step is applied to resolve synonymous subjects like *N.Y.C.* and *New York* and predicates like *born in* and *has birthplace*. However, CESI has two major limitations: First, some entity linking is required, and side information is domain-specific, i.e., it is not transferable. Second, using clustering does not yield explainable results. As an example, CESI outputs a list of different predicates belonging to the same cluster. On the one hand, the number of obtained clusters is quite unclear. Finding a good number of clusters is a general problem when clustering. On the other hand, adding a precise predicate label to represent all synonymous predicates is difficult, especially if the predicates'

[1] https://github.com/HermannKroll/KGExtractionToolbox

context is unavailable. Overall, CESI is an exciting approach, but it requires domain-specific side information and has hard-to-interpret outputs.

III. KG Extraction Toolbox

This chapter describes the essential components of our toolbox and our novel methods that close the missing gap between open information extraction and practical knowledge graphs. Returning to our scenario, we aim to build a biomedical knowledge graph that captures knowledge about drugs, diseases, and more. Subsequently, all examples stem from the biomedical domain. However, the toolbox can be transferred to other domains because we bypass the need for supervision.

A. Knowledge Graph

First, we will define knowledge graphs for our purposes. The Semantic Web community and the W3C recommend the Resource Description Framework (RDF) to store knowledge [18]. A triple, called **fact**, consists of a subject, a predicate, and an object. A fact represents a piece of knowledge, e.g., (*simvastatin*, *treats*, *hypercholesterolemia*). Collections of these facts are usually called **knowledge graphs**. Knowledge graphs are entity-centric, i.e., only one node represents the entity *simvastatin*. In a broad sense, an **entity** is an important concept someone is looking for, e.g., drugs and diseases. We denote the set of all entities as $\mathcal{E}$. *Values* such as dates, locations, numeric values, or strings might be of interest as well, e.g., the melting point of some substance. These values are called **literals**, and we denote the set of all literals as $\mathcal{L}$. Formally,

Definition 1: A knowledge graph $KG = (V, E, \Sigma)$ is a collection of knowledge. $V \subseteq (\mathcal{E} \cup \mathcal{L})$ is a set of nodes and $E \subseteq \mathcal{E} \times \Sigma \times (\mathcal{E} \cup \mathcal{L})$ is a set of directed and labeled edges. $f = (s, p, o) \in E$ is a fact with $s \in \mathcal{E}$ being a subject, $p \in \Sigma$ being a predicate and $o \in (\mathcal{E} \cup \mathcal{L})$ being an object.

Yet, a fact is a labeled relation, denoted by a predicate, between a subject and an object. These predicates stem from a set of predicate labels Σ. The RDF standard covers many more things that are beyond the scope of this paper [18]. We focus on relations between entities and literals as the core of each knowledge graph. We discuss the necessary steps to build knowledge graphs from texts in digital libraries in the following. A schematic overview of our pipeline is depicted in Fig. 1.

B. Entity Linking

Entity linking is the task to link text spans to pre-known entities [5]. These entities usually stem from vocabularies or ontologies. Vocabularies collect important entities plus adequate synonymous terms, descriptions, and more. Ontologies may provide additional information about entities like subclass relationships, e.g., *simvastatin is a drug* and *drugs are chemical compounds*. Biomedical researchers already spend much work designing suitable vocabularies and ontologies; see BioPortal[2] for an overview. Designing ontologies is a well-known task for digital libraries, e.g., PubMed uses so-called Medical Subject Headings[3] (MeSH) to accelerate the retrieval quality by resolving synonyms or finding relevant sub-concepts. In a broad sense, entities might be seen as arbitrary resources, e.g., drugs, processes, treatment options, study types, and many more. The Dublin Core Metadata Initiative[4] already proposes a plethora of different vocabularies and gives hints on how to design them in a standardized way. In the following, we will consider the terms vocabulary and ontology synonymous in being collections of entity entries. The process of entity linking is well-known in many digital libraries, e.g., PubMed uses human curators and automatic processes to annotate publications with additional MeSH terms. Returning to our toolbox, we must identify these entities in written texts to extract knowledge about them.

We implement a dictionary-based entity linker to support unsupervised entity linking in our toolbox. The entity linker is designed to handle large amounts of text, i.e., it is designed to have a fast performance. Our entity linker requires an entity vocabulary and text documents as its input. Then, our linker produces entity annotations between the text and the vocabulary as its output. Usually, supporting synonyms and resolving conflicts is straightforward, i.e., entities plus their adequate synonyms are identified by unique identifiers. However, dictionary-based linking typically struggles with minor typing errors, unknown synonyms, homonyms, or custom abbreviations by design. Therefore, our linker supports custom abbreviations in a document. Suppose an author introduces the abbreviation *ASR* for *Aspirin* via *Aspirin (ASR)*. In that case, our linker will resolve the abbreviation in the rest of the corresponding document correctly. Short entity names like well-known abbreviations of some entities may lead to wrongly tagged homonyms. Our linker only links short abbreviations if the corresponding entity is at least detected a single time with its complete mention in the document's scope. In this way, we minimize wrongly linked homonyms. The user can adjust the length of a required complete mention. We support a configuration file to adjust these settings for a user's purpose.

Named Entity Recognition (NER) is a broader method to detect entities and important concepts in texts. NER may recognize entity mentions in the text but does not link these mentions against pre-known entity vocabularies [5], [19]. For example, the Stanford Stanza NER detects person names, organizations, locations, dates, and more in texts [19]. Stanza supports the annotation of 18 different named entity types. Especially, the detection of dates and locations might be beneficial across domains. However, NER comes with the limitation of not providing unique entity ids. A text span is identified as an entity type, but a precise entity id is not provided. NER may lead to synonymous entities in a practical knowledge graph. To demonstrate the usefulness of NER in practice, we integrate an interface for Stanza into our toolbox supporting the annotation of more general named entity types.

[2]https://bioportal.bioontology.org last access: 06.2021

[3]https://meshb.nlm.nih.gov last access: 06.2021

[4]https://www.dublincore.org/specifications/dublin-core/dces/ l.a.: 06.2021

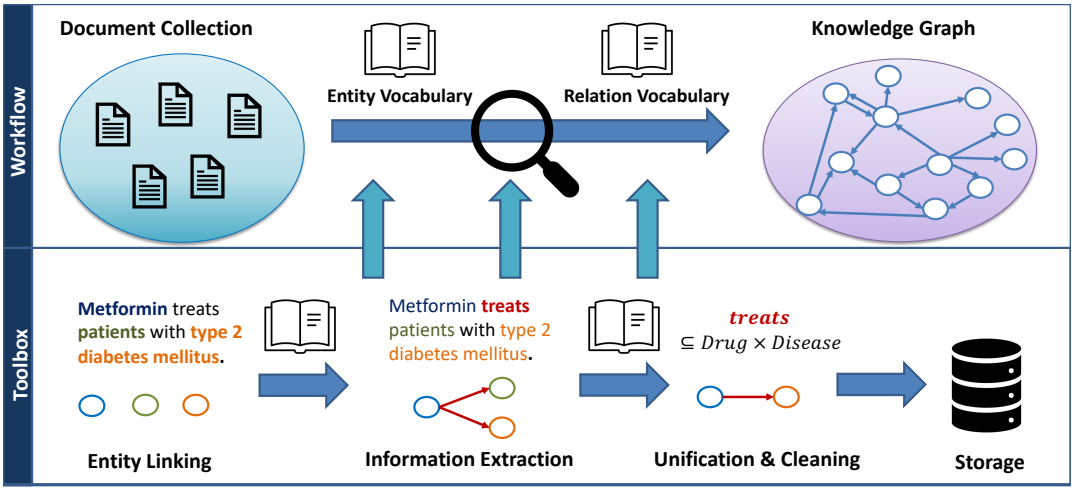


Fig. 1. The Toolbox's Systematic Overview: Entity linking detects important concepts and information extraction extracts important relations between them. Then, the output will be cleaned and loaded into a structured repository.

Details about the quality of Stanza can be found on its project website[5] or in [19]. However, our toolbox might be easily extended by integrating domain-specific and supervised entity linkers. For example, the National Library of Medicine (NLM) provides two powerful and freely usable tools: TaggerOne [10] detects chemicals and diseases, and GNormPlus [9] detects genes and species in texts. We have implemented interfaces to both tools into our toolbox to demonstrate how domain-specific entity linkers may be integrated. Lastly, the detection of arbitrary literals like melting points in texts might be solved via regular expressions. However, the literal detection strongly depends on a given domain's requirements. Thus, we do not integrate literal detection in our toolbox.

C. Open Information Extraction

Information extraction is the task to transform unstructured information into structured information [5], [6]. In this paper, we understand information extraction as the extraction of facts from texts. As a reminder, a fact is a simple triple containing a subject, a predicate and an object, e.g., the fact (*simvastatin*, *treats*, *hypercholestorelemia*) may be extracted from *simvastatin is used to treat hypercholestorelemia in patients*. Information extraction is usually limited to pre-defined relations and entities, and that is why we build upon *open* information extraction methods. Open information extraction is not limited to pre-known relations and hence, can be used across domains. They take arbitrary text as an input and produce facts as an output. Many OpenIE tools are available as free-to-use software and work out-of-the-box. In addition, these natural language processing toolkits work with a plethora of different languages, e.g., the Stanford OpenIE tool supports seven different languages [7], and Stanza supports even 66 different languages [19]. Recently, Kolluru et al. developed a novel OpenIE6 extraction method and analyzed the quality and performance compared to established OpenIE tools for the natural language processing community. Their findings show that OpenIE tools come at best with a F1-measure between 40.0% and 65.6% (tested on several benchmarks). The best performing system is OpenIE6 (2020), which can process up to 31.7 sentences per second on a Nvidia Tesla V100. OpenIE6 builds upon the latest neural extraction methods and is pre-trained on a large variety of text. It does not require domain-specific training and could thus be understood as an unsupervised extraction method. At the same time, the Stanford CoreNLP tool is an older rule-based and model-based approach that is well-supported and has a fast runtime performance [7]. Our toolbox implements interfaces for both OpenIE methods, namely Stanford CoreNLP and OpenIE6.

Although the quality and runtime performance sound sufficient, we can boost their performance further by using entity linking information. Our focus is on constructing knowledge graphs, and hence, we are only interested in facts between entities and literals. First, this restriction boosts the runtime performance, i.e., we only need to process sentences containing at least two different entities/literals mentions. This filtering may significantly reduce the number of sentences to process, depending on the number of annotated entities and literals. The toolbox applies this filtering step before extracting information automatically if desired. OpenIE output usually tends to be more general because OpenIE has no starting point, e.g., subject or object could be anything within a sentence's scope. Therefore, the toolbox uses entity linking information to filter OpenIE fact extractions by subjects and objects. Consider the sample sentence: *Metformin treats patients with diabetes*. OpenIE applied to that sentence result in extractions such as (*metformin*, *treats*, *patients with diabetes*). Subjects and objects should be entities (objects might be literals as well), which is not the default case for OpenIE; see our example above: *Patients with diabetes* is not an entity. The object includes *diabetes* only partially. We assume a partially included entity to be sufficient and rewrite the fact to: (*metformin*, *treats*,

[5]https://stanfordnlp.github.io/stanza/ last access: 06.2021

diabetes). Our toolbox supports a parameter to select the entity filtering mode: None (keep all OpenIE extractions), partial (subject and objects must partially include an entity mention), complete (subject and object must be a fully annotated entity). Then, it cleans the results of the supported OpenIE tools automatically. The toolbox automatically converts passive voice to active voice by the following rule: If the lemmatized predicate includes *be* and the predicate contains a verb in past particle. We utilize the Spacy NLP toolkit to quickly lemmatize the predicate and compute the Part-of-Speech tags for the predicate[6].

D. PathIE

In contrast to conventional information extraction, where arbitrary information is extracted, we only consider interactions between entities and literals. Usually, supervised relation extraction methods do precisely this: They already know the subject and object candidates. OpenIE does not have this information available in the extraction phase. And obviously, we could hardly integrate it into existing tools. However, we already have entity linking information available but want to bypass supervised relation extraction. The central question for a fact extraction is how entities are related within the sentence. We design a high-performance extraction method called **PathIE** utilizing the available entity information. In typical natural language processing, each sentence is represented as a sequence of tokens, i.e., single words. Furthermore, each word is assigned a part-of-speech tag (POS tag), i.e., a word category as nouns, verbs, etc. Tokens are syntactically arranged in a so-called dependency parse tree, i.e., each token has specific relations to other tokens within the sentence (subject, etc.). PathIE utilizes the syntactical structure of a sentence to answer the question of how entities are related. Tools, such as the Stanford CoreNLP suite, offer high performance when tokenizing, POS-tagging, and dependency parsing a sentence [7]. Consider the following example sentence: *Metformin is widely considered to be the optimal initial therapy for patients with type 2 diabetes mellitus*. Our entity linking for this sentence results in *metformin* (drug) and *type 2 diabetes mellitus* (disease).

PathIE utilizes these entity linking information and searches upon the sentence's grammatical structure to derive the relation between both entities. We transform the syntactical sentence structure into a graph, i.e., nodes represent tokens, and edges represent grammatical dependencies between the tokens. We take advantage of the graph representation to perform a path search between the tokens of both entities. Here, we compute the shortest paths because we are interested in the shortest and most substantial syntactical relation. The shortest path for our example sentence is the following sequence of tokens: (*metformin*, *considered*, *therapy*, *patients*, *type 2 diabetes mellitus*). The corresponding relation between both entities can be identified by 1. searching for all verbs on the path (via POS-tags), and 2. by searching for special keywords like *treatment*, *therapy* or *inhibition* on the path. These special keywords can optionally be pre-defined in a vocabulary before applying PathIE. Hence, relations between entities are identified by 1) detecting all verbs on the path via the token's POS tags (VBN, VB, etc.), and 2) optionally detecting hand-crafted vocabulary terms on the path. These terms can be seen as special words like *treatment*, *metabolite* and more. Subject, object, and each identified predicate are composed to a fact extraction for the sentence. The path search is not directed, and thus, we extract both directions for the interaction-keyword *therapy*: (*metformin*, *therapy*, *diabetes*) and (*diabetes*, *therapy*, *metformin*). These facts may be cleaned in the cleaning step discussed subsequently. In some cases, such a path might contain words like *not* or *may* which could lead to a wrong extraction. We support two parameters for PathIE to ignore extractions which contain a *not* or *may*. We assume our path-based extraction technique allows a more flexible extraction yielding a higher recall, but on the other side, decreasing the precision. PathIE relies on a NLP tool to compute dependency parses. We support the computation of dependency parses via Stanford CoreNLP (rule-based and faster) and Stanford Stanza (neural and more precise).

E. Unifying Synonymous Predicates

OpenIE and PathIE yield a variety of different predicates in their extractions by design. As an example, the predicates *treats* and *aids* have the same meaning when talking about the cure of some disease by a drug. We have to unify these synonymous predicates to build a knowledge graph with a manageable set of relations. Hence, our goal is to design a *relation vocabulary*, i.e., a set of relations with a list of synonymous predicates. The relation *treats* might have the synonyms *aids*, *improves* and *prevents*. Using a relation vocabulary allows us to unify the extracted synonymous predicates. Obviously, going through thousands of synonyms manually and building a relation vocabulary is too time-consuming. Hence, the process must be automated, at best, without supervision.

Word embeddings embed words into a high-dimensional vector space by considering their context [20]. Word vectors whose words share a similar context should be located close to each other. Moreover, word embeddings can be trained on arbitrary text without supervision and are already known for their ability to find synonyms for a given word. But, how can we create a vocabulary of relations by unifying the extracted predicates? We cannot entirely bypass the need for supervision here because we need information on how relevant some predicates are in a domain. We design an iterative semi-supervised algorithm allowing domain experts to make these decisions. The algorithm is depicted in Fig. 2 and works as follows:

1) All fact extractions are grouped by their predicate. Then, the group's size is counted.
2) The distances between each predicate and all entries of the relation vocabulary are computed. The nearest neighbor is kept for each predicate. Hence, we obtain

[6]https://spacy.io last access: 06.2021

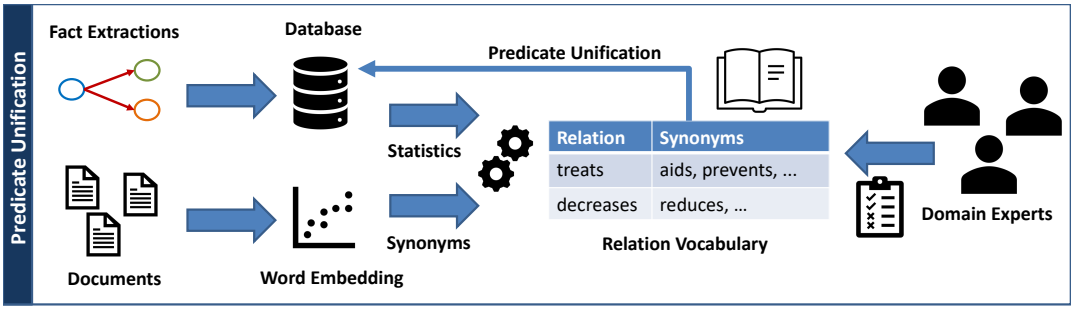


Fig. 2. Systematic overview of our novel semi-supervised iterative predicate unification algorithm. The algorithm reuses extraction information, a word embedding and expert feedback to build a relation vocabulary iteratively. The relation vocabulary is used to clean the open information extractions later.

a mapping between a predicate and a term of the relation vocabulary.

3) All mappings with a distance below a threshold t are removed.
4) A list of mappings between the predicates and relations of the vocabulary is computed. This list is sorted by the number of extractions per predicate. The sorted list is shown to the domain experts.
5) The experts can go through the top entries of the list (maybe top 10). Suppose a predicate is mapped to the wrong relation. In that case, they can improve their relation vocabulary by introducing a new relation or adding the predicate as a synonym to an already included relation.
6) If all the important predicates are mapped correctly, the experts can abort here. If not, the algorithm will repeat at step 2 with the new relation vocabulary. The algorithm outputs the predicate mappings against the relation vocabulary.

Wrapping up, the algorithm takes a word embedding, a threshold t, a relation vocabulary, and a set of fact extractions as input. It results in mappings between predicates and relations of the vocabulary. The algorithm shows the most important predicate (sorted by the number of extractions) to the experts by design. In this way, they can iteratively build the relation vocabulary. They may start with an empty relation vocabulary or may have some first ideas. The parameter t allows experts to choose a gap between precision and recall. If a high threshold is chosen, the algorithm only maps those predicates that are more likely to be synonymous (closer location in the vector space). Hence the precision of correct mappings will be higher. If a lower threshold is chosen, the algorithm may yield wrong mappings but include more correct ones (higher recall). The threshold should be adjusted for domain-specific word embeddings. Finally, the algorithm produces a reliable relation vocabulary plus predicate mappings against it. In this way, many synonymous predicates can be unified with an acceptable amount of labor. Indeed, the algorithm cannot bypass supervision completely but boils the supervision down to building a relation vocabulary with a manageable set of entries iteratively. Our biomedical relation vocabulary needs three iterations and around 63 different vocabulary entries.

F. Knowledge Graph Constraints

Entity linking, information extraction, and predicate cleaning return a set of fact extractions. In practice, knowledge graphs should contain relations with precise semantics, e.g., *treats* is a relation between drugs and diseases. Good examples are **type constraints**, e.g., *treats* should be a relation where all subjects are *drugs* and all objects are *diseases* (*treats* $\subseteq$ *Drugs* $\times$ *Diseases*). In a large-scale extraction scenario, extraction errors are likely to occur, e.g., an erroneously extracted treat relation between two diseases. Obviously, cleaning such a relation by type constraints will increase the overall quality, e.g., *treats* is a relation that has drugs as its subjects and diseases as its objects. Hence, a relation type constraint defines the allowed entity types for subjects and objects. Our toolbox supports type constraints to clean the fact extractions to increase the overall quality. Domain experts can formulate these integrity constraints, and our tool will automatically check them in the cleaning phase. In addition to type constraints, broader integrity constraints might be helpful. For example, if we know that X is the treatment for some disease Y, then Y cannot be an adverse effect of X. Such integrity constraints require domain-specific logic and hence, must manually be formulated by domain experts.

G. Storage and Provenance

The toolbox generates various outputs like entity linking information and information extraction. To minimize the need to handle several files, the toolbox utilizes an object-relational mapper as an interface to an underlying relational database. We use a Postgres system by default, but the relational mapper also supports other systems like SQLite or MySQL. The toolbox stores all information within a single place to reuse specific information if necessary, e.g., the toolbox automatically queries entity linking information in the extraction phase if required. However, the toolbox supports the export of outputs in different formats; See the GitHub page for more details. Entity linking information can be exported as PubTator documents or in a JSON format. Consider the following scenario: *Metformin is used to treat patients with diabetes* is a sentence in some document. The entity linking steps may yield that *Metformin* represents the ChEMBL[7] identifier *CHEMBL1431*

[7]https://www.ebi.ac.uk/chembl/ last access: 06.2021

and *patients with diabetes* are associated with the disease *Diabetes Mellitus* also known as the MeSH identifier *D003920*. The information extraction steps yields (*CHEMBL1431*, *is used to treat*, *D003920*). The cleaning step will map the predicate *is used to treat* to *treats*, so that, (*CHEMBL1431*, *treats*, *D003920*) is obtained. Next, the toolbox must export the extractions in some format. The easiest way would be to export facts like (*CHEMBL1431*, *treats*, *D003920*). However, if the knowledge graph is used in downstream applications, it might be helpful to provide additional provenance information. Provenance ranges from just storing a reference to the document, in which a fact was extracted, to storing all information starting by the entity linking step. That means we must store a tuple with the signature (*document*, *subject_str*, *subject_id*, *subject_type*, *predicate_str*, *predicate*, *object_str*, *object_id*, *object_type*, *sentence*). For our example, a tuple would look like (*doc_123*, *Metformin*, *CHEMBL1431*, *Drug*, *is used to treat*, *treats*, *patients with diabetes*, *D003920*, *Disease*, *Metformin is used to treat patients with diabetes*). The toolbox supports the export of facts plus provenance information to support both scenarios. The fact extractions and useful provenance information can be exported as a TSV file or RDF-serialization format. More details can be found on our GitHub page.

IV. Evaluation

In the following, we evaluate our toolbox by applying it to established biomedical benchmarks. All experiments are performed on our server, which has two Intel Xeon E5-2687W (3.1 GHz, eight cores, 16 threads), 377 GB of DDR3 main memory, NVME SDDs as its primary storage, and a Nvidia 1080 GTX TI as a GPU. We enable the GPU support for the Stanford Stanza toolkit and OpenIE6.

A. Entity Linking

We evaluate and report the quality of entity linking with our toolbox subsequently. Therefore, we have selected four established biomedical benchmarks: 1. disease normalization in NCBI Disease [21], 2. disease normalization in Biocreative V CD-R [22], 3. chemical normalization in Biocreative V CD-R [22] and 4. human gene normalization in Biocreative II Gene Normalization [23]. All of these benchmarks require entity detection in text. Then, the entity mentions must be linked to normalized (disambiguated) concepts (entity identifiers). All benchmarks provide entity vocabularies that we use as inputs for our linker. In comparison to our entity linker, we report the results of the latest biomedical entity linkers TaggerOne [10] and GNormPlus [9]. TaggerOne and GNormPlus are both supervised. We report all results in Table I.

TaggerOne recognizes diseases on the NCBI Disease Benchmark with a precision of 82.2% and a recall of 79.2%. Our entity linker achieves a precision of 74.5% and recall of 55.1%. On the BioCreative V benchmark, TaggerOne detects diseases with a precision of 84.6% and a recall of 82.7%. In comparison, our entity linker achieves a precision of 82.8% and a recall of 62.0%. Chemicals are found with a precision of 88.8% and a recall of 90.3% by evaluating TaggerOne on the BioCreative V benchmark. Our entity linker achieves a precision of 76.6% and a recall of 78.7% when linking chemicals. GNormPlus achieves a precision of 87.1% and a recall of 86.4% on BioCreative II. In comparison, our linker achieves a precision of 60.1% and a recall of 52.4%.

Next, we evaluate the linking quality when designing our own entity vocabularies. In joint work with two pharmaceutical domain experts, we design entity vocabularies for drugs, plant families, and dosage forms. We apply our entity linker against a random sample of PubMed abstracts and randomly pick 50 produced entity annotations for each entity type for evaluation purposes. We gave these entity annotations and the corresponding sentence to both domain experts. They carefully read the sentence (context) and decide together if the annotated entity is mentioned. Hence, we could only estimate the precision for these entity types. Drugs are tagged with 90% precision, plant families with 82% precision, and dosage forms with 82% precision. Concerning NER, Stanza has already been evaluated on two different benchmarks, namely CoNLL03 and OntoNotes. Stanza achieves a F1-score of 92.1% on CoNLL03 and 88.8% on OntoNotes [19].

Next, we report the linkers' runtime to estimate if they are applicable in a large-scale scenario. First, we randomly sample 10k PubMed titles and abstracts containing at least a single drug (to ensure that they contain relevant entities). Then, we run each entity linker three times on this sample to measure its runtime. TaggerOne takes around $(149 \pm 1)\,\mathrm{min}$ and GNormPlus takes around $(118 \pm 1)\,\mathrm{min}$ to complete. Our dictionary-based linker takes around $(77 \pm 1)\,\mathrm{s}$ to complete. Stanza takes around $(41 \pm 1)\,\mathrm{min}$ utilizing our GPU. Deactivating GPU supports leads to a runtime of about 9 hours.

a) Discussion: The evaluation of linking entities reveals how well an unsupervised method might work. Our entity linker lacks around 7.7% points (NCBI disease) and 1.8% points (BioCreative V) precision behind TaggerOne when detecting diseases. Although the precision of TaggerOne is not far ahead, the recall of our linker clearly lacks behind: 79.2% and 82.7% (TaggerOne) vs. 55.1% and 62.0% (our linker). However, TaggerOne takes around 150 minutes, whereas our linker needs around 77 seconds. A similar observation could be made for linking chemicals on BioCreative V. Indeed, linking human genes is challenging because gene descriptions are often short and ambiguous to other terms. Here, our entity linker clearly falls behind GNormPlus. Especially if terms are unambiguous, our entity linker achieves a high precision, e.g., 90% when linking drugs. Hence, our entity linker is a worthy competitor: Our linker is fast, achieves good precision but lacks behind in recall. Nevertheless, our entity linker does not require supervision which is a significant advantage. In summary, the development of an entity linker for a specific domain depends on the complexity and disambiguation of entity terms. Indeed, dictionary-based methods already achieve a good performance and bypass the need for supervision here. However, supervised methods should be preferred in scenarios where context is essential, e.g., when linking genes.

TABLE I
ENTITY LINKING QUALITY ON BIOMEDICAL BENCHMARKS: STATE-OF-THE-ART (SOTA) TAGGERS ARE COMPARED TO OUR UNSUPERVISED ENTITY LINKER. THE SOTA-TAGGING QUALITY RESULTS ARE FROM TAGGERONE [10] AND GNORMPLUS [9].

Benchmark	**Entity Type**	**Quality of SOTA Entity Linker**				**Quality of our Entity Linker**		
		Name	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
NCBI Disease [21]	Disease	TaggerOne	82.2 %	79.2 %	80.7 %	74.5 %	55.1 %	63.3 %
BioCreative V CD-R [22]	Disease	TaggerOne	84.6 %	82.7 %	83.7 %	82.8 %	62.0 %	70.9 %
BioCreative V CD-R [22]	Chemical	TaggerOne	88.8 %	90.3 %	89.5 %	76.6 %	78.7 %	77.6 %
BioCreative II GN [23]	Human Gene	GNormPlus	87.1 %	86.4 %	86.7 %	60.1 %	52.4 %	56.0 %

TABLE II
QUALITY OF OUR SEMI-SUPERVISED ITERATIVE PREDICATE CLEANING ALGORITHM. WE APPLY THREE ITERATIONS ON A PUBMED SAMPLE.

Relation	**W. Prec.**	**Top Predicate Mappings**
decreases	80.4%	reduce(1.6M), decrease(1M), mediate(430K), attenuate(339K), lower(275K), ...
induces	88.8%	induce(3.5M), increase(1.9M), cause(1.3M), result(800K), lead(698K), ...
treats	77.8%	treatment(1.1M), treats(713K), use(654K), therapy(456K), improve(192K), ...
metabol.	99.8%	metabolism(31K), catalyze(19K), metabolite(10K), metabolize(8.7K), oxidize(3K), ...
inhibits	98.6%	inhibitor(182K), inhibit(149K), inhibition(89K), suppress(44K), downregulate(9.8K), ...
interacts	69.6%	bind(497K), regulate(345K), act(148K), modulate(131K), interact(118K), ...

TABLE III
CDR2015 BENCHMARK EVALUATION [22]. THE TABLE REPORTS THE EXTRACTION QUALITY FOR OPENIE TOOLS, PATHIE AND BASELINES.

Method	**Quality**		
	Prec.	Rec.	F1
CoreNLP OpenIE	64.9 %	5.8 %	10.6 %
OpenIE6	53.1 %	5.5 %	10.0 %
PathIE	50.8 %	31.7 %	39.1 %
PathIE Stanza	51.1 %	30.9 %	38.5 %
Workshop Best Precision [22]	90.5 %	80.8 %	85.4 %
Workshop Best Recall [22]	86.1 %	86.2 %	86.1 %

B. Predicate Unification

We perform an expert evaluation to estimate our novel semi-supervised predicate cleaning algorithm's quality in the following. Therefore, we apply PathIE on a PubMed sample of about 5.6 million PubMed documents. The sample contains documents in which at least a single drug was linked (because we are interested in pharmaceutical relations). We use the biomedical word embedding trained on PubMed from [24]. Together with two pharmaceutical domain experts, we have designed a relation vocabulary with ten relations and around 53 entries. We build the vocabulary incrementally by performing three iterations. We tested a few thresholds for this paper and found a threshold of 0.4 to deliver good results.

Next, we evaluate six relations by selecting the top-30 predicate mappings for each relation (ranked by the vector distance). We give these mappings to two domain experts for evaluation, i.e., they decide whether a mapping is correct. The results are listed in Table II. We compute the weighted precision to weight the mapped predicates based on their frequency, i.e., predicates that occur more frequently have a greater influence on the weighted precision. We report the top five predicates that are mapped to the corresponding relation with their extraction frequency. For example, the top 30 predicates mapped to the relation *decreases* have a weighted precision of 80.4%. The weighted precision of the results is between 69.6% (interacts) up to 99.8% (metabolizes). The quality depends on how precise a relation can be formulated with corresponding synonyms, e.g., metabolizes has precise and unambiguous terms. Hence, most of the mapped predicates are correct. In contrast, the predicate *uses* is wrongly mapped to treats. Further improvements to the vocabulary can quickly be made by applying the predicate unification algorithm again., e.g., *uses* could be mapped to another relation.

C. Information Extraction

In the following, we evaluate the information extraction quality and measure the runtime to estimate whether OpenIE tools are applicable in large-scale scenarios. We evaluate both OpenIE tools in our toolbox, namely Stanford OpenIE and OpenIE6. In comparison, we analyze our PathIE extraction method based on Stanford CoreNLP and PathIE Stanza based on Stanford Stanza. For the evaluation, we apply our toolbox to already established benchmarks: 1. BioCreative V CD-R (relations between chemicals and diseases), and 2. BioCreative VI ChemProt (relations between chemicals and proteins).

a) BioCreative V CD-R: The benchmark [22] requires the extraction of *induces* relations between chemicals and diseases. The benchmark provides PubMed abstracts that are annotated with chemicals and diseases. Here, we apply our unsupervised extraction methods plus cleaning to extract *induce* relations from texts. We reuse the previously defined relation vocabulary. It comprises around ten synonyms for the *induces* relation. We did not adjust the vocabulary for this benchmark. Hence, we do not require training data here at all. The results are reported in Table III. For comparison, we include the workshop's best-performing systems concerning precision and recall.

CoreNLP OpenIE yields a precision of 59.3% and a low recall of 5.1%. OpenIE6 yields a precision of 53.1% and a recall of 5.5%. PathIE yields a precision of 50.8% and a recall of 31.7%. PathIE Stanza produces comparable results, i.e., 51.1% precision and 30.9% recall. The workshop's best performing and supervised systems achieve a precision of

TABLE IV
BIOCREATIVE VI CHEMPROT EVALUATION [25]. THE TABLE REPORTS THE EXTRACTION QUALITY FOR OPENIE, PATHIE AND BASELINES.

Method	Quality		
	Prec.	Rec.	F1
CoreNLP OpenIE	59.3 %	5.1 %	9.3 %
OpenIE6	55.9 %	6.2 %	11.1 %
PathIE	30.3 %	55.3 %	39.1 %
PathIE Stanza	29.4 %	56.6 %	38.7 %
Sentence Co-Mention [25]	4.4 %	98.0 %	0.08 %
Workshop Best Precision [25]	74.4 %	55.3 %	63.4 %
Workshop Best Recall [25]	56.1 %	67.8 %	61.4 %
BioBERT [13]	77.0 %	75.9 %	76.5 %

90.5% and 86.1%, with a corresponding recall of 80.8% and 86.2%.

b) BioCreative VI ChemProt: The benchmark [25] requires the extraction of relations between chemicals and proteins from the text. Therefore, PubMed abstracts with chemical and protein annotations are given. The task is to extract five relations, namely, *inhibits*, *upregulates*, *agonist*, *antagonist* and *substrate*. Together in cooperation with both domain experts, we carefully read the relation descriptions and build a relation vocabulary for this benchmark. The relation vocabulary comprises five relations and a few synonyms for each relation. To assist the process of finding suitable synonyms, we briefly had a look at the benchmarks training data. The creation of the vocabulary takes around one hour. Then, we evaluate our extraction methods on the benchmark's test data. The results are listed in Table IV. For comparison, we include the workshop's best performing concerning precision and recall. In addition, we include a simple sentence co-mention baseline [25] and the BioBERT relation extraction findings [13].

CoreNLP OpenIE yield 59.3 % precision and 5.1 % recall. OpenIE6 comes with a precision of 55.9 % and a recall of 6.2 %. PathIE achieves 30.3 % precision and 55.3 % recall. PathIE Stanza has a slightly lower precision 29.4 %, but higher recall of 56.6 %. Just for a comparison, the sentence co-mention baseline yields only a precision of 4.4 % and a recall of 98.0 %. Hence, a few relations must be mentioned across sentences. The best precision-oriented baselines achieves 74.4 % precision and 55.3 % recall. The best recall-oriented baseline system achieves 56.1 % precision and 67.8 % recall. BioBERT, a language model trained on the whole PubMed collection, was fine-tuned for the relation extraction task [13]. Then, BioBERT yields 77.0 % precision and 75.9 % recall. Both workshop baselines and the fine-tuning of BioBERT rely on supervision.

c) Performance Analysis: Next, we analyze the runtime of our extraction methods on a random sample of two million PubMed abstracts that include at least a single drug (biomedical focus). In summary, this sample has 178.5k entity annotations. We extract 52.6k sentences that include two different entity mentions. PathIE takes about two minutes, and PathIE Stanza takes 42 minutes on our GPU. CoreNLP takes 8.5 minutes, and OpenIE6 takes about one hour on our GPU.

d) Discussion: The runtime evaluation demonstrates that all four extraction methods are applicable in a large-scale scenario. However, the comparison to supervised methods shows disadvantages concerning precision and recall. Although supervised methods outperform our unsupervised methods, especially PathIE is a strong competitor. PathIE does not require training data at all and still may come with a precision of 50%. PathIE is designed to extract all relations between entities in sentences if connected via a predicate or a special keyword in the grammatical structure. Having a closer look at the BioCreative VI ChemProt benchmark, PathIE yields about 40% precision for the *inhibits* relation, but only 18% precision for *upregulates*. Thus, PathIE can extract some relations with good quality, but not in all cases. As already expected, OpenIE tools lose recall in comparison with PathIE. Here, OpenIE fails to extract facts from long, complex, or nested sentences. For example, OpenIE can find an inhibition in a precise clause like *Metformin inhibits mtor*. However, OpenIE could not extract the relation *inhibits* in a phrase like *Metformin is a known inhibitor for mtor*. The problem here is that the verb *is* does not give enough information to extract a meaningful *inhibits* relation. Further advancement in OpenIE would be necessary to extract such relations with a higher recall. As a last remark, biomedical relation extraction benchmarks tend to include complex, long, and nested sentences. The extraction quality of our toolbox might hence be better in another domain if sentences are more straightforward.

V. CONCLUSION

In this paper, we have developed a nearly-unsupervised toolbox to construct knowledge graphs from texts in digital libraries. An overview of our toolbox and its components is given in Table V. We have implemented a dictionary-based entity linker supporting custom abbreviations and short abbreviation resolution. In practice, our toolbox may be extended by domain-specific entity linkers like we already demonstrated with TaggerOne and GNormPlus. Our toolbox provides interfaces for the latest OpenIE tools, namely Stanford CoreNLP and OpenIE6. In addition, we design a recall-oriented and flexible extraction method PathIE. Reliable fact extractions are produced by combining these unsupervised extractions methods with entity-based filtering and a novel iterative semi-supervised predicate unification algorithm. Type constraints ensure precise semantics for relations, and integrity constraints might minimize errors in the extraction phase.

The evaluation demonstrates that we already achieve a good quality on established benchmarks. Supervised methods outperform our linker by a small margin for entity linking, but they rely on training data and are way slower. Next, supervised relation extraction outperforms our unsupervised extraction methods clearly. Moreover, the best quality can only be achieved by utilizing language models like BioBERT for relation extraction. However, training a language model for a given domain can be a very cost-intensive task [13]. The training of BioBERT took even 23 days on eight Nvidia V100 GPUs [13]. Collecting enough training data for a re-

TABLE V
AN OVERVIEW OF OUR TOOLBOX'S COMPONENTS. WE REPORT WHETHER THE COMPONENT RELIES ON SUPERVISION AND IS DOMAIN-SPECIFIC.

Component	Supervision	Domain-Specific	Supported Tools
Entity Linking	no	no	A dictionary-based entity linker for arbitrary vocabularies. Named Entity Recognition (NER) via Stanford Stanza (Location, Time, and more) [19]. We integrate TaggerOne (Diseases, Chemicals) [10] and GNormPlus (Genes, Species) [9] as examples.
Information Extraction	no	no	PathIE, PathIE Stanza, CoreNLP OpenIE [7] and OpenIE6 [6]
Predicate Cleaning	yes	yes	Entity-based filtering and an iterative semi-supervised predicate unification
Constraint Cleaning	no	yes	Cleaning via type constraints
Storage	no	no	Object-relational-mapper for relational databases and JSON/RDF-serialization export

liable entity linking or relation extraction comes even with a price. In practice, this could hinder the construction of a knowledge graph. Precisely here, we propose our toolbox. The toolbox requires entity vocabularies and expert interaction when cleaning the extracted predicates. Many domains already have designed entity vocabularies that are ready to use. And if not, tools like Stanza or utilizing entity information of existing knowledge graphs like Wikidata may close the gap here. In practice, predicate cleaning boils down to selecting a few hand-crafted relations plus synonyms in an iterative fashion. Experts control which predicates are mapped to the corresponding relation, and similar predicates are found via unsupervised word embeddings. We believe that our toolbox offers the possibility of harvesting knowledge from text across domains. Although the quality might not be the best, there is often no alternative in practice. Collecting training data and training language models is often too cost-intensive to concern.

ACKNOWLEDGMENT

Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): PubPharm – the Specialized Information Service for Pharmacy (Gepris 267140244).

REFERENCES

[1] A. Isaac and B. Haslhofer, "Europeana linked open data–data. europeana. eu," *Semantic Web*, vol. 4, no. 3, pp. 291–297, 2013.

[2] H. Kilicoglu, D. Shin, M. Fiszman, G. Rosemblat, and T. C. Rindflesch, "Semmeddb: a pubmed-scale repository of biomedical semantic predications," *Bioinformatics*, vol. 28, no. 23, pp. 3158–3160, 2012.

[3] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrandečić, "Introducing wikidata to the linked data web," in *The Semantic Web – ISWC 2014*. Cham: Springer Int. Publishing, 2014, pp. 50–65.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The Semantic Web*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 722–735.

[5] G. Weikum, L. Dong, S. Razniewski, and F. M. Suchanek, "Machine knowledge: Creation and curation of comprehensive knowledge bases," *CoRR*, vol. abs/2009.11564, 2020.

[6] K. Kolluru, V. Adlakha, S. Aggarwal, Mausam, and S. Chakrabarti, "OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 3748–3761.

[7] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, Maryland: Association for Computational Linguistics, 2014, pp. 55–60.

[8] C. Funk, W. Baumgartner, B. Garcia, C. Roeder, M. Bada, K. B. Cohen, L. E. Hunter, and K. Verspoor, "Large-scale biomedical concept recognition: an evaluation of current automatic annotators and their parameters," *BMC Bioinformatics*, vol. 15, no. 1, p. 59, 2014.

[9] C.-H. Wei, H.-Y. Kao, and Z. lu, "Gnormplus: An integrative approach for tagging genes, gene families, and protein domains," *BioMed research international*, vol. 2015, p. 918710, 2015.

[10] R. Leaman and Z. Lu, "TaggerOne: joint named entity recognition and normalization with semi-Markov Models," *Bioinformatics*, vol. 32, no. 18, pp. 2839–2846, 2016.

[11] J. Li, Y. Li, X. Wang, and W.-C. Tan, "Deep or simple models for semantic tagging? it depends on your data," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 2549–2562, 2020.

[12] B. D. Trisedya, G. Weikum, J. Qi, and R. Zhang, "Neural relation extraction for knowledge base enrichment," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 229–240.

[13] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[14] A. Smirnova and P. Cudré-Mauroux, "Relation extraction using distant supervision: A survey," *ACM Comput. Surv.*, vol. 51, no. 5, 2018.

[15] A. R. S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: Rapid training data creation with weak supervision," *Proceedings of the VLDB Endowment*, vol. 11, no. 3, 2017.

[16] S. Vashishth, P. Jain, and P. Talukdar, "Cesi: Canonicalizing open knowledge bases using embeddings and side information," in *Proceedings of the 2018 World Wide Web Conference*, ser. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, p. 1317–1327.

[17] S. Dash, G. Rossiello, N. Mihindukulasooriya, S. Bagchi, and A. Gliozzo, "Joint entity and relation canonicalization in open knowledge graphs using variational autoencoders," *CoRR*, vol. abs/2012.04780, 2020.

[18] F. Manola and E. Miller, "RDF primer," WWW Consortium, Recommendation REC-rdf-primer-20040210, 2004.

[19] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, "Stanza: A python natural language processing toolkit for many human languages," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics, 2020, pp. 101–108.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[21] R. I. Doğan, R. Leaman, and Z. Lu, "Ncbi disease corpus: A resource for disease name recognition and concept normalization," *Journal of Biomedical Informatics*, vol. 47, pp. 1–10, 2014.

[22] C.-H. Wei, Y. Peng, R. Leaman, A. P. Davis, C. J. Mattingly, J. Li, T. C. Wiegers, and Z. Lu, "Overview of the biocreative v chemical disease relation (cdr) task," in *Proceedings of the fifth BioCreative challenge evaluation workshop*, vol. 14, 2015.

[23] A. A. Morgan, Z. Lu, X. Wang, A. M. Cohen, J. Fluck, P. Ruch, and et al., "Overview of biocreative ii gene normalization," *Genome Biology*, vol. 9, no. 2, p. S3, 2008.

[24] Y. Zhang, Q. Chen, Z. Yang, H. Lin, and Z. Lu, "Biowordvec, improving biomedical word embeddings with subword information and mesh," *Scientific data*, vol. 6, no. 1, pp. 1–9, 2019.

[25] M. Krallinger, O. Rabal, S. A. Akhondi, M. P. Pérez, J. Santamaría, G. P. Rodríguez *et al.*, "Overview of the biocreative vi chemical-protein interaction track," in *Proceedings of the sixth BioCreative challenge evaluation workshop*, vol. 1, 2017, pp. 141–146.

B.3. ICADL 2021: Narrative Query Graphs for Entity-Interaction-Aware Document Retrieval

ICADL'21

Hermann Kroll, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. "Narrative Query Graphs for Entity-Interaction-Aware Document Retrieval". International Conference on Asian Digital Libraries (ICADL), Online, 2021, Springer. DOI: https://doi.org/10.1007/978-3-030-91669-5_7

Narrative Query Graphs for Entity-Interaction-Aware Document Retrieval

Hermann Kroll[0000−0001−9887−9276], Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke[0000−0002−5443−1215]

Institute for Information Systems, TU Braunschweig, Braunschweig, Germany
{kroll,kalo,balke}@ifis.cs.tu-bs.de and
{j.pirklbauer, morris.kunz, j.ruthmann}@tu-bs.de

Abstract. Finding relevant publications in the scientific domain can be quite tedious: Accessing large-scale document collections often means to formulate an initial keyword-based query followed by many refinements to retrieve a *sufficiently complete, yet manageable* set of documents to satisfy one's information need. Since keyword-based search limits researchers to formulating their information needs as a set of unconnected keywords, retrieval systems try to guess each user's intent. In contrast, distilling short narratives of the searchers' information needs into simple, yet precise entity-interaction graph patterns provides all information needed for a precise search. As an additional benefit, such graph patterns may also feature variable nodes to flexibly allow for different substitutions of entities taking a specified role. An evaluation over the PubMed document collection quantifies the gains in precision for our novel entity-interaction-aware search. Moreover, we perform expert interviews and a questionnaire to verify the usefulness of our system in practice.

Keywords: Narrative Queries, Graph-based Retrieval, Digital Libraries

1 Introduction

PubMed, the world's most extensive digital library for biomedical research, consists of about 32 million publications and is currently growing by more than one million publications each year. Accessing such an extensive collection by simple means such as keyword-based retrieval over publication texts is a challenge for researchers, since they simply cannot read through hundreds of possibly relevant documents, yet cannot afford to miss relevant information in retrieval tasks. Indeed, there is a dire need for retrieval tools tailored to specific information needs in order to solve the above conflict. For such tools, deeper knowledge about the particular task at hand and the specific semantics involved is essential. Taking a closer look at the nature of scientific information search, interactions between entities can be seen to represent a short narrative [8], a short story of interest: how or why entities interact, in what sequence or roles they occur, and what the result or purpose of their interaction is [3, 8].

Indeed, an extensive query log analysis on PubMed in [4] clearly shows that researchers in the biomedical domain are often interested in interactions between entities such as drugs, genes, and diseases. Among other results, the authors report that a) on average significantly more keywords are used in PubMed queries than in typical Web searches, b) result set sizes reach an average of (rather unmanageable) 14,050 documents, and c) keyword queries are on average 4.3 times refined and often include more specific information about the keywords' intended semantic relationships, e.g., *myocardial infarction AND aspirin* may be refined to *myocardial infarction prevention AND aspirin*. Given all these observations, native support for entity-interaction-aware retrieval tasks can be expected to be extremely useful for PubMed information searches and is quite promising to generalize to other kinds of scientific domains, too. However, searching scientific document collections curated by digital libraries for such narratives is tedious when being restricted to keyword-based search, since the same narrative can be paraphrased in countless ways [1, 4].

Therefore, we introduce the novel concept of *narrative query graphs for scientific document retrieval* enabling users to formulate their information need as entity-interaction queries explicitly. Complex interactions between entities can be precisely specified: Simple interactions between two entities are expressed by a basic query graph consisting of two nodes and a labeled edge between them. Of course, by adding more edges and entity nodes, these basic graph patterns can be combined to form arbitrarily complex graph patterns to address highly specialized information needs. Moreover, narrative query graphs support *variable nodes* supporting a far broader expressiveness than keyword-based queries. As an example, a researcher might search for treatments of *some disease* using *simvastatin*. While keyword-based searches would broaden the scope of the query far in excess of the user intent by just omitting any specific disease's name, narrative query graphs can focus the search by using a variable node to find documents that describe treatments of *simvastatin* facilitated by an entity of the type *disease*. The obtained result lists can then be clustered by possible node substitutions to get an entity-centric literature overview. Besides, we provide provenance information to explain why a document matches the query.

In summary, our contributions are:

1. We propose narrative query graphs for scientific document retrieval enabling fine-grained modeling of users' information needs. Moreover, we boost query expressiveness by introducing variable nodes for document retrieval.
2. We developed a prototype that processes arbitrary narrative query graphs over large document collections. As a showcase, the prototype performs searches on six million PubMed titles and abstracts in real-time.
3. We evaluated our system in two ways: On the one hand, we demonstrated our retrieval system's usefulness and superiority over keyword-based search on the PubMed digital library in a manual evaluation including practitioners from the pharmaceutical domain. On the other hand, we performed interviews and a questionnaire with eight biomedical experts who face the search for literature on a daily basis.

2 Related Work

Narrative query graphs are designed to offer complex querying capabilities over scientific document collections aiming at high precision results. Focusing on retrieving entity interactions, they are a subset of our conceptual overlay model for representing narrative information [8]. We discussed the first ideas to bind narratives against document collections in [9]. This paper describes the complete retrieval method and evaluation of narrative query graphs for document retrieval. In the last decade three major research areas were proposed to improve text-based information retrieval.

Machine Learning for Information Retrieval. Modern personalized systems try to guess each user's intent and automatically provide more relevant results by query expansion, see [1] for a good overview. Mohan et al. focus on information retrieval of biomedical texts in PubMed [13]. The authors derive a training and test set by analyzing PubMed query logs and train a deep neural network to improve literature search. Entity-based language models are used to distinguish between a term-based and entity-based search to increase the retrieval quality [16]. Yet, while a variety of approaches to improve result rankings by learning how a query is related to some document [13, 19, 20], have been proposed, gathering enough training data to effectively train a system for all different kinds of scientific domains seems impossible. Specialized information needs, which are not searched often, are hardly covered in such models.

Graph-based Information Retrieval. Using graph-based methods for textual information retrieval gained in popularity recently [3, 17, 18, 20]. For instance, Dietz et al. discuss the opportunities of entity linking and relation extraction to enhance query processing for keyword-based systems [3] and Zhao et al. demonstrate the usefulness of graph-based document representations for precise biomedical literature retrieval [20]. Kadry et al. also include entity and relationship information from the text as a learning-to-rank task to improve support passage retrieval [5]. Besides, Spitz et al. build a graph representation for Wikipedia to answer queries about events and entities more precisely [17]. But in contrast to our work, the above approaches focus on unlabeled graphs or include relationships only partially.

Knowledge Bases for Literature Search. GrapAl, for example, a graph database of academic literature, is designed to assist academic literature search by supporting a structured querying language, namely Cypher [2]. GrapAl mainly consists of traditional metadata like authors, citations, and publication information but also includes entities and relationship mentions. However, complex entity interactions are not supported, as only a few basic relationships per paper are annotated. As a more practical system that extracts facts from text to support question answering, QKBfly has been presented [14] . It constructs a knowledge base for ad-hoc question answering during query time that provides journalists with the latest information about emergent topics. However, they focus on retrieving relevant facts concerning a single entity. In contrast, our focus is on document retrieval for complex entity interactions.

3 Narrative Query Graphs

Entities represent things of interest in a specific domain: Drugs and diseases are prime examples in the biomedical domain. An entity $e = (id, type)$, where id is a unique identifier and $type$ the entity type. To give an example, we may represent the drug *simvastatin* by its identifier and entity type as follows: $e_{simvastatin} = (D019821, Drug)$. Typically, entities are defined by predefined ontologies, taxonomies, or controlled vocabularies, such as NLM's MeSH or EMBL's ChEBI. We denote the set of known entities as $\mathcal{E}$. Entities might also be classes as well, e.g., the entity *diabetes mellitus* (Disease) refers to a class of specialized diabetes diseases such as DM type 1 and DM type 2. Thus, these classes can be arranged in subclass relations, i.e., DM type 1 is the subclass of general diabetes mellitus. Since we aim to find entity interactions in texts, we need to know where entities are mentioned. In typical natural language processing, each sentence is represented as a sequence of tokens, i.e., single words. Therefore, an **entity alignment** maps a token or a sequence of tokens to an entity from $\mathcal{E}$ if the tokens refer to it.

We call an interaction between two entities a **statement** following the idea of knowledge representation in the Resource Description Framework (RDF) [12]. Hence, a **statement** is a triple (s, p, o) where $s, o \in \mathcal{E}$ and $p \in \Sigma$. Σ represents the set of all interactions we are interested in. We focus only on interactions between entities, unlike RDF, where objects might be literals too. For example, a *treatment* interaction between *simvastatin* and *hypercholesterolemia* is encoded as ($e_{simvastatin}$, *treats*, $e_{hypercholesterolemia}$). We call a set of extractions from a single document a so-called **document graph**.

Document graphs support narrative querying, i.e., the query is answered by matching the query against the document's graph. Suppose a user formulates a query like ($e_{simvastatin}$, *treats*, $e_{hypercholesterolemia}$). In that case, our system retrieves a set of documents containing the searched statement. Narrative query graphs may include typed variable nodes as well. A user might query ($e_{simvastatin}$, *treats*, *?X(Disease)*), asking for documents containing *some* disease treatment with *simvastatin*. Hence, all documents that include *simvastatin* treatments for diseases are proper matches. Formally, we denote the set of all variable nodes as $\mathcal{V}$. Variable nodes consist of a name and an entity type to support querying for entity types. We also support the entity type *All* to query for arbitrary entities. We write variable nodes by a leading question mark. Hence, a narrative query graph might include entities stemming from $\mathcal{E}$ and variable nodes from $\mathcal{V}$. Formally, a **fact pattern** is a triple $fp = (s, p, o)$ where $s, o \in (\mathcal{E} \cup \mathcal{V})$ and $p \in \Sigma$. A **narrative query graph** q is a set of fact patterns similar to SPARQL's basic graph patterns [15]. When executed, the query produces one or more matches μ by binding the variable symbols to actual entities, i.e., $\mu : \mathcal{V} \to \mathcal{E}$ is a partial function. If several fact patterns are queried, all patterns must be contained within a document forming a proper query answer. If queries include entities that are classes and have subclasses, then the query will be expanded to also query for these subclasses, i.e., direct and transitive subclasses.

4 Narrative Document Retrieval

In the following section we describe our system for narrative query graph processing. First, we perform a pre-processing that involves entity linking, information extraction, cleaning, and loading. It extracts document graphs from text and stores them in a structured repository. Then, a query processing that matches a user's query against the document graphs takes place. In this way, we can return a structured visualization of matching documents. An overview of the whole system is depicted in Figure 1.

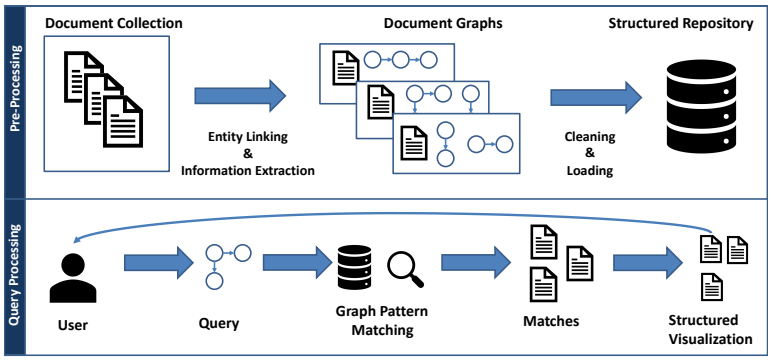


Fig. 1. System Overview: Document graphs are extracted from texts, cleaned, indexed, and loaded into a structured repository. Then, narrative query graphs can be matched against the repository to retrieve the respective documents.

4.1 Document Graph Extraction

The pre-processing step, including entity linking and information extraction, utilizes our toolbox for the nearly-unsupervised construction of knowledge graphs [10]. The toolbox requires the design of two different vocabularies: 1. An entity vocabulary that contains all entities of interest. An entry consists of a unique entity id, an entity name, and a list of synonyms. 2. A relation vocabulary that contains all relations of interest. An entry consists of a relation and a set of synonyms.

For this paper, we built an entity vocabulary that comprises *drugs, diseases, dosage forms, excipients, genes, plant families, and species*. Next, we wanted to extract interactions between these entities from texts since interactions between entities are essential to support retrieval with narrative query graphs. Although the quality of existing open information extraction like OpenIE 6 sounded promising [6], we found that open information extraction methods highly lack recall when processing biomedical texts, see the evaluation in [10]. That is why we developed a recall-oriented extraction technique **PathIE** in [10] that flexibly extracts interactions between entities via a path-based method. This method was evaluated and shared in our toolbox as well.

PathIE yields many synonymous predicates (treats, aids, prevents, etc.) that represent the relation *treats*. The relation vocabulary must have clear semantics and was built with the help of two domain experts. We designed a relation vocabulary comprising 60 entries (10 relations plus 50 synonyms) for the cleaning step. This vocabulary enables the user to formulate her query based on a well-curated vocabulary of entity interactions in the domain of interest. We applied our semi-supervised predicate unification algorithm to clean the extractions. To increase the quality of extractions, we introduced type constraints by providing fixed domain and range types for each interaction. Extracted interactions that did not meet the interaction's type constraints were removed. For example, the interaction *treats* is typed, i.e., the subject must be a drug, and the object must be a disease or species. Some interactions in our vocabulary like *induces* or *associated* are more general and thus were not annotated with type constraints.

4.2 Document Retrieval

Finally, the extracted document graphs had to be stored in a structured repository for querying purposes. For this paper, we built upon a relational database, namely PostgresV10. Relational databases support efficient querying and allowed us to provide additional provenance information and metadata for our purposes. For example, our prototype returned document titles, sentences, entity annotations, and extraction information to explain matches to the user. Due to our focus on pharmaceutical and medical users, we selected a PubMed subset that includes drug and excipient annotations. Therefore, we annotated the whole PubMed collection with our entity linking component, yielding 302 million annotations. Around six million documents included a drug or excipient annotation. Performing extraction and cleaning on around six million documents yielded nearly 270 million different extractions. Hence, the current prototype's version comprises about six million documents. We incrementally have increased the available data, but we entirely covered the relevant pharmaceutical part (drug and excipient).

As a reminder, a narrative query graph consists of fact patterns following simple RDF-style basic graph patterns. Our system automatically translates these narrative query graphs into a structured query language: They are translated into SQL statements for querying the underlying relational database. A single fact pattern requires a selection of the extraction table with suitable conditions to check the entities and the interaction. Multiple fact patterns require self-joining of the extraction table, and adding document conditions in the where clause, i.e., the facts matched against the query must be extracted from the same document. We developed an in-memory and hash-based matching algorithm that quickly combines the results. Another point to think about were ontological subclass relations between entities. For example, querying for treatments of *Diabetes Mellitus* would require to also search for the subclasses *Diabetes Mellitus Type 1* and *Diabetes Mellitus Type 2*. Query rewriting is necessary to compute complete results for queries that involve entities with subclasses [11]. We rewrite queries that include entities with subclasses to also query for these subclasses. Due to

the long-standing development of databases, such a query processing can be performed very quickly when using suitable indexes. We computed an inverted index, i.e., each extraction triple was mapped to a set of document ids. Besides, we implemented some optimization strategies to accelerate the query processing, e.g., match fact patterns with concrete entities first and fact patterns with variable nodes afterward. We remark on our system's query performance in our evaluation.

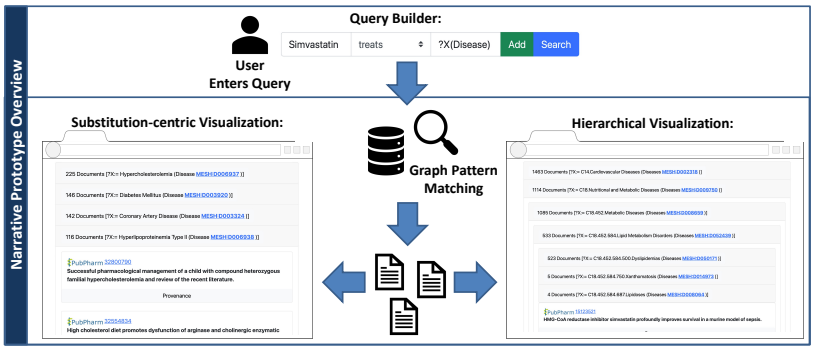


Fig. 2. A schematic overview of our prototype implementation. A query builder helps the users to formulate their information need. If the narrative query involves variable nodes, the results can be visualized in a substitution-centric visualization (left side) or in a hierarchical visualization (right side).

4.3 Prototype Design

We present a prototype resulting from joint efforts by the university library, the institute for information systems, and two pharmaceutical domain experts who gave us helpful feedback and recommendations. The prototype[1] offers precise biomedical document retrieval with narrative query graphs. A general overview of our prototype is shown in Fig. 2. We implemented a REST service handling queries and performing the query processing on the backend side. Furthermore, we supported the user with a query builder and suitable result visualization on the frontend side. In an early prototype phase, we tested different user interfaces to formulate narrative query graphs, namely, 1. a simple text field, 2. a structured query builder, and 3. a graph designer tool. We found that our users preferred the structured query builder which allows them to formulate a query by building a list of fact patterns. For each fact pattern, the users must enter the query's subject and object. Then, they can select an interaction between both in a predefined selection. The prototype assists the user by suggesting around three million terms (entity names plus synonyms). Variable nodes can be formulated, e.g., by writing *?X(Drug)* or just entering the entity type like *Drug* in

[1] http://www.pubpharm.de/services/prototypes/narratives/

the subject or object field. When users start their search, the prototype sends the query to the backend and visualizes the returned results. The returned results are sorted by their corresponding publication date in descending order. The prototype represents documents by a document id (PubMedID), a title, a link to the digital library entry (PubMed), and provenance information. Provenance includes the sentence in which the matching fact was extracted. We highlight the linked entities (subject and object) and their interaction (text term plus mapping to the interaction vocabulary). Provenance may be helpful for users to understand why a document is a match. If a query contains multiple fact patterns, we attach a list of matched sentences in the visualization. Visualizing document lists is comparable to traditional search engines, but handling queries with variable nodes requires novel interfaces. We will discuss such visualizations for queries, including variable nodes, subsequently.

4.4 Retrieval with Variable Nodes

Variable nodes in a narrative query graph may be restricted to specific entity types like *Disease*. We also allow a general type *All* to support querying for arbitrary entities. For example, a user might formulate the query (*Simvastatin*, *treats*, *?X(Disease)*). Several document graphs might match the query with different variable substitutions for $?X$. A document d_1 with the substitution $\mu_1(?X) =$ *hypercholestorelemia* as well as a document d_2 with $\mu_2(?X) =$ *hyperlipidemia* might be proper matches to the query. How should we handle and present these substitutions to the users? Discussions with domain experts led to the conclusion that aggregating documents by their substitution seems most promising. Further, we present two strategies to visualize these document result groups in an user interface: *substitution-centric* and *hierarchical visualization*.

Substitution-centric Visualization. Given a query with a variable node, the first strategy is to aggregate by similar variable substitutions. We retrieve a list of documents with corresponding variable substitutions from the respective document graph. Different substitutions represent different groups of documents, e.g., one group of documents might talk about the treatment of *hypercholestorelemia* while the other group might talk about *hypertriglyceridemia*. These groups are sorted in descending order by the number of documents in each group. Hence, variable substitutions shared by many documents appear at the top of the list. Our query prototype visualizes a document group as a collapsible list item. A user's click can uncollapse the list item to show all contained documents. Provenance information is used to explain why a document matches her query, i.e., the prototype displays the sentences in which a query's pattern was matched. Provenance may be especially helpful when working with variable nodes.

Hierarchical Visualization. Entities are arranged in taxonomies in many domains. Here, diseases are linked to MeSH (Medical Subject Heading) descriptors arranged in the MeSH taxonomy. The hierarchical visualization aims at showing document results in a hierarchical structure. For example, *hypercholestorelemia* and *hypertriglyceridemia* share the same superclass in MeSH, namely *hyperlipidemias*. All documents describing a treatment of *hypercholestorelemia* as well as

hypertriglyceridemia are also matches to *hyperlipidemias*. Our prototype visualizes this hierarchical structure by several nested collapsible lists, e.g., *hyperlipidemias* forms a collapsible list. If a user's click uncollapses this list, then the subclasses of *hyperlipidemias* are shown as collapsible lists as well. We remove all nodes that do not have any documents attached in their node or all successor nodes to bypass the need to show the whole MeSH taxonomy.

5 System Evaluation & User Study

Subsequently, we analyze our retrieval prototype concerning two research questions: *Do narrative query graphs offer a precise search for literature? And, do variable nodes provide useful entity-centric overviews of literature?* We performed three evaluations to answer the previous questions:

1. Two pharmaceutical experts created test sets to quantify the retrieval quality (100 abstracts and 50 full-text papers). Both experts are highly experienced in pharmaceutical literature search.
2. We performed interviews with eight pharmaceutical experts who search for literature in their daily research. Each expert was interviewed twice: Before testing our prototype to understand their information need and introducing our prototype. After testing our prototype, to collect feedback on a qualitative level, i.e., how they estimate our prototype's usefulness.
3. Finally, all eight experts were asked to fill out a questionnaire. The central findings are reported in this paper.

5.1 Retrieval Evaluation

After having consulted the pharmaceutical experts, we decided to focus on the following typical information needs in the biomedical domain: I1: Drug-Disease treatments (*treats*) play a central role in the mediation of diseases. I2: Drugs might decrease the effect of other drugs and diseases (*decrease*). I3: Drug treatments might increase the expression of some substance or disease (*induces*). I4: Drug-Gene inhibitions (*inhibits*), i.e., drugs disturb the proper enzyme production of a gene. I5: Gene-Drug metabolisms (*metabolizes*), i.e., gene-produced enzymes metabolize the drug's level by decreasing the drug's concentration in an organism. Narrative query graphs specify the exact interactions a user is looking for. For each information need (I1-5), we built narrative query graphs with well-known entities from the pharmaceutical domain: Q1: *Metformin treats Diabetes Mellitus (I1)*, Q2: *Simvastatin decreases Cholesterol (I2)*, Q3: *Simvastatin induces Rhabdomyolysis (I3)*, Q4: *Metformin inhibits mtor (I4)*, Q5: *CYP3A4 metabolizes Simvastatin AND Erythromycin inhibits CYP3A4 (I4/5)*, and Q6: *CYP3A4 metabolizes Simvastatin AND Amiodarone inhibits CYP3A4 (I4/5)*.

Further, we used the entities for each query to search for document candidates on PubMed, e.g., for Q1 we used *metformin diabetes mellitus* as the PubMed

Table 1. Expert evaluation of retrieval quality for narrative query graphs in comparison to PubMed and a MeSH-based search on PubMed. Two experts have annotated PubMed samples to estimate whether the information need was answered. Then, precision, recall and F1-measure are computed for all systems.

Query	**#Hits**	**#Sample**	**#TP**	**PubMed**	**MeSH Search**			**Narrative QG**		
				Prec.	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
Q1	12.7K	25	19	0.76	0.82	0.47	0.60	**1.00**	0.42	0.59
Q2	5K	25	16	0.64	**0.73**	0.50	0.59	0.66	0.25	0.36
Q3	427	25	17	0.68	0.77	0.59	0.67	**1.00**	0.35	0.52
Q4	726	25	16	0.64	**0.78**	0.44	0.56	0.71	0.31	0.43
Q5	397	25	6	0.24	-	-	-	**1.0**	0.17	0.25
Q6	372	25	5	0.20	-	-	-	**1.0**	0.20	0.33

query. We kept only documents that were processed in our pipeline. Then, we took a random sample of 25 documents for each query. The experts manually read and annotated these sample documents' abstracts concerning their information need (true hits / false hits). Besides, we retrieved 50 full texts documents of PubMed Central (PMC) for a combined and very specialized information need (Q5 and Q6). The experts made their decision for PubMed documents by considering titles and abstracts, and for PMC documents, the full texts. Subsequently, we considered these documents as ground truth to estimate the retrieval quality. We compared our retrieval to two baselines, 1) queries on PubMed and 2) queries on PubMed with suitable MeSH headings and subheadings.

PubMed MeSH Baseline. PubMed provides so-called MeSH terms for documents to assists users in their search process. MeSH (Medical Subject Headings) is an expert-designed vocabulary comprising various biomedical concepts (around 26K different headings). These MeSH terms are assigned to PubMed documents by human annotators who carefully read a document and select suitable headings. Prime examples for these headings are annotated entities such as drugs, diseases, etc., and concepts such as study types, therapy types, and many more. In addition to headings, MeSH supports about 76 subheadings to precisely annotate how a MeSH descriptor is used within the document's context. An example document might contain the subheading *drug therapy* attached to *simvastatin*. Hence, a human annotator decided that *simvastatin* is used in drug therapy within the document's context. The National Library of Medicine (NLM) recommends subheadings for entity interactions such as treatments and adverse effects. In cooperation with our experts who read the NLM recommendations, we selected suitable headings and subheadings to precisely query PubMed concerning the respective entity interaction for our queries.

Results. The corresponding interaction and the retrieval quality (precision, recall, and F1-score) for each query are depicted in Table 1. The sample size and the number of positive hits in the sample (TP) are reported for each query. The PubMed search contains only the entities as a simple baseline, and hence, achieved a recall of 1.0 in all cases. PubMed search yielded a precision of around

0.64 up to 0.76 for abstracts and 0.2 up to 0.24 for full texts. The PubMed MeSH search achieved a moderate precision of about 0.73 to 0.82 and recall of about 0.5 for PubMed titles and abstracts (Q1-Q4). Unfortunately, the important MeSH annotations were missing for all true positive hits for Q5 and Q6 in PMC full texts. Hence, the PubMed MeSH search did not find any hits in PMC for Q5 and Q6. Narrative query graphs (Narrative QG) answered the information need with good precision: Q1 (*treats*) and Q3 (*induces*) were answered with a precision of 1.0 and a corresponding recall of 0.42 (Q1) and 0.47 (Q3). The minimum achieved precision was 0.66, and the recall differed between 0.17 and 0.42. Our prototype could answer Q5 and Q6 on PMC full texts: One correct match was returned for Q5 as well as for Q6, leading to a precision of 1.0.

5.2 User Interviews

The previous evaluation demonstrated that our system could achieve good precision when searching for specialized information needs. However, the next questions are: How does our prototype work for daily use cases? And, what are the prototype's benefits and limitations in practice? Therefore, we performed two interviews with each of the eight pharmaceutical experts who search for literature in their daily work. All experts had a research background and worked either at a university or university hospital.

First Interview. In the first interview, we asked the participants to describe their literature search. They shared two different scientific workflows that we have analyzed further: 1. searching for literature in a familiar research area, and 2. searching for a new hypothesis which they might have heard in a talk or read in some paper. We performed think-aload experiments to understand both scenarios. They shared their screen, showed us at least two different literature searches, and how they found relevant documents answering their information need. For scenario 1), most of them knew suitable keywords, works or journals already. Hence, they quickly found relevant hits using precise keywords and sorting the results by their publication date. They already had a good overview of the literature and could hence answer their information need quickly. For scenario 2), they guessed keywords for the given hypothesis. They had to refine their search several times by varying keywords, adding more, or removing keywords. Then, they scanned titles and abstracts of documents looking for the given hypothesis. We believe that scenario 1) was recall-oriented: They did not want to miss important works. Scenario 2) seemed to be precision-oriented, i.e., they quickly wanted to check whether the hypothesis may be supported by literature. Subsequently, we gave them a short introduction to our prototype. We highlighted two features: The precision-oriented search and the usage of variable nodes to get entity-centric literature overviews. We closed the first interview and gave them three weeks to use the prototype for their literature searches.

Second Interview. We asked them to share their thoughts about the prototype: What works well? What does not work well? What could be improved? First, they considered querying with narrative query graphs, especially with variable nodes, different and more complicated than keyword-based searches.

Querying with variable nodes by writing *?X(Drug)* as a subject or an object was deemed too cryptic. They suggested that using *Drug*, *Disease*, etc. would be easier. Another point was that they were restricted to a fixed set of subjects and objects (all known entities in our prototype). For example, querying with pharmaceutical methods like *photomicrography* was not supported. Next, the interaction vocabulary was not intuitive for them. Sometimes they did not know which interaction would answer their information need. One expert suggested to introduce a hierarchical structure for the interactions, i.e., some general interactions like *interacts* that can be specified into *metabolizes* and *inhibits* if required. On the other side, they appreciated the prototype's precise search capability. They all agreed that they could find precise results more quickly using our prototype than other search engines. Besides, they appreciated the provenance information to estimate if a document match answers their information need. They agreed that variable nodes in narrative query graphs offered completely new search capabilities, e.g., *In which dosage forms was Metformin used when treating diabetes?* Such a query could be translated into two fact patterns: (*Metformin*, *administered*, *?X(DosageForm)* and (*Metformin*, *treats*, *Diabetes Mellitus*). The most common administrations are done *orally* or via an *injection*. They agreed that such information might not be available in a specialized database like DrugBank. DrugBank covers different dosage forms for Metformin but not in combination with diabetes treatments. As queries get more complicated and detailed, such information can hardly be gathered in a single database. They argued that the *substitution-centric visualization* helps them to estimate which substitutions are relevant based on the number of supporting documents. Besides, they found the *hierarchical visualisation* helpful when querying for diseases, e.g., searching for (*Metformin*, *treats*, *?X(Disease)*). Here, substitutions are shown in an hierarchical representation, e.g., *Metabolism Disorders*, *Glucose Disorders*, *Diabetes Mellitus*, *Diabetes Mellitus Type 1*, etc. They liked this visualization to get a drug's overview of treated disease classes. All of them agreed that searches with variable nodes were helpful to get an entity-structured overview of the literature. Four experts stated that such an overview could help new researchers get better literature overviews in their fields.

5.3 Questionnaire

We asked each domain expert to answer a questionnaire after completing the second interview. The essential findings and results are reported subsequently. First, we asked to choose between precision and recall when searching for literature. Q1: *To which statement would you rather agree when you search for related work?* The answer options were (rephrased): A1a: *I would rather prefer a complete result list (recall). I do not want to miss anything.* A2a: *I would rather prefer precise results (precision) and accept missing documents.* Six of eight experts preferred recall, and the remaining two preferred precision. We asked a similar question for the second scenario (hypothesis). Again, we had let them select between precision and recall (A1a and A1b). Seven of eight preferred precision, and one preferred recall when searching for a hypothesis. Then,

Table 2. Questionnaire Results: Eight participants were asked to rate the following statements about our prototype on a Likert scale ranging from 1 (disagreement) to 5 (agreement). The mean ratings are reported.

Statement about the Prototype	**Mean**
The prototype allows me to formulate precise questions by specifically expressing the interactions between search terms.	4.0
The formulation of questions in the prototype is understandable for me.	4.0
The displayed text passage from the document (Provenance) is helpful for me to understand why a document matches my search query.	5.0
The prototype provides precise results for my questions (I quickly find a relevant match).	3.5
Basically, grouping results is helpful for me when searching for variable nodes.	4.5
When searching for related work, I would prefer the prototype to a search using classic search tools (cf. PubPharm, PubMed, etc.).	2.8
When searching for or verifying a hypothesis, I would prefer the prototype to a search using classic search tools (cf. PubPharm, PubMed, etc.).	3.4
I could imagine using the prototype in my literature research.	3.9

we asked Q3: *To which statement would you rather agree for the vast majority of your searches?* Again, seven of eight domain experts preferred precise hits over complete result lists. The remaining one preferred recall. The next block of questions was about individual searching experiences with our prototype: different statements were rated on a Likert scale ranging from 1 (disagreement) to 5 (full agreement). The results are reported in Table 2. They agreed that the prototype allows to formulate precise questions (4.0 mean rating), and the formulation of questions was understandable (4.0). Besides, provenance information was beneficial for our users (5.0). They could well imagine using our prototype in their literature research (3.9) and searching for a hypothesis (3.4). Still, users were reluctant to actually switch to our prototype for related work searches (2.8). Finally, the result visualization of narrative query graphs with variables was considered helpful (4.5).

5.4 Performance Analysis

The query system and the database ran on a server, having two Intel Xeon E5-2687W (3,1GHz, eight cores, 16 threads), 377GB of DDR3 main memory, and SDDs as primary storage. The preprocessing took around one week for our six million documents (titles and abstracts). We randomly generated 10k queries asking for one, two, and three interactions. We measured the time of query execution on a single thread. Queries that are not expanded via an ontology took in average 21.9ms (1-fact) / 52ms (2-facts) / 51.7ms (3-facts). Queries that are expanded via an ontology took in average 54.9ms (1-fact) / 158.9ms (2-facts) / 158.2ms (3-facts). However, the query time heavily depends on the interaction (selectivity) and how many subclasses are involved. In sum, our system can retrieve documents with a quick response time for the vast majority of searches.

6 Discussion and Conclusion

In close cooperation with domain experts using the PubMed corpus, our evaluation shows that overall document retrieval can indeed decisively profit from graph-based querying. The expert evaluation demonstrates that our system achieves a moderate up to good precision for highly specialized information needs in the pharmaceutical domain. Although the precision is high, our system has only a moderate recall. Moreover, we compared our system to manually curated annotations (MeSH and MeSH subheadings), which are a unique feature of PubMed. Most digital libraries may support keywords and tags for documents but rarely support how these keywords, and primarily, how entities are used within the document's context. Therefore, we developed a document retrieval system with a precision comparable to manual metadata curation but without the need for manual curation of documents.

The user study and questionnaire reveal a strong agreement for our prototype's usefulness in practice. In summary, the user interface must be intuitive to support querying with narrative query graphs. Further enhancements are necessary to explain the interaction vocabulary to the user. We appreciate the idea of hierarchical interactions, i.e., showing a few basic interactions that can be specified for more specialized needs. Especially the search with variable nodes in detailed narrative query graphs offers a new access path to the literature. The questionnaire reveals that seven of eight experts agreed that the vast majority of their searches are precision-oriented. Next, they agreed that they prefer our prototype over established search engines for precision-oriented searches. The verification of hypotheses seems to be a possible application because precise hits are preferred here. We believe that our prototype should not replace classical search engines because there are many recall-oriented tasks like related work searches. The recall will always be a problem by design when building upon error-prone natural language processing techniques and restricting extractions to sentence levels. Although the results seem promising, there are still problems to be solved in the future, e.g., improve the extraction and the user interface.

Conclusion. Entity-based information access catering even for complex information needs is a central necessity in today's scientific knowledge discovery. But while structured information sources such as knowledge graphs offer *high query expressiveness* by graph-based query languages, scientific document retrieval is severely lagging behind. The reason is that graph-based query languages allow to describe the desired characteristics of and interactions between entities in sufficient detail. In contrast, document retrieval is usually limited to simple keyword queries. Yet unlike knowledge graphs, scientific document collections offer *contextualized knowledge*, where entities, their specific characteristics, and their interactions are connected as part of a coherent argumentation and thus offer a clear advantage [7,8]. The research in this paper offers a novel workflow to bridge the worlds of structured and unstructured scientific information by performing graph-based querying against scientific document collections. But as our current workflow is clearly precision-oriented, we plan to improve the recall without having to broaden the scope of queries in future work.

References

1. Azad, H.K., Deepak, A.: Query expansion techniques for information retrieval: A survey. Information Processing & Management **56**(5), 1698–1735 (2019)
2. Betts, C., Power, J., Ammar, W.: GrapAL: Connecting the dots in scientific literature. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 147–152. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-3025
3. Dietz, L., Kotov, A., Meij, E.: Utilizing knowledge graphs for text-centric information retrieval. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. p. 1387–1390. SIGIR '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3209978.3210187
4. Herskovic, J.R., Tanaka, L.Y., Hersh, W., Bernstam, E.V.: A Day in the Life of PubMed: Analysis of a Typical Day's Query Log. Journal of the American Medical Informatics Association **14**(2), 212–220 (03 2007)
5. Kadry, A., Dietz, L.: Open relation extraction for support passage retrieval: Merit and open issues. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 1149–1152. SIGIR '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3077136.3080744
6. Kolluru, K., Adlakha, V., Aggarwal, S., Mausam, Chakrabarti, S.: OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction. In: Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP). pp. 3748–3761. ACL (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.306
7. Kroll, H., Kalo, J.C., Nagel, D., Mennicke, S., Balke, W.T.: Context-compatible information fusion for scientific knowledge graphs. In: Digital Libraries for Open Knowledge. pp. 33–47. Springer (2020). https://doi.org/10.1007/978-3-030-54956-5_3
8. Kroll, H., Nagel, D., Balke, W.T.: Modeling narrative structures in logical overlays on top of knowledge repositories. In: Conceptual Modeling. pp. 250–260. Springer (2020). https://doi.org/10.1007/978-3-030-62522-1_18
9. Kroll, H., Nagel, D., Kunz, M., Balke, W.T.: Demonstrating narrative bindings: Linking discourses to knowledge repositories. In: Fourth Workshop on Narrative Extraction From Texts, Text2Story@ECIR2021. CEUR Workshop Proceedings, vol. 2860, pp. 57–63. CEUR-WS.org (2021), http://ceur-ws.org/Vol-2860/paper7.pdf
10. Kroll, H., Pirklbauer, J., Balke, W.T.: A toolbox for the nearly-unsupervised construction of digital library knowledge graphs. In: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2021. JCDL '21, Association for Computing Machinery, New York, NY, USA (2021)
11. Krötzsch, M., Rudolph, S.: Is your database system a semantic web reasoner? KI-Künstliche Intelligenz **30**(2), 169–176 (2016). https://doi.org/10.1007/s13218-015-0412-x
12. Manola, F., Miller, E., McBride, B., et al.: RDF primer. W3C recommendation **10**(1-107), 6 (2004)
13. Mohan, S., Fiorini, N., Kim, S., Lu, Z.: A fast deep learning model for textual relevance in biomedical information retrieval. In: Proceedings of the 2018

World Wide Web Conference. p. 77–86. WWW '18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2018). https://doi.org/10.1145/3178876.3186049
14. Nguyen, D.B., Abujabal, A., Tran, N.K., Theobald, M., Weikum, G.: Query-driven on-the-fly knowledge base construction. Proc. VLDB Endow. **11**(1), 66–79 (Sep 2017). https://doi.org/10.14778/3151113.3151119
15. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of sparql. ACM Transactions on Database Systems **34**(3) (Sep 2009). https://doi.org/10.1145/1567274.1567278
16. Raviv, H., Kurland, O., Carmel, D.: Document retrieval using entity-based language models. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 65–74. SIGIR '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2911451.2911508
17. Spitz, A., Gertz, M.: Terms over load: Leveraging named entities for cross-document extraction and summarization of events. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 503–512. SIGIR '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2911451.2911529
18. Vazirgiannis, M., Malliaros, F.D., Nikolentzos, G.: Graphrep: Boosting text mining, nlp and information retrieval with graphs. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. p. 2295–2296. CIKM '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3269206.3274273
19. Xiong, C., Power, R., Callan, J.: Explicit semantic ranking for academic search via knowledge graph embedding. In: Proceedings of the 26th International Conference on World Wide Web. p. 1271–1279. WWW '17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2017). https://doi.org/10.1145/3038912.3052558
20. Zhao, S., Su, C., Sboner, A., Wang, F.: Graphene: A precise biomedical literature retrieval engine with graph augmented deep learning and external knowledge empowerment. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. p. 149–158. CIKM '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3357384.3358038

B.4. JCDL 2022a: What a Publication Tells You – Benefits of Narrative Information Access

JCDL'22a

Hermann Kroll, Florian Plötzky, Jan Pirklbauer, and Wolf-Tilo Balke. "What a Publication Tells You – Benefits of Narrative Information Access in Digital Libraries". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Cologne, Germany, 2022, ACM. DOI: https://doi.org/10.1145/3529372.3530928

What a Publication Tells You – Benefits of Narrative Information Access in Digital Libraries

Hermann Kroll
kroll@ifis.cs.tu-bs.de
Institute for Information Systems, TU Braunschweig
Braunschweig, Lower Saxony, Germany

Florian Plötzky
ploetzky@ifis.cs.tu-bs.de
Institute for Information Systems, TU Braunschweig
Braunschweig, Lower Saxony, Germany

Jan Pirklbauer
j.pirklbauer@tu-bs.de
Institute for Information Systems, TU Braunschweig
Braunschweig, Lower Saxony, Germany

Wolf-Tilo Balke
balke@ifis.cs.tu-bs.de
Institute for Information Systems, TU Braunschweig
Braunschweig, Lower Saxony, Germany

ABSTRACT

Knowledge bases allow effective access paths in digital libraries. Here users can specify their information need as graph patterns for precise searches and structured overviews (by allowing variables in queries). But especially when considering textual sources that contain narrative information, i.e., short stories of interest, harvesting statements from them to construct knowledge bases may be a serious threat to the statements' validity. A piece of information, originally stated in a coherent line of arguments, could be used in a knowledge base query processing without considering its vital context conditions. And this can lead to invalid results. That is why we argue to move towards narrative information access by considering contexts in the query processing step. In this way digital libraries can allow users to query for narrative information and supply them with valid answers. In this paper we define narrative information access, demonstrate its benefits for Covid 19 related questions, and argue on the generalizability for other domains such as political sciences.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; *Information integration*; **Web searching and information discovery**.

KEYWORDS

Narrative Information Access, Information Retrieval, Digital Libraries

ACM Reference Format:
Hermann Kroll, Florian Plötzky, Jan Pirklbauer, and Wolf-Tilo Balke. 2022. What a Publication Tells You – Benefits of Narrative Information Access in Digital Libraries. In *The ACM/IEEE Joint Conference on Digital Libraries in 2022 (JCDL '22), June 20–24, 2022, Cologne, Germany.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3529372.3530928

JCDL '22, June 20–24, 2022, Cologne, Germany

ACM ISBN 978-1-4503-9345-4/22/06…$15.00
https://doi.org/10.1145/3529372.3530928

1 INTRODUCTION

From the beginnings of human language, knowledge was shared and passed on following a narrative oral tradition, i.e., they exchange stories and have structured debates and conversations [16]. With the advent of written language, these oral presentations were made persistent by writing up stories, comments and discussions in articles and books. The central way to encode all this knowledge is to tell a story: a narrator relates what was observed and how more complex conclusions were derived from basic claims. We thus understand this process as *composing narratives*, i.e., action patterns bound to real-world entities or concepts to form rich lines of arguments [7].

Today digital libraries play a key role in making knowledge publicly available in large-scale repositories. The necessary curation builds on a long-standing library sciences tradition and results in a variety of novel digital technologies to manage and access knowledge repositories, including the FAIR principles [27]: On the one hand, extensive collections need to be effectively maintained and efficiently archived. Here additional metadata enrichment is already used in each source to prepare the data for later access (Findability & Accessibility). On the other hand, digital libraries face an increasing amount of data collected from distributed sources. This is done either by providing unifying interfaces to individual collections of linked open data or by using information integration techniques over extractions from different sources (Interoperability & Reuse).

The traditional solution is to provide a simple keyword-based access path to the underlying data. Then users have to retrieve this data and *determine* what is actually *told* by the data. What happens here is that users try to *understand* the data to reuse the information of interest for their purposes. We understand this *exploratory process of understanding* as *gradually composing narratives*, in the sense of extracting and generalizing patterns that are 'told' by the data. Take, for instance, the COVID-19 pandemic: Patient records might describe suffered conditions after they have been vaccinated by a SARS-CoV-2 vaccine. Biomedical experts can then read through these records and extract typical story patterns, e.g., patients may experience headaches and pain, or even worse, may suffer from dangerous cerebral sinus venous thrombosis. Although this manual workflow is common, the rapid speed of the COVID-19 pandemics has shown that, given the amount of data available, it is hard to stay

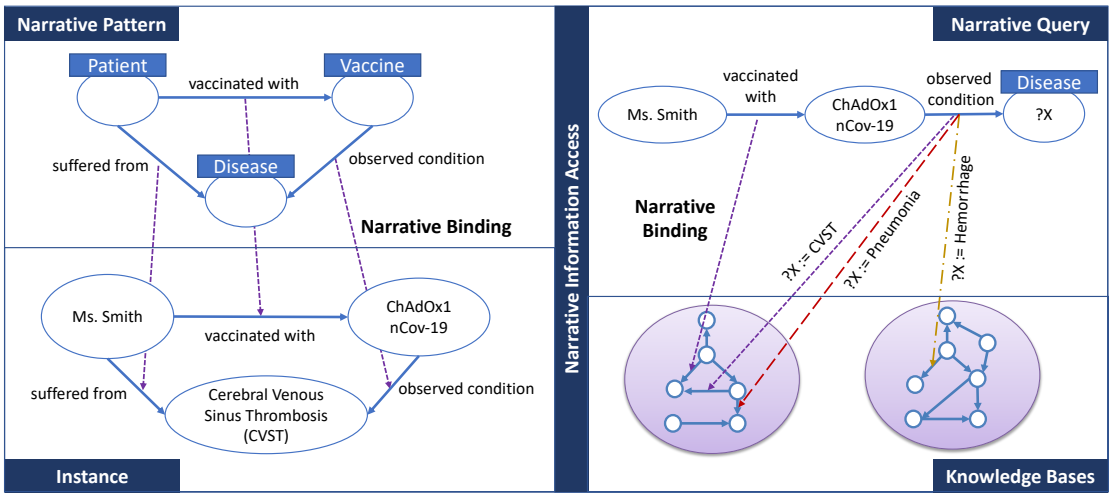


Figure 1: Systematic overview: A narrative pattern (upper left corner) describes a template how different entity types interact with each other. An instance then substitutes the entity types by concrete entities (lower left corner). These substitutions are called narrative bindings. On the right, the narrative query processing is depicted: Narrative bindings are found for each statement of a narrative query. Bindings that share the same context are depicted in the same colour and shape.

up-to-date. Even when restricting information sources only to well-curated ones, researchers would have to cover nearly 200k peer-reviewed articles about COVID-19 published in the US National Library of Medicine over the last two years[1].

Such rapid developments ask for novel and more efficient access methods. For example, a comprehensive database of all possible conditions observed in COVID-19 vaccinations might be helpful for improved diagnostics. Yet, when building such a knowledge base by harvesting statements about COVID-19 from textual sources, the answer quality may not be sufficient in practice. This is because the *observed conditions* are torn from the original course of vaccination as exhibited by some concrete patient. For example, some conditions might only be observed in elderly patients and thus, might not apply to children, or some complications might only be possible when a certain pre-existing condition is present in a patient. This means that although each condition was correctly extracted, the reusing of the resulting statements in a knowledge base may not be valid because the information's *contexts* do not match. When humans read through publications and retrieve arguments, they usually consider all essential context conditions such as the treated group or relevant pre-existing conditions. Moreover, in addition to contexts, humans also consider the connection between statements within a line of argument, e.g., do the assumptions within the arguments leading to a conclusion actually make sense together?

We argue that digital libraries need to move towards narrative information access, i.e., to offer query capabilities in the form of narrative patterns while considering vital contexts. Therefore we first define narrative information access. We then argue on contexts and how digital libraries can retain them. In addition, we perform two case studies on top of our narrative retrieval system, published last year [13]. We investigate COVID-19-related research questions in cooperation with domain experts. We also asked an expert from the political sciences domain to study the system and describe how the political sciences domain could benefit from such a retrieval system. Finally, we discuss the generalizability, benefits, and challenges of narrative information access for digital libraries.

[1]https://www.ncbi.nlm.nih.gov/research/coronavirus/

2 NARRATIVE INFORMATION ACCESS

In the following section we define the concept of narrative information access and discuss its key components. To ease understanding, we start with a running example from the biomedical field as a narrative pattern: Covid 19 vaccinations and their possible side effects. Consider the following short narrative:

Example 1. *Some patients that were vaccinated by ChAdOx1 nCov-19 Vaccine (also known as Astra Zeneca) suffered Cerebral Venous Sinus Thrombosis (CVST). Hence Intracranial Sinus Thrombosis is an observed disease condition for the ChAdOx1 nCov-19 vaccine.*

Three types of entities participate in this example: a vaccine, patients, and a disease condition. In addition, three possible relations between the entity types are expressed: patients *are vaccinated* with the vaccine, patients *suffer from* a disease condition, and the disease condition *is observed* for the vaccine. Thus narrative patterns are described by typing their participants and naming their relations (see Fig. 1). The following ideas are based on an eased version of a narrative model that we introduced in [11].

Based on the encoding of knowledge in the Resource Description Framework (RDF) [18], we define narrative patterns by:

Definition 1 (Narrative Pattern). *A narrative pattern is a connected, node- and edge-labeled directed graph, where each edge (labeled with a predicate name) represents a statement in the form of a (subject, predicate, object)-triple. Each node either represents a subject reflecting some entity type or an object reflecting either an entity type or literal values from a certain domain.*

Any knowledge base in RDF format can then be seen as a graph containing a collection of *instances of* narrative patterns as subgraphs, i.e., all nodes have been instantiated (either by URIs in the case of entities or by concrete literal values). We can now translate our previous example narrative using a narrative pattern as a kind of skeleton for the narrative. A possible instance is depicted in Fig. 1 (please note that for simplification, we replaced long URI prefixes with short entity names).

In brief, we have a graph representation of a concrete narrative structured by some narrative pattern. Hence narrative patterns can be understood as (sub-)graphs isomorphisms on RDF knowledge bases. We then define narrative queries using such patterns:

Definition 2 (Narrative Query). *A narrative query is a narrative pattern where each node is either instantiated by a concrete entity or literal value or replaced by a variable (labeled by a variable name).*

By design our proposed querying method has very similar semantics to querying RDF knowledge bases with SPARQL: If a narrative query does not contain a variable, then the answer is whether there exists an instance in the knowledge base that is isomorphic to the query's narrative pattern and features all the query's exact entities/literal values in the right places (cf. ASK queries in SPARQL). If a narrative query contains one or more variables, then these variables must be substituted by concrete entities from the knowledge base during query processing. Of course, all matches to the query must be valid with regard to variable substitutions, i.e., the substituted pattern and the respective entities/values must be contained in the knowledge base. We understand such a matching process as *binding a query* [12], i.e., we take some edge of the query's narrative pattern and bind it against a knowledge base edge and bind concrete entities and literal values to the respective entity types or literal domains in the pattern.

Returning to our example, we may query which *disease conditions* the *ChAdOx1 nCov-19* vaccinated patient *Smith* could possibly suffer from. The respective narrative query is depicted in Fig. 1. The first step to answer this query is to compute narrative bindings against the underlying knowledge base(s). We may find a binding b_1 confirming that *Ms. Smith* has been vaccinated with *ChAdOx1 nCov-19*. In addition, we must *substitute* the variable *?X (of type disease)*. Here we may find three bindings with suitable substitutions: b_2 (CVST), b_3 (Pneumonia), and b_4 (Hemorrhage). In common graph querying we would now join the intermediate results to list all conditions that *Ms. Smith* could possibly expect: CVST, pneumonia and hemorrhage.

Now, assume for the time being that *pneumonia* have only been observed in elderly people, whereas *Ms. Smith* is still young. Then *pneumonia* as a possible side effect of the vaccination might no longer apply to *Ms. Smith*, although the respective binding observing *pneumonia* as a possible side effect of a *ChAdOx1 nCov-19* vaccination is perfectly correct. The problem here is that b_3 would not be valid *in general*, because the observed conditions do not apply to all patients, but only to elderly patients. Although the bindings are correctly retrieved, not all of them might actually fit into the context of *Ms. Smith*.

Here information was torn apart regarding a sensitive context such as the target group information. One might argue that extracting RDF-style knowledge from individual patient records could even in the best case be problematic and should not be done in this way. While we agree that all patients are somewhat unique cases, this kind of extraction is common practice in real life applications, e.g., the *causes* relation in SemMedDB [8], *medical causes* in Wikidata [26][2], and *causes* in DBpedia [1][3].

The effect is that even if knowledge bases did only contain correct statements, fusing them to answer a query may still produce incorrect results. Indeed, it is a good scientific practice to arrange statements as complex lines of arguments, i.e., authors are sure to mention all essential contexts, settings, assumptions made, necessary conditions, hypotheses, experimental designs, etc. It is essential to fuse only those arguments fitting into the same context provided in the form of constraints by other arguments or the query terms. We call bindings *context-compatible* if they can safely be fused to form valid knowledge. Based on the idea of context-compatibility, we are now ready to propose a novel query processing method that considers contexts as constraints upon the query process to bypass the previous issues.

Definition 3 (Narrative Query Processing). *Given a narrative query and a set of knowledge bases, the query processing has to a) bind each individual query statement against underlying data of the knowledge base(s) and b) check the context-compatibility of the computed bindings. The result of the query process is thus a set of valid bindings, individually binding all query statements and being context-compatible.*

Thus narrative query processing ensures that contexts are considered while matching graph patterns. All bindings must in this way share a compatible context. And with this narrative query processing method we can now define narrative information access:

Definition 4 (Narrative Information Access). *Narrative Information Access allows users to formulate their information need as a narrative query. A narrative retrieval system then performs narrative query processing for this pattern and returns the results to the user. If results are found, we call the narrative pattern plausible.*

2.1 The Problem of Context-Compatibility

In this section we investigate the problem of context-compatibility in more detail and discuss suitable solutions how digital libraries can retain contexts in practice. Contexts define the *scope* in which a piece of information can be fused with other statements. This means that a context has to involve all information that need to be known to validate some larger, fused piece of information. But unfortunately, essential parts of contexts may get lost during information extraction. Generally speaking, problems with context compatibility come in at least two distinct flavors: *constraining contexts and correspondence contexts*. Constraining contexts scope the validity of fusions of statements over the entire query, i.e., for some statements in a substitution, a fusion is impossible because they have been extracted from contradicting contexts. In contrast, correspondence contexts limit the actual fusion of individual pieces

[2]https://www.wikidata.org/wiki/Property:P828
[3]https://dbpedia.org/property/causes

of knowledge between which a fusion would generally be possible but is not warranted by the data from which the information was extracted.

For a problematic case with *constraining contexts* consider the following example:

Example 2. *"We report a case of a 62-year-old man who developed cerebral venous sinus thrombosis with subarachnoid hemorrhage and concomitant thrombocytopenia, which occurred 13 days after ChAdOx1 nCov-19 injection." [2]*

Among others we may extract the following statements:

- *(patient, vaccinated by, ChAdOx1 nCov-19)*
- *(patient, suffered from, cerebral venous sinus thrombosis)*

But the statement that some patient suffered from cerebral venous sinus thrombosis is only sensible within the context of this particular patient record. Unfortunately, there is no information whether the statement can be generalized to other patients. Thus if the extractions' context (e.g., the patient's age, or that he was recently vaccinated) is lost, information fusions or reasoning processes relying on this specific piece of information may produce invalid results and even run into inconsistencies.

In brief, constructing knowledge bases with insufficiently contextualized statements and then using them to answer complex query patterns may result in invalid answers: Vaccinations with *ChAdOx1 nCov-19* may indeed lead to a *pneumonia* although probably not in all contexts.

For a problematic case with *corresponding contexts* consider the following example:

Example 3. *"Secondary analyses found increased risk of CVST after ChAdOx1 nCoV-19 vaccination (4.01, 2.08 to 7.71 at 8-14 days), after BNT162b2 mRNA vaccination (3.58, 1.39 to 9.27 at 15-21 days), and after a positive SARS-CoV-2 test." [9]*

We may extract the following statements:

- *(ChAdOx1 nCov-19, observed condition, CVST)*
- *(BNT162 Vaccine, observed condition, CVST)*
- *(CVST, risk after vaccination, 4.01)*
- *(CVST, risk after vaccination, 3.58)*

Now information fusion for answering the query (?x, observed condition, CVST) AND (CVST, risk after vaccination, ?y). would compute the Cartesian product producing four results (two of which are correct, while the other two are incorrect). This is because the binary extraction has lost the information, which risk factor belongs to which vaccine.

In brief, although all statements are mentioned within the close scope of a clinical trial having inclusion and exclusion criteria, an information extraction process may loose how statements belong together within that context.

Here the text expresses a ternary relation between *vaccines*, *conditions* and *probabilities* that is broken down into binary relations. Moreover, note that this is not an artifact of automatic processes, as even manual extraction may yield the same result because of the restriction of using only binary relations.

In conclusion, although all of our example statements were *syntactically* correct, vital *semantics* have been lost because the context was neglected. This forms a serious threat to the **validity** of query results, i.e., even correctly extracted but subsequently fused statements may not always produce valid answers in query processing or reasoning. Specifically, invalid answers are those cases that do not match the user's context or connect statements that do not belong together.

Since these problems are the main reason we argue to move towards narrative information access, we will take a closer look at possible remedies in the following section.

2.2 Maintaining Contexts in Digital Libraries

So how can we retain contexts in practical digital library projects? This subsection discusses research and methods to combat both loss of constraining contexts and loss of correspondence contexts.

N-ary Relations. Ernst et al. [6] proposed an n-ary extraction method to precisely retain complex relations, e.g., a relation *vaccinated_patients_suffer* that involves the *target group*, *vaccine* and *side effects*. However designing appropriate n-ary relation signatures a-priori is challenging because it requires extensive domain knowledge. The authors collected examples to train a suitable extraction model for their relations. In addition, they performed partial reasoning to compose partial statements to n-ary statements because their extraction method was also limited to sentences. The reasoning step helped to increase the extraction recall but required the definition of rules (which facts should be composed). Although n-ary relations are strongly appreciated, practical extraction methods hardly support them because defining signatures, providing enough training examples, and formulating reasoning constraints is an exhausting task.

Explicit Context Models. McCarthy introduced an explicit context model based on the first-order predicate logic [19]. The model allows users to formulate context conditions for arbitrary statements explicitly. In addition, he discussed relations between contexts, e.g., one context might *specialize* another context. Hand-crafted rules were then formulated to determine how to combine contexts and their enclosed statements. *VIKEF* is an example digital library project supporting explicit context information in an RDF knowledge base [23].

Implicit Contexts. We proposed using document references as an implicit and practical context model [10]. We suggested to store references to the source documents when harvesting statements from it. These references were then used to estimate which statements can safely be combined to produce valid answers. When combining only statements extracted from the same document, the resulting precision in a downstream application will increase, but the recall is bound to decrease. We therefore proposed measures to estimate *compatibility* between contexts to flexibly manage the precision/recall trade-off, e.g., text and author similarities.

Such implicit context models might be suitable candidates to retain context in digital libraries because they are cheap to maintain, i.e., only references to the statements' sources must be retained. But their quality and explainability are somewhat limited, e.g., how should we explain why two documents are context-compatible based on some text similarity measure. Keyword extraction might be a good method to retrieve context proxies here; See YAKE [5] for example. In summary, implicit context models are easy to use

and may yield good precision, but estimating context-compatibility remains challenging, and the overall quality achieved might still not be good enough for digital libraries.

Provenance. Provenance information is often understood to be any kind of information that may validate some statement's quality or origin [28]. Provenance might range from storing a reference to the statement's origin to storing information about the creation process, e.g., author, release date, point in time, and more. The Prov-O Ontology Description is a common standard for defining and storing general provenance information [17]. Prov-O supports complex provenance graphs to describe the origin of some statements. As an alternative, the Wikidata project supports qualifiers (property-value pairs) to retain provenance for its statements [26], e.g., references, determination methods, time and location information.

Nevertheless, using qualifiers and provenance information in practical applications, especially in query processing, remains an exception. Returning to our example, how could we use a qualifier information about the *62-year-old man* in query processing? Should we formulate hand-crafted rules on how different provenance information affects the actual query processing? How do we know when qualifiers describe the same or a compatible context? We understand Prov-O and provenance in general as possible implementations to store contexts. However they do not provide a ready-to-use solution to retain both by default. Domain experts and digital library curators must carefully define corresponding statements and describe how they are used for a practical application.

3 NARRATIVE QUERY PROCESSING IN PRACTICE – CASE STUDIES

We performed case studies to understand the benefits and limitations of narrative information access. In particular, we built on our publicly available narrative retrieval system called *Narrative Query Graphs for Entity-Interaction Document Retrieval* by [13]. We built a working document retrieval system that allows formulating information needs as graph patterns, i.e., entities and their corresponding interactions. We transformed biomedical document abstracts into a graph representation called document graph as knowledge bases. Then the retrieval system allows matching user queries against these document graphs and returns all matches. Since document graphs match queries only within single documents, contexts are to some degree considered in query processing because the context can quite safely be assumed to be consistent within each document abstract.

3.1 Narrative Query Graphs for Covid 19

In cooperation with pharmaceutical domain experts, the Robert-Koch Institute in Germany and the ZB MED library, we enhanced the narrative retrieval system to answer Covid 19-related research questions:

(1) We included the LitCovid collection from PubMed (peer-reviewed articles about Covid 19) and the latest Covid 19-related pre-prints supplied by ZB MED [14, 15]. These pre-prints can be accessed via their Preview service[4].

(2) We developed a vaccine entity vocabulary by utilizing Wikidata and the Medical Subject Headings (MeSH). In addition, we derived an entity for Long Covid 19 from MeSH.

The prototype of the enhanced narrative query system is publicly available[5]. In the following we investigate whether typical research questions from the pharmacy domain can be translated into narrative query graphs and how helpful such searches are in practice. Please note that this case study does not yet contain a comprehensive evaluation. We are currently preparing a large-scale study with our partners.

Long Covid Related Questions. The development of the Covid 19 pandemics has shown that *Long Covid* is a severe threat to a patient's health. So what are *common symptoms that are reported for Long Covid*? We formulated the following query graph: (*post-acute COVID-19 syndrome*, *associated*, *?X(Disease)*). *?X(Disease)* means that we search with a variable named *?X* that should be substituted by entities of the type *Disease*. *Post-acute COVID-19 syndrome* is an entity from the Medical Subject Headings (MeSH)[6]. The system responded with a list of commonly known conditions such as *Fatigue (44)*, *Dyspnea (19)*, *Anossmia (10)*, *Cognitive Dysfunction (9)* and *Headache (7)*. The number in brackets refers to how many documents share the corresponding variable substitution. The system can show the origin of the extraction, i.e., the sentence in which the pattern was matched. However also substitutions such as *Covid 19 (143)* and *Infections (61)* were not helpful.

We adjusted the previous query to search for patient cases: (*post-acute COVID-19 syndrome*, *associated*, *Human*) AND (*Human*, *associated*, *?X(Disease)*) . Here *Humans* is an entity that stand for *patients*, *men*, *women*, etc. The current version of the system did not support searching for specific target groups. This query could be matched against abstracts such as: *"[...] post-COVID-19 syndrome in patients with primary Sjogren's syndrome (pSS) affected by acute SARS-CoV-2 infection. [...] More than 40% of pSS patients reported the persistence of four symptoms or more, including anxiety/depression (59%), arthralgias (56%), sleep disorder (44%), fatigue (40%), anosmia (34%) and myalgias (32%)."* [3] Here the implicit context ensured that both statements must be matched against a single abstract. But the number of found results were decreased: *Fatigue (15)*, *Dyspnea (8)*, *Cognitive Dysfunction (4)* and *Headache (3)*.

A quick look over both results revealed that publications were missed because they did not explicitly contain the entity *post-acute COVID-19 syndrome*. Instead, publications may describe Covid 19 infections and observations made six months later. Here entity linking did not detect the explicit entity.

Vaccinations. We formulated a query to list commonly used vaccines that are associated with Covid 19: (*Covid 19*, *associated*, *?X(Vaccine)*. Helpful substitutions were for example: *BTN162 aka Pfizer (175)*, *ChAdOx1 nCoV-19 aka Astra Zeneca (79)*, and *2019-nCoV Vaccine mRNA-1273 aka Moderna (76)*. In addition, miss leading substitutions like *Vaccine (3472)* and *Covid-19 Vaccines (685)* were found and not helpful because they were far too general. We enhanced the query by asking for common side effects of *ChAdOx1 nCoV-19*: (*ChAdOx1 nCoV-19*, *associated*, *?X(Disease)*. Substitutions such as

[4] https://preview.zbmed.de/

[5] http://www.pubpharm.de/services/prototypes/narratives/

[6] https://meshb.nlm.nih.gov/record/ui?ui=C000711409

Thrombosis (93), *Thrombocytopenia (79)*, and *CVST (18)* were found. The system yielded also not helpful results like *Covid-19 (79)* and *Infections (27)* caused by wrong extractions. Again, we added the *Human* entity to precisely query for studies: (*Human*, *associated*, *?X(Disease)*) AND (*Chadox1 Ncov-19*, *associated*, *Human*). Here we could quickly find a case study [25] for CVST investigation.

Treatments. We were also interested in queries that consider treatments for Covid-19 symptoms. Therefore, we formulated the query: (*?X(Drug)*, *treats*, *Covid 19*). Helpful substitutions were *Hydroxychloroquiene (829)* and *Remdesivir (581)*. The system's provenance information (matched sentences) showed that the system found the statement in sentences like: *"An example of which is remdesivir which has now been approved for use in COVID-19 patients by the US Food and Drug Administration."* [4] We rewrote the query by integrating the patient again, similar to the previous approaches. Here we retrieved matches such as *"We identified 55 patients who were treated with remdesivir for COVID-19 and analyzed inflammatory markers and clinical outcomes."* [22]

Discussion. The case study showed that narrative information access indeed could support typical tasks like generating structured overviews of the latest literature or quickly finding precise hits: On the one hand, suitable substitutions for *Long Covid 19 symptoms* or *Covid 19 drug treatments* were indeed found, thus successfully structuring the latest literature. On the other hand, the expressive query format enabled the integration of *patients* in the query to ensure that the results had to connect the disease or drug to a concrete target group.

As a small caveat, note that all queries were matched only against implicit document contexts, ensuring the statements' context compatibility. In this way retaining the context for query processing came cheap: The origin of the statements needed to be stored and the query processing had to be restricted to document graphs. Of course, this (overly careful) restriction to document graphs also comes with severe limitations since combining knowledge from different sources is a common practice and vital necessity in scientific research. While the precision in our query tasks was very high and thus matches were accurate, the respective recall was admittedly marginal. More open yet effective measures for controlling context-compatibility than using documents graphs will be needed to build large-scale practical narrative retrieval systems (as previously discussed in section 2.2).

3.2 Narrative Query Graphs in Political Sciences

In cooperation with the specialized information service for political sciences [21](Pollux)[7] we were interested how the political sciences can benefit from narrative information access. We asked an expert (Ph.D. in political sciences) to study the biomedical narrative query graph retrieval system. He then formulated questions that would be of interest in political sciences. Due to the lack of available knowledge bases we could not realize a practical retrieval system here. Instead, we went through two of his questions and argue in the following how they could be answered and why narrative information access is vital. In addition we report on opportunities and potential obstacles in political sciences. In the following we picked two of his questions as showcases:

(1) *How do heads of government in Latin America and Scandinavia present the question what action is needed in relation to climate change?*
(2) *How do Germany's major daily newspapers negotiate the course of the refugee crisis in 2015 and 2016?*

So why do we need narrative information access to answer his questions? The main reason here is that both questions asked to combine several information: For the first question, we have to combine statements about climate change in the time period of corresponding presidents (temporal and location context). The temporal and location contexts and the source of information (the heads of government) are vital to determine statements' validity. For the second question, we have to generate a structured overview of statements and viewpoints (e.g., conservative, progressive, etc.) from daily newspapers about the refugee crisis in 2015 and 2016 (temporal context, framing, and wording). The selection of keywords (wording) may express different viewpoints. Again, context (e.g., the kind and target group of a newspaper) was vital to align the statements with a certain viewpoint.

Parts of both queries could be answered with today's knowledge bases already. Consider, for example, the usage of Wikidata: Concerning question (1), formulating a SPARQL query allowed us to retrieve a list of heads of governments in both geographical regions. And we could also combine the results with their temporal context:

- (*?country*, *head_of_state*, *?stmt*) AND
  (*?stmt*, *head_of_state*, *?person*) AND
  (*?stmt*, *start_time*, *?time*) AND
  (*?country*, *part_of*, *Latin America*).

Note, the *?stmt* notation is necessary to query Wikidata for qualifiers. This query yielded 66 results.

Concerning question (2), major newspaper could be easily identified by querying Wikidata: (*?newspaper*, *instance_of*, *daily newspaper*) AND (*?newspaper*, *country*, *Germany*). Querying Wikidata resulted in 58 newspapers. Newspapers are often associated with a *political ideology*. And indeed, Wikidata stores information that the *Frankfurter Allgemeine Zeitung (FAZ)* has the *political ideology liberal conservatism*[8]. In this way we could derive additional context information when analyzing statements from a newspaper. Note that this might be a good approximation but newspapers might also include articles that follow different ideologies.

The next part would include context-sensitive information retrieval based on the Wikidata results. To answer both questions, we had to rely on texts, e.g., from Pollux or specialized knowledge bases for claims such as ClaimsKG [24]. Here a comprehensive extraction is necessary to identify statements in texts.

But even if a knowledge base had been available, question (2) asked for different levels of granularity regarding the context of statements. In a simple scenario, it might be enough to extract statements from news articles and cluster them by their *political ideology* from Wikidata if available. However guest commentary or changes in the editorial board might include statements that stemmed from a different ideology. Therefore, we have to classify

[7] https://www.pollux-fid.de

[8] https://www.wikidata.org/wiki/Q10184

the ideology based on an article's wording and framing, and may not solely rely on the general ideology of a newspaper.

Challenges. Political sciences have a broad range of essential concepts, e.g., viewpoints, schools of thought, and ambiguous terms. These concepts are hard to identify in a text, unlike biomedical entities. Here wording and framing of texts might determine the viewpoint, whereas a drug in medicine remains the same drug regardless of wording. Moreover, central terms like "Democracy" or "Society" are not unambiguously defined and can be interpreted differently, depending on a school of thought. Furthermore, even if we identify the concepts, extracting structured information remains challenging. Statements in this domain are more complex than just expressing a binary relation between a patient and a disease condition.

These issues have to be addressed to realize a convenient narrative information access. Although solving them remains challenging, the previous cases showed that political sciences could benefit from such access. Structuring publications into schools of thought or clustering viewpoints regarding a topic would be beneficial here. Moreover, without considering the context of information, such access could hardly be realized.

3.3 Investigating Common Knowledge Bases

After we performed both case studies, we also were interested in the generalizability of the benefits of narrative information access to other domains. We first had a look at publicly available knowledge bases for their application and possible issues.

Interestingly, the following statement is included in Wikidata[9]:

- (*Barack Obama*, *born in*, *Kenya*)

In Wikidata this statement is complemented by a qualifier that states *mentioned in a conspiracy theory*. A qualifier is a statement about some other statement, i.e., a property-value pair attached to a statement. But this incorrect statement that *Barack Obama was born in Kenya* can only be sensible when considering it in the context of some *conspiracy theory*. Wikidata marks this data in their user interface by an colour encoding: *green* for fact-checked and *red* for not fact-checked. However the decision whether something is fact-checked or not is often not easy, e.g., partially fact-checked statements. In addition, different school of thoughts may accept or reject a certain statement. And having a general decision here, whether something is *true* or not, remains open.

We found another interesting example in the real-world knowledge base DBpedia[10].

- (*Barack Obama*, *was*, *Senator of Illinois*)
- (*Barack Obama*, *predecessor*, *Peter G. Fitzgerald*)
- (*Barack Obama*, *was*, *U.S. President*)
- (*Barack Obama*, *predecessor*, *George W. Bush*)

Suppose a user asks the following query: *Who was the predecessor of the U.S. President Barack Obama?* In that case the results are *George W. Bush* (correct) and *Peter G. Fitzgerald* (wrong). Thus querying DBpedia with such queries can lead to wrong results. The example query could have been answered correctly if the connection between the statements had been retained.

[9] https://www.wikidata.org/wiki/Q76 01.2022
[10] https://dbpedia.org/page/Barack_Obama 01.2022

Both examples show that the loss of context is also an issue in common knowledge bases. Information can quickly be broken down lossy and cannot be reassembled lossless afterward.

4 DISCUSSION

Narrative information access ensures that the binding process must consider contexts when making a narrative plausible. Here bindings must be context-compatible which ensures that the bindings form valid answers. We do not claim that knowledge bases cannot do the job. But if they are built without considering context and statements are restricted to triples, then information is broken down in a lossy fashion and cannot be reassembled lossless afterward. Thus contexts definitely have to be considered when designing knowledge bases to supply narrative information access.

4.1 Generalizability

Although we made our central use case in the biomedical domain, we argue that we can generalize our findings across domains. The Obama examples show how easily context can be lost in common knowledge bases. In addition, we reported on opportunities and challenges in political sciences. Here proposed use cases showed how beneficial narrative information access could be. Due to the lack of structured knowledge bases, we could hardly realize an access here. But context like temporal periods or a newspaper's viewpoint is essential to answer narrative queries correctly.

4.2 Benefits for Digital Libraries

The Covid 19 pandemics has shown how important it is to carefully handle scientific claims. Tearing such claims apart from the original lines of arguments has caused many miss leading debates (based on fake news) and movements across the world. Digital libraries should head for a more comprehensive knowledge curation by allowing narrative information access. Here the vital contexts are considered when answering queries. Our case study has shown how context-aware query systems can be applied to Covid 19 related questions. Although our study lacked a comprehensive evaluation, we demonstrate such benefits in practice: Narrative Information access allows to structure the latest literature or quickly find suitable information. Realizing and implementing suitable workflows may be cost-intensive, but digital libraries can benefit from them.

4.3 Future Work

A new challenge that has to be addressed for narrative information access is the growing heterogeneity of data sources with digital libraries, such as textual sources, image collections, experimental data or structured knowledge bases. Research data sets are a good consideration to link narrative queries against [20]. Making these heterogeneous repositories accessible in a unified way and integrating their different kinds of information requires effective access paths that often have to be intelligently customized to the content types. For narrative information access this means that bindings on (sub-)graphs of narrative queries have to be computed against extractions (either precomputed or extracted on-the-fly) from different media. Investigating such extraction is thus essential for broader applicability of narrative retrieval systems.

5 CONCLUSION

Although knowledge bases allow effective access paths in digital libraries, we demonstrated their limitations when handling narrative information. Here information, originally stated in coherent lines of arguments, can be broken into pieces that cannot be reassembled lossless afterward. This paper defines narrative information access as an extension to common knowledge base querying. Here the context of statements must be retained and considered to produce valid answers when querying narrative information. Realizing narrative information access in digital libraries can be cost-intensive in practice, but like the case study for Covid 19 retrieval has shown, implicit document contexts may approximate it. The examples of Barack Obama in common knowledge bases, our investigation in Covid 19 related questions, and the discussion in political sciences have shown how beneficial narrative information access can be. Even now existing methods and techniques can be used to implement narrative information access in digital libraries reliably. However handling heterogeneous library content (research data, tables, images, etc.) would be the next step to enhance such access further.

ACKNOWLEDGMENT

Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): PubPharm – the Specialized Information Service for Pharmacy (Gepris 267140244).

REFERENCES

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer, Busan, Korea, 722–735.

[2] Alice Bérezné, David Bougon, Florence Blanc-Jouvan, Nicolas Gendron, Cecile Janssen, Michel Muller, Sébastien Bertil, Florence Desvard, Isabelle Presot, Benjamin Terrier, et al. 2021. Deterioration of vaccine-induced immune thrombotic thrombocytopenia treated by heparin and platelet transfusion: Insight from functional cytometry and serotonin release assay. *Research and Practice in Thrombosis and Haemostasis* 5, 6 (2021), e12572.

[3] P. Brito-Zerón, N. Acar-Denizli, V. C. Romão, B. Armagan, R. Seror, F. Carubbi, S. Melchor, R. Priori, V. Valim, S. Retamozo, S. G. Pasoto, V. F. M. Trevisani, B. Hofauer, A. Szántó, N. Inanc, G. Hernández-Molina, A. Sebastian, E. Bartoloni, V. Devauchelle-Pensec, M. Akasbi, F. Giardina, M. Bandeira, A. Sisó-Almirall, and M. Ramos-Casals. 2021. Post-COVID-19 syndrome in patients with primary Sjögren's syndrome after acute SARS-CoV-2 infection. *Clin Exp Rheumatol* 39 Suppl 133, 6 (2021), 57–65.

[4] A. B. Butnariu, A. Look, M. Grillo, T. A. Tabish, M. J. McGarvey, and M. Z. I. Pranjol. 2022. SARS-CoV-2-host cell surface interactions and potential antiviral therapies. *Interface Focus* 12, 1 (Feb 2022), 20200081.

[5] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences* 509 (2020), 257–289. https://doi.org/10.1016/j.ins.2019.09.013

[6] Patrick Ernst, Amy Siu, and Gerhard Weikum. 2018. HighLife: Higher-Arity Fact Harvesting. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1013–1022. https://doi.org/10.1145/3178876.3186000

[7] James B Freeman. 2011. *Argument Structure:: Representation and Theory*. Vol. 18. Springer Science & Business Media, Berlin/Heidelberg, Germany.

[8] Halil Kilicoglu, Dongwook Shin, Marcelo Fiszman, Graciela Rosemblat, and Thomas C. Rindflesch. 2012. SemMedDB: a PubMed-scale repository of biomedical semantic predications. *Bioinformatics* 28, 23 (10 2012), 3158–3160. https://doi.org/10.1093/bioinformatics/bts591

[9] O. H. Klungel and A. Pottegård. 2021. Strengthening international surveillance of vaccine safety. *BMJ* 374 (08 2021), n1994.

[10] Hermann Kroll, Jan-Christoph Kalo, Denis Nagel, Stephan Mennicke, and Wolf-Tilo Balke. 2020. Context-Compatible Information Fusion for Scientific Knowledge Graphs. In *Digital Libraries for Open Knowledge*. Springer, Lyon, France, 33–47. https://doi.org/10.1007/978-3-030-54956-5_3

[11] Hermann Kroll, Denis Nagel, and Wolf-Tilo Balke. 2020. Modeling Narrative Structures in Logical Overlays on Top of Knowledge Repositories. In *Conceptual Modeling*. Springer, Vienna, Austria, 250–260. https://doi.org/10.1007/978-3-030-62522-1_18

[12] Hermann Kroll, Denis Nagel, Morris Kunz, and Wolf-Tilo Balke. 2021. Demonstrating Narrative Bindings: Linking Discourses to Knowledge Repositories. In *Fourth Workshop on Narrative Extraction From Texts, Text2Story@ECIR2021 (CEUR Workshop Proceedings, Vol. 2860)*. CEUR-WS.org, 57–63. http://ceur-ws.org/Vol-2860/paper7.pdf

[13] Hermann Kroll, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. 2021. Narrative Query Graphs for Entity-Interaction-Aware Document Retrieval. In *Towards Open and Trustworthy Digital Societies - 23rd International Conference on Asia-Pacific Digital Libraries, ICADL 2021, Virtual Event, December 1-3, 2021, Proceedings (Lecture Notes in Computer Science, Vol. 13133)*. Springer, Online, 80–95. https://doi.org/10.1007/978-3-030-91669-5_7

[14] Lisa Langnickel, Roman Baum, Johannes Darms, Sumit Madan, and Juliane Fluck. 2021. COVID-19 preVIEW: Semantic Search to Explore COVID-19 Research Preprints. In *Public Health and Informatics*. IOS Press, Amsterdam, the Netherlands, 78–82. https://doi.org/10.3233/SHTI210124

[15] Lisa Langnickel, Johannes Darms, Roman Baum, and Juliane Fluck. 2021. preVIEW: from a fast prototype towards a sustainable semantic search system for central access to COVID-19 preprints. *Journal of EAHIL* 17, 3 (Sep. 2021), 8–14. https://doi.org/10.32384/jeahil17484

[16] János László. 2008. *The science of stories: An introduction to narrative psychology*. Routledge, Oxfordshire, England, UK.

[17] T. Lebo, S. Sahoo, and D. McGuinness. 2013. PROV-O: The PROV Ontology. https://www.w3.org/TR/prov-o/.

[18] Frank Manola, Eric Miller, Brian McBride, et al. 2004. RDF primer. *W3C recommendation* 10, 1-107 (2004), 6.

[19] John McCarthy. 1993. Notes on Formalizing Context. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence. Chambéry, France, August 28 - September 3, 1993*. Morgan Kaufmann, Chambéry, France, 555–562. http://www-formal.stanford.edu/jmc/context3/context3.html

[20] Denis Nagel, Till Affeldt, and Wolf-Tilo Balke. 2021. Data Narrations - Using flexible Data Bindings to support the Reproducibility of Claims in Digital Library Objects. In *Proceedings of the Workshop on Digital Infrastructures for Scholarly Content Objects (DISCO 2021) co-located with ACM/IEEE Joint Conference on Digital Libraries 2021(JCDL 2021), Online (Due to the Global Pandemic), September 30, 2021 (CEUR Workshop Proceedings, Vol. 2976)*. CEUR-WS.org, Online, 19–23. http://ceur-ws.org/Vol-2976/short-2.pdf

[21] Tim Schardelmann and Wolfgang Otto. 2018. POLLUX – von der Bedarfsanalyse zur technischen Umsetzung. *Bibliotheksdienst* 52, 3-4 (2018), 225–234. https://doi.org/10.1515/bd-2018-0029

[22] K. Stoeckle, B. Witting, S. Kapadia, A. An, and K. Marks. 2022. Elevated inflammatory markers are associated with poor outcomes in COVID-19 patients treated with remdesivir. *J Med Virol* 94, 1 (01 2022), 384–387.

[23] Heiko Stoermer, Ignazio Palmisano, Domenico Redavid, Luigi Iannone, Paolo Bouquet, and Giovanni Semeraro. 2006. Contextualization of a RDF Knowledge Base in the VIKEF Project. In *Digital Libraries: Achievements, Challenges and Opportunities*. Springer Berlin Heidelberg, Berlin, Heidelberg, 101–110.

[24] Andon Tchechmedjiev, Pavlos Fafalios, Katarina Boland, Malo Gasquet, Matthäus Zloch, Benjamin Zapilko, Stefan Dietze, and Konstantin Todorov. 2019. ClaimsKG: A Knowledge Graph of Fact-Checked Claims. In *The Semantic Web – ISWC 2019*. Springer International Publishing, Cham, 309–324.

[25] C. Thompson, H. Karunadasa, D. Varma, M. Schoenwaelder, and W. Clements. 2021. Impact of COVID vaccination rollout on the use of computed tomography venography for the assessment of cerebral venous sinus thrombosis. *J Med Imaging Radiat Oncol* 65, 7 (Dec 2021), 883–887.

[26] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[27] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, and et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3, 1 (15 Mar 2016), 160018. https://doi.org/10.1038/sdata.2016.18

[28] M. Wylot, P. Cudré-Mauroux, M. Hauswirth, and P. Groth. 2017. Storing, Tracking, and Querying Provenance in Linked Data. *IEEE Transactions on Knowledge and Data Engineering* 29, 8 (2017), 1751–1764.

B.5. JCDL 2022b: A Library Perspective on Nearly-Unsupervised Information Extraction Workflows in Digital Libraries

JCDL'22b

Hermann Kroll, Jan Pirklbauer, Florian Plötzky, and Wolf-Tilo Balke. "A Library Perspective on Nearly-Unsupervised Information Extraction Workflows in Digital Libraries". ACM/IEEE Joint Conference on Digital Libraries (JCDL), Cologne, Germany, 2022, ACM. DOI: https://doi.org/10.1145/3529372.3530924

A Library Perspective on Nearly-Unsupervised Information Extraction Workflows in Digital Libraries

Hermann Kroll
kroll@ifis.cs.tu-bs.de
Institute for Information Systems, TU Braunschweig
Braunschweig, Lower Saxony, Germany

Jan Pirklbauer
j.pirklbauer@tu-bs.de
Institute for Information Systems, TU Braunschweig
Braunschweig, Lower Saxony, Germany

Florian Plötzky
ploetzky@ifis.cs.tu-bs.de
Institute for Information Systems, TU Braunschweig
Braunschweig, Lower Saxony, Germany

Wolf-Tilo Balke
balke@ifis.cs.tu-bs.de
Institute for Information Systems, TU Braunschweig
Braunschweig, Lower Saxony, Germany

ABSTRACT

Information extraction can support novel and effective access paths for digital libraries. Nevertheless, designing reliable extraction workflows can be cost-intensive in practice. On the one hand, suitable extraction methods rely on domain-specific training data. On the other hand, unsupervised and open extraction methods usually produce not-canonicalized extraction results. This paper tackles the question how digital libraries can handle such extractions and if their quality is sufficient in practice. We focus on unsupervised extraction workflows by analyzing them in case studies in the domains of encyclopedias (Wikipedia), pharmacy and political sciences. We report on opportunities and limitations. Finally we discuss best practices for unsupervised extraction workflows.

CCS CONCEPTS

• **Information systems** → **Information extraction**; *Data extraction and integration*; *Document representation.*

KEYWORDS

Open Information Extraction, Workflows, Digital Libraries

ACM Reference Format:
Hermann Kroll, Jan Pirklbauer, Florian Plötzky, and Wolf-Tilo Balke. 2022. A Library Perspective on Nearly-Unsupervised Information Extraction Workflows in Digital Libraries. In *The ACM/IEEE Joint Conference on Digital Libraries in 2022 (JCDL '22), June 20–24, 2022, Cologne, Germany*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3529372.3530924

1 INTRODUCTION

Extracting structured information from textual digital library collections enables novel access paths, e.g., answering complex queries over knowledge bases [2, 24], providing structured overviews about the latest literature [7], or discovering new knowledge [6]. However, utilizing information extraction (IE) tools in digital libraries is usually quite cost-intensive which hampers the implementation in practice. On the one hand, extraction methods usually rely on supervision, i.e., ten thousands of examples must be given for training suitable extraction models [28]. On the other hand, utilizing the latest natural language processing (NLP) tools in productive pipelines requires high expertise and computational resources.

In addition to supervised IE, Open IE methods (OpenIE) have been developed to work out-of-the box without additional domain-specific training [9, 17]. But why aren't they used broadly in digital library applications? The reason is that OpenIE generates non-canonicalized (not normalized) results, i.e., several extractions describing the same piece of information may be structured in completely different ways (synonymous relations, paraphrased information, etc.). But such non-canonicalized results are generally not helpful in practice, because a clear relation and entity semantics like in supervised extraction workflows is vital for information management and query processing. Since the lack of clear semantics has been recognized as a major issue, cleaning and canonicalization methods have been investigated to better handle such extractions [25]. Still are they ready for application in digital libraries?

In this paper case studies are used to find out how suitable nearly-unsupervised methods are to design reliable extraction workflows. In particular we analyze extraction and cleaning methods from the perspective of a digital library by assessing the required expertise, domain knowledge, computational costs and result quality.

Therefore we selected our toolbox for a nearly-unsupervised extraction from text published in last year's JCDL [12]. The toolbox contains interfaces to the latest named entity recognition (NER) and open information extraction methods. In addition, it includes cleaning and canonicalization methods to handle noisy extractions by utilizing domain-specific information. Our corresponding paper [12] advertises the toolbox to considerably decrease the need for supervision and to be transferable across domains, nevertheless it comes with several limitations:

(1) Although we did report on the extraction quality (good precision, low recall), we did not report on the costs of applying the toolbox, i.e., how much expertise and computational costs are required for a reliable workflow.
(2) We applied the toolbox only in the biomedical domain, which lessens the generalizability of our findings.
(3) Moreover, we did not report what is technically and conceptually missing in such extraction workflows.

JCDL '22, June 20–24, 2022, Cologne, Germany

ACM ISBN 978-1-4503-9345-4/22/06…$15.00
https://doi.org/10.1145/3529372.3530924

In this paper we address the previous issues by analyzing the toolbox application in three distinct real-world settings from a library perspective: 1. We extracted knowledge about scientists from the online encyclopedia Wikipedia (controlled vocabularies, descriptive writing). 2. We applied the toolbox to the pharmaceutical domain (controlled vocabularies, entity-centric knowledge) in cooperation with the specialized information service for pharmacy (www.pubpharm.de). 3. We applied the toolbox in political sciences (open vocabulary, topic/event-centric knowledge) in cooperation with the specialized information service for political sciences [23] (www.pollux-fid.de). For Pharmacy and Political Sciences, we recruited associated domain experts for expertise in the evaluation. We performed these three case studies to answer the following questions:

(1) *How much expertise and effort is required to apply nearly-unsupervised extractions across different domains?*
(2) *How generalizable are these state-of-the-art extraction methods and particularly, how useful are the extraction results?*
(3) *What is missing towards a comprehensive information extraction from texts, e.g., for retaining the original information?*

2 STUDY OBJECTIVES

In the following we briefly summarize the nearly-unsupervised extraction toolbox, raise research questions for our case studies, and explain why we selected the three domains here. Our main objective is to analyze unsupervised extraction workflows from a digital library perspective.

2.1 Overview of the Toolbox

The extraction toolbox covers three common IE areas: entity detection, information extraction and canonicalization. We shared our toolbox as open-source software and made it publicly available[1]. We focus on this toolbox because it proposed an eased and nearly-unsupervised extraction workflow by integrating latest unsupervised extraction plus suitable cleaning methods.

Entity Detection. The toolbox integrates interfaces to one of the latest NER tools, Stanford Stanza [21]. Stanza is capable of detecting 18 general purpose entity types like *persons*, *organizations*, *countries*, and *dates* in texts; See [21] for a complete overview. In addition, the toolbox supports the linking of custom entity vocabularies via a dictionary-based lookup method. The entity linker supports an abbreviation resolution and handling of short homonymous terms (link if the entity is mentioned with a longer mention in the text).

Information Extraction. The toolbox integrates implements interfaces to OpenIE methods, Stanford CoreNLP [17] and OpenIE6 [9]. Besides, the toolbox includes a self-developed path-based extraction method named PathIE. PathIE extracts statements between entities in a sentence if connected in the grammatical structure via verb phrases or custom keywords (e.g., treatment, inhibition, award, and member of) that can be specified beforehand. The OpenIE methods work entirely without entity information, whereas the PathIE requires entity annotations as starting points.

Cleaning and Canonicalization. OpenIE and PathIE may produce non-helpful and non-canonicalized outputs, i.e., synonymous noun and verb phrases that describe the same information. The toolbox supports canonicalizing and filtering such outputs automatically. First, extracted noun phrases can be filtered by entity annotations, i.e., only noun phrases that include relevant entities are kept. Here three different filters are supported to filter noun phrases: exact (noun phrase matches an entity), partial (noun phrase partially includes an entity), and no filter (keep original noun phrase).

Second, an iterative cleaning algorithm is integrated that can canonicalize synonymous verb phrases to precise relations, e.g., *birthplace* or *place of birth* to *born in*. Therefore, users can export statistics about the non-canonicalized verb phrases and build a so-called relation vocabulary. Each entry of this vocabulary is a relation consisting of a name and a set of synonymous. The toolbox utilizes this vocabulary to automatically map synonymous verb phrases to precise relations. Word embeddings are supported in the canonicalization procedure to bypass an exhausting editing of the relation vocabulary. The central idea of word embeddings is that words with a similar context appear close in the vector space [19]. The word embedding is then used to automatically map a new verb phrase to the closest match (most similar) in the vocabulary. Relation type constraints can then be used to filter the extractions further, i.e., a relation type constraint describes which entity types are allowed as subjects and objects. For example, *born in* can be defined as a relation between *persons* and *countries*. Other extractions that hurt these constraints are then removed. We did already report on some challenges of OpenIE extractions, especially on handling noun phrases [10]. In contrast to our previous works, this work analysis the complete workflow in three domains from a library perspective.

2.2 Study Goals

The study goals concern three concrete areas of study: 1. application costs, 2. generalizability, and 3. limitations for a comprehensive IE. However answering these questions on a purely quantitative level is challenging, e.g., how can the costs be measured? That is why we report our findings as a mixture of quantitative measures (e.g., time spent and runtimes) and qualitative observations (what works well and what not). We define evaluation criteria for all of the three aspects in the following.

Application Costs. We understand everything necessary to implement a workflow with the toolbox as *application costs*. We estimate the application costs in terms of

Data Preparation: transforming data into toolbox formats (e.g., JSON), working with toolbox outputs (TSV/JSON)
Implementation: computational costs (runtime and space), scalability, executed steps, effort to choose parameters, encountered issues
Domain Knowledge: entity and relation vocabulary design, required knowledge for canonicalization

Generalizability. In short, how well are the proposed methods generalizable across domains and how useful are the results?

Extraction quality: benchmarks (precision and recall), observations, extraction limitations
Usefulness: relevance of statements (e.g., non-obvious statements), domain insights, helpfulness for domain experts, usefulness in applications

[1] https://github.com/HermannKroll/KGExtractionToolbox

Table 1: The number of documents and sentences is reported for each collection and sample.

Collection	Size	Sample	
		#Documents	*#Sentences*
English Wikipedia	6.3M	2,373	74.5k
PubMed	33M	10k	87.1k
Political Sciences	1.7M	10k	66.9k

Information, originally connected in coherent written texts, might be broken into not helpful pieces in the end. For a good example, consider a drug-disease treatment: Here context information like the dose or treatment duration, which could give more information about the statement's validity [11], might get lost. We refer to such information as the **context** of statements, e.g., the surrounding scope in which a statement is valid. In addition, the connection between statements might get lost too, e.g., an assumption might lead to a conclusion. We call this the **coherence of statements**. They are crucial for real-world applications, but are they yet considered?

On Context and Coherence. Contexts affect the validity of statements and coherence describes how statements belong together. We evaluate the following criteria:

Contexts: relevance of contexts, which kind of information requires context, how does the context affect the validity of extracted statements, what must be done to retain context

Coherence: complex information that is broken into pieces, which kind of information is broken down, what are the subsequent problems with such a decomposition

2.3 Case Study Selection

We applied the toolbox in three different domains to generalize the findings in this paper. Here we focused on natural language texts written in the English language. We describe the domains and their characteristics in the following. Statistics about the used data sets and samples are listed in Table 1.

Wikipedia. A prime example of an encyclopedia is the free and collaborative Wikipedia. Encyclopedic texts should be written in a descriptive and objective language, i.e., wording and framing should not play any role. Wikipedia captures knowledge about certain items (persons, locations, events, etc.), in our understanding, entities. Here controlled ontologies about entities and relations are available; See Wikidata [26] as a good example. However Wikipedia texts also tend to include very long and complex sentences. For this case study we focus on knowledge about famous fictional and non-fictional scientists (about 2.4k scientists with an English Wikipedia article and Wikidata entry). This case study was selected because sentences are written objectively and controlled vocabularies are available for usage.

Pharmaceutical Domain. The pharmaceutical domain focuses on entity-centric knowledge, i.e., statements about entities such as drugs, diseases, treatments, and side effects. Many vocabularies and ontologies are curated to describe relevant biomedical entities, e.g., the National Library of Medicine (NLM) maintains the so-called Medical Subject Headings (MeSH)[2]. These headings are entities with descriptions, ontological relations (subclasses), and suitable synonyms. In this paper we select a subset of the most comprehensive biomedical collection, the NLM Medline collection[3]. Medline includes around 33 million publications with metadata (title, abstracts, keywords, authors, publication information, etc.). The specialized information service for pharmacy was interested in statements about drugs. That is why we selected a PubMed subset that contains drugs. Therefore, we applied the entity linking step to all Medline abstracts (Dec. 2021) and then randomly picked a subset of 10k abstracts that include at least one drug mention.

Political Sciences. The political sciences domain encompasses a diverse range of content, e.g., publications about topics and events, debates, news, and political analyses. Due to its diversity this domain does not have extensive curated vocabularies and ontologies available. We argue that entity subsets of knowledge bases like Wikidata [26] or DBpedia [2] might be good starting points to derive some entity vocabularies regarding persons, events, locations, and more. Still Wikidata and DBpedia are built as general-purpose knowledge bases and are thus not focused on political sciences (in contrast to MeSH for the biomedical domain). Nevertheless they might be helpful to analyze texts in political sciences and that is why we analyze them for a practical application here. In addition, descriptions of entities in political sciences tend to be subjective, i.e., they depend on different viewpoints and schools of thought. For example, the accession of Crimea to Russia in 2014 was a highly discussed topic whether this event could be seen as peaceful secession or as an annexation. In contrast to objective and entity-centric statements in biomedicine, political sciences are far more based on the wording and framing of certain events. This case study analyzes how far IE methods can bring structure into these texts and where these methods fail. The specialized information service for political sciences (Pollux) provided us with around three million publications (around 1.3 million English abstracts). Our case study is based on a random sample of 10k abstracts selected from the English subset. In addition, domain experts manually selected five abstracts due to their focus on the diverse topics of the EU, philosophy, international relations, and parliamentarism.

3 CASE STUDIES

For our case studies we developed scripts, produced intermediate results, and implemented some improvements for the toolbox. The details, used data and produced results of every case study can be found in our evaluation scripts on GitHub (see the Toolbox GitHub Repository). We included a Readme file to document the following case studies. All of our experiments and time measurements were performed on our server, having two Intel Xeon E5-2687W (3,1GHz, eight cores, 16 threads), 377GB of DDR3 main memory, one Nvidia 1080 TI GTX GPU, and SSDs as storage.

3.1 Wikipedia Case Study

This first case study was based on 2.3k English Wikipedia full-text articles about scientists. The conversion of Wikipedia articles

[2]https://meshb.nlm.nih.gov/search

[3]https://www.nlm.nih.gov/medline/medline_overview.html

Table 2: Extraction statistics for all three domains: Sentences (number, percentage of complex sentences, number of sentences with at least two entities mentions), Entity Detection (number of Stanza NER and dictionary-based entity linking annotations), OpenIE6 (percentage of complex subjects and objects, number of extractions computed by the different entity filters [no, partial, exact, subject]) and PathIE (number of extractions).

	Sentences			Entity Det.		OpenIE6						PathIE
	#Sent.	*Compl.*	*#w2E*	*#NER*	*#EL*	*C. Subjs.*	*C. Objs.*	*#No EF*	*#Part. EF*	*#Exact EF*	*#Subj. EF*	*#Extr.*
Wikipedia	74.5k	92.7%	50.3k	155.0k	113.2k	16.2%	74.5%	177.1k	317.8k	2.9k	80.9k	1.3M
Pharmacy	87.1k	92.2%	47.4k	-	232.5k	37.8%	72.1%	207.6k	88.0k	291	151.0k	430.8k
Pol. Sci.	66.9k	93.2%	17.6k	80.0k	3.7k	32.0%	74.3%	147.2k	28.6k	128	7.3k	-

was simple: We downloaded the available English Wikipedia dump (Dec. 2021), used the WikiExtractor [1] to retrieve plain texts, and filtered these texts by our scientist's criteria (title must be about a scientist of Wikidata). Next we developed a Python script to transform the plain texts into a JSON format for the toolbox. The data transformations took half a person-day.

Entity Linking. In this case study we focused on statements about scientists such as works, scientific organizations, and degrees. Therefore, we performed entity linking to identify these concepts and use them to filter the extraction outputs. We derived corresponding entity vocabularies from Wikidata by utilizing the official SPARQL endpoint. We retrieved vocabularies by asking for English labels and alternative labels for the following entity types: *Academia of Sciences*, *Awards*, *Countries*, *Doctoral Degrees*, *Religions* and *Irreligions*, *Scientists*, *Professional Societies*, *Scientific Societies* and *Universities*. We adjusted the SPARQL queries to directly download the vocabularies as TSV files in the toolbox format.

A first look over this entity vocabulary revealed some misleading labels (e.g., the, he, she, and, or), which we removed. We applied the dictionary-based entity linker utilizing our vocabulary on the articles. The linker yielded many erroneously linked entities due to very ambiguous labels in the dictionary, e.g., the mentions *doctor*, *atom* and *observation* were linked to fictional characters which are scientists regarding the Wikidata ontology. Next synonyms like *Einstein* were erroneously linked when talking about his family or talking about the term *Einstein* in the sense of *genius*. The linker also ignored pronouns completely, i.e., no coreference resolution was applied. Especially in Wikipedia articles, pronouns are often used. In addition, we executed Stanford Stanza to recognize general-purpose entity types like dates or organizations. We found short entity names to be too ambiguous. That is why we removed all detected entities with less than five characters. This step yielded 155k Stanza NER mentions and 113.2k dictionary-based entity links.

Information Extraction. We applied the OpenIE6 method and the entity filter methods (no filter, partial, exact). We obtained 117.1k (no filter), 317.8k (partial) and 2.9k (exact) extractions. Note that statements can be duplicated for the partial filter if multiple entities are included within the same noun phrase. We exported 100 results for each filter randomly and analyzed them. In the following we report on some examples of good and bad extractions.

Some interesting results about Albert Einstein are listed in Table 3. OpenIE6 produced correct and helpful extractions when sentences were short and simple (no nested structure, no relative clauses, etc.). When sentences became longer, the tool yielded short subjects but long and complex objects, e.g., a whole subordinate clause like *that science was often inclined to do more harm than good.* See E3.1 in Table 3.

We developed a short script to quantify them to better understand how many sentences, subjects, and objects were complex. Therefore, we formulated regular expressions to check if a sentence contained multiple clauses split by punctuation (,|;|:), or words (and|or|that|thus| hence|because|due|etc.). We counted sentences, subjects, and objects as complex if they matched one of these regular expressions. In addition, if a sentence was denoted as complex and the extracted noun phrase was larger than 50% (character count) of the sentence or it contained words like (by|at|for|etc.), we considered it complex. For our sample, 92.7% of the sentences, 16.2% of subjects, and 74.5% of objects were classified as complex. We iterated over these classifications to verify the filter criteria.

Partial Entity Filter. This filter yielded problematic results because much information was lost, e.g., a whole subordinate clause was broken down to a single entity regardless of where the entity appeared in this clause. In some cases, this filtering completely altered the sentence's original information; See E2.2 for a good example. Here the extraction *Einstein was elected the Royal Society* was nonsense because *Foreign Member* was filtered out. In E2.1, the extracted statement missed that the *philosopher* was *Eric Gutkind*, and thus lost relevant information.

Exact Entity Filter. The exact filter was very restrictive because the number of extractions was reduced from 117.9k to 2.9k. However the extraction seemed to have good quality. In E1.1, the extraction *Einstein was visiting the US* was correct, but the context about the year *1933* was lost. Extraction E1.2 showed that OpenIE6 was capable of extracting implicit statements like *be Professor of*. Again, the surrounding context about the year and Einstein was lost. Other extractions showed that a coreference resolution would be beneficial to resolve mentions like *his*, *in the same article*, and, *these models*.

We observed many complex object phrases (74.5% in sum). These complex phrases contained more information than a single entity. Filtering them led to many wrongly extracted statements. In contrast, subject phrases were often simple and might stand for a single entity (only 16.2% are complex). Due to these observations, we developed a subject entity filter, i.e., only subjects had to match entities directly. The idea was to identify subjects as precise entities and keep object phrases in their original form to retain all information.

Subject Entity Filter. This filter worked as expected: In E3.1 and E3.2, the subject was identified as the Person *Einstein* whereas the original information was kept in the object phrase. This filtering

Table 3: OpenIE6 example extractions from the Wikipedia article of Albert Einstein. On the left the corresponding entity filter is shown (subject, partial and exact). Subject[S], predicate[P] and object[O] are highlighted respectively.

Wikipedia	**Exact**	E1.1	In 1933, while **Einstein**[S] **(Person)** *was visiting*[P] **the United States**[O] **(Country)**, [...]
		E1.2	On 30 April 1905, Einstein completed his thesis, with **Alfred Kleiner**[S] **(Person)**, *[be] Professor*[P] of **Experimental Physics**[O] **(ORG)**, serving as "pro-forma" advisor.
	Partial	E2.1.	In a German-language letter to **philosopher**[O] **(Profession)** Eric Gutkind, dated 3 January 1954, **Einstein**[S] **(Person)** *wrote*[P]: [...]
		E2.2	**Einstein**[S] **(Person)** *was elected*[P] a Foreign Member of **the Royal Society**[O] **(Org)** (ForMemRS) in 1921.
	Subject	E3.1	During an address to Caltech's students, **Einstein**[S] **(Person)** *noted*[P] **that science was often inclined to do more harm than good**[O].
		E3.2	**Einstein**[S] **(Person)** *started teaching*[P] himself **calculus at 12**[O], and as a 14-year-old [...]

allowed us to generate a structured overview about Albert Einstein, for example.

In addition to OpenIE6, we investigated how useful PathIE is to extract relations between the relevant entity types such as scientists and awards. PathIE allowed us to specify keywords that can indicate a relation. In a first attempt, we applied PathIE with a small relation vocabulary of Wikidata. We exported the English labels and alternative labels of eleven Wikidata properties that describe the relations between the given entity types: academic degree, award received, date of birth, date of death, field of work, member of, native language, occupation, religion, and writing language.

We exported 100 randomly selected PathIE extractions for evaluation. When several entities were detected in long and nested sentences, PathIE yielded many wrong extractions because the corresponding entities were connected via some verb phrases, e.g., *Einstein return Zurich* from *Einstein visited relatives in Germany while Maric returned to Zurich* or *Written languages write Leningrad*. Filtering these extractions by entity types like (Person, Date) or (Person, Award) revealed more helpful extractions, e.g., *Einstein win Nobel Prize* from *Einstein received news that he had won the Nobel Prize in November*.

However we encountered severe entity linking issues when analyzing the cleaned OpenIE6 and PathIE extractions. On the one hand, ambiguous terms were linked wrongly. On the other hand, fragments of a text span were linked against an entity although the whole text span referred to a single entity, e.g., only linking *Albert Einstein* in the text mention *Albert Einstein's Theory of Relativity was published in 1916*. These issues directly affected the extraction quality. We stopped the extraction part at this point.

Canonicalization. We used our small relation vocabulary to canonicalize the extractions. This procedure did work out for PathIE because it directly extracted the vocabulary entries from the texts. For example we could retrieve a list of statements that indicate an *award received* relation. However further cleaning was required to obtain *award received* relations between persons and awards. We analyzed 100 entries for this relation. Although some extraction were correct, 60 of 100 extractions had linked awards that were not helpful, e.g., *awards*, *doctor*, *medal*, *president* and *master*. The remaining 40 extractions displayed six wrongly identified persons. However the remaining 34 extractions seemed to be plausible, although some information was missed, like the *Nobel prize's* category.

In contrast, the canonicalization procedure did not work for OpenIE6 extractions. The reason was that the extracted verb phrases did not appear directly in the vocabulary. Thus we used a pretrained English Wikipedia word embedding from fasttext[4] to find similar matches in the relation vocabulary. We adjusted the cleaning parameters (how similar terms must be and how often terms must occur) and canonicalized the OpenIE6 verb phrases. However most verb phrases were mapped wrongly because the vocabulary was relatively small, e.g., *divorce* was mapped to *date of death* because it was the closest match.

We then derived a list of 120 Wikidata properties that involved persons (ignoring usernames and identifiers) to find more matches. We repeated the canonicalization and analyzed 100 extractions obtained by the subject entity filter because it retrieved the most helpful results in the previous step.

Most of the canonicalized verb phrases were mapped incorrectly, e.g., mapping *start teach* to *educated at* or *begin* to *death of place* was wrong. For a positive example, the verb phrase *publish* was mapped to the relation *notable work* and *write* to *author*, e.g., *Galileo publish (↦ notable work) Dialogue Concerning the Two Chief World Systems*. Although this relation was correct for some fewer extractions, most of these mappings were problematic, e.g., *Einstein publish (↦ notable work) his own articles describing the model among them.* Here the object phrase did not contain a notable work in the sense of how we would understand it. In summary, the canonicalization procedure had many problems for OpenIE6 extractions. The main issue was that the canonicalization procedure only considered the verb phrase and not the surrounding context in a sentence. But this surrounding context is essential to determine the relation. In addition, the relation vocabulary obtained from Wikidata might be insufficient because it did not contain verb phrases as we would expect them. Wikidata describes relations by using substantives and nouns, e.g., notable work of, notable work by, notably created by for the relation *notable work*.

Application Costs. We spent much of our time understanding the Wikidata ontology and formulating suitable SPARQL queries to retrieve the utilized vocabularies. The corresponding vocabularies could be exported directly from Wikidata and did not need transformations besides concatenation of files. We formulated several SQL queries to analyze, clean, and filter entity annotations and

[4] https://fasttext.cc/docs/en/pretrained-vectors.html

extractions in the toolbox's underlying database. In summary, three persons performed this case study within three person-days.

Generalizability. We had a close look at existing Wikipedia relation extraction benchmarks for evaluation. Unfortunately, these benchmarks are often built distantly supervised, i.e., if two entities appear in a sentence, and both entities have a relation in a knowledge base, then this relation is the class that must be predicted for this sentence. In other words, the relation does not have to appear within the sentence. Furthermore these benchmarks often require domain knowledge, e.g., if a football player started his career at a sports team, then the football player *played for* this team. This additional knowledge is typically not included in OpenIE methods. OpenIE extracts statements based on grammatical patterns in a sentence: For the previous example, the tool would extract that the football player started his career at the sports team, but not that he also played for the team. That is why we did not evaluate the extraction tool on existing benchmarks because we expected the quality to be low by design. Moreover, mapping verb phrases to precise relations would also be challenging. In contrast, we wanted to understand how useful the results were for practical applications.

First, an improved entity linking would have solved several issues in our case study. Next the handling of complex noun phrases was an issue: Although the exact entity filter was too restrictive, it resulted in suitable extractions. The partial entity filter messed up the original information and was thus not helpful. OpenIE6 and the subject entity filter allowed us to retrieve a list of actions performed by Albert Einstein, for example. However this filtering did not yield a canonicalized knowledge base by design. Our case study has shown that PathIE could extract relations between scientists and awards. Although we could not evaluate the quality in rough numbers, we spent three person-days designing a possible extraction workflow. Here the toolbox allowed us to retrieve such semi-structured information in an acceptable amount of time.

What is missing. Handling of complex noun phrases was a significant issue: On the one hand, the decisive context was lost if phrases were broken down into small entities. On the other hand, if phrases were retained in their original form, context was kept, but the canonicalization remained unclear. To the best of our knowledge there is no out-of-the-box solution that will solve these issues.

3.2 Pharmaceutical Case Study

We applied the toolbox to a subset of the biomedical Medline collection for our second case study. The PubMed Medline is available in different formats, among other things, in the PubTator format which is supported by the toolbox. We downloaded the document abstracts from the PubTator Service [27].

Entity Linking. We utilized existing entity annotations (diseases, genes, and species) from the PubTator Central service. In addition, we selected subsets of MeSH (diseases, methods, dosage forms), ChEMBL [18] (drugs and chemicals), and Wikidata (plant families) to derive suitable entity vocabularies. We developed scripts that retrieved relevant entries from these vocabularies. This step required us to export relevant entries from XML and CSV files into TSV files.

We then applied the entity linker and analyzed the results by going through the most frequent annotations. Our first attempt yielded frequently, but obviously wrongly linked words such as *horse*, *target*, *compound*, *monitor*, and *iris*. These words were derived from ChEMBL because they were trade names for drugs. We found such trade names to be very ambiguous and removed them. But we also found annotations such as *major*, *solution*, *relief*, *cares*, *aim*, and *advances*. We went through the 500 most tagged entity annotations to remove such words by building a list of ignored words. We repeated the entity linking by ignoring these words and computed 232.5k entity mentions. We did not apply Stanford Stanza NER here because we were interested in biomedical entities.

Information Extraction. The domain experts were interested in statements between entities. That is why we applied OpenIE6 and analyzed the partial and exact entity filter. OpenIE6 extracted 207.6k extractions and filtering them yielded 88k (partial) and 291 (exact) extractions. An analysis of the extractions showed that 92.2% of sentences, 37.8% of subjects, and 72.1% of objects were complex. The exact entity filter was too restrictive and not helpful because the remaining extractions were too few for a practical application.

Partial Entity Filter. A closer look at 100 randomly sampled extractions indicated that many noun phrases were complex again. The partial entity filter mixed up the original sentence information by filtering out the important information. For example consider the following sentence: *Inhibition of P53-MDM2 interaction stabilizes P53 protein and activates P53 pathway*. Here the partial entity filter extracts the statement: (*MDM2*, *stabilizes*, *protein*). This statement mixed up the original information. Our analysis showed that the vast majority of filtered extractions were incorrect. In addition, OpenIE6 is focused on verb phrases to extract statements (here *stabilizes*).

However many relevant statements are expressed by using special keywords, e.g. *treatment*, *inhibition*, *side effect*, and *metabolism*. That means that these OpenIE methods will usually not extract a statement from clauses like *metformin therapy in diabetic patients* by design. A similar observation was already made in the original toolbox paper, where OpenIE methods' recall was clearly behind supervised methods (5.8% vs. 86.2% and 6.2% vs. 75.9% on biomedical benchmarks) [12]. Supervised extraction methods would engage this problem by learning typical patterns of how a treatment can be expressed within a sentence.

To integrate such specialized keywords in the extraction process, we applied the recall-oriented PathIE method. In the previous example, the entities *metformin* and *diabetic patients* are connected via the keyword *therapy*. In this way PathIE extracted a helpful statement. However we had to build a relation vocabulary to define these specialized keywords. In cooperation with domain experts, we built such a vocabulary by incrementally extracting statements with PathIE, looking at extractions and example sentences to find out what we were missing. In sum, we had three two-hour sessions to build the final relation vocabulary. The final PathIE step yielded 430.8k extractions and took two minutes to complete. Some interesting results are listed in Table 4. We then iterated over a sample of 100 of these extractions.

PathIE was capable of extracting statements from long and nested sentences, e.g., a treatment statement in P1.1. in Table 4. However we also encountered several issues with PathIE. If a sentence contains information about treatments' side effects (also linked as diseases), PathIE extracted them wrongly as the treated condition (See P1.2). A similar problem occurred when a drug therapy was

Table 4: PubMed PathIE example extractions. On the left the canonicalized relation is annotated.

Pharmacy	**Treats**	P1.1	We tested whether short-term, low-dose *treatment*[P] with the fluvastatin and **valsartan**[S] **(drug)** combination could improve impaired arterial wall characteristics in type 1 **diabetes mellitus**[O] **(disease)** patients.
		P1.2.	We encountered two cases of cerebellar **hemorrhage**[O] **(Disease)** in patients *treated*[P] with **edoxaban**[S] **(Drug)** for PVT after hepatobiliary surgery during the past 2 years.
	Inhibits	P2.1	**Anthraquinone**[S] **(Drug)** derivative emodin inhibits tumor-associated angiogenesis through *inhibition*[P] of **extracellular signal-regulated kinase 1**[O] **(Gene)**/2 phosphorylation.
		P2.2	Impact of **aspirin**[S] **(Drug)** on the gastrointestinal-sparing effects of **cyclooxygenase-2**[O] **(Gene)** *inhibitors*[P].
	Induces	P3.1	**Hyperglycemia**[O] **(Disease)**-*induced*[P] mitochondrial dysfunction plays a key role in the pathogenesis of diabetic **cardiomyopathy**[S] **(Disease)**.
		P3.2	Conclusions H. pylori **Infection**[S] **(Disease)** appears to *cause*[P] decreases in **Vitamin B12**[O] **(Excipient)**[...].

used to treat two diseases simultaneously. Here PathIE yielded six statements (three mirrored): two therapy statements about the drug and each disease, and one therapy statement between both diseases, which is wrong. In example P2.2, PathIE failed to recognize that aspirin *effects* the inhibitors and is not an inhibitor itself.

A second problem was the direction of extracted relations: A *treats* relation could be defined as a relation between *drugs* and *diseases*. If a relation has precise and unique entity types, then an entity type filter removes all other, and possibly wrong, extractions. Suppose that a disease causes another disease (think about a disease that causes severe effects). In that case, PathIE would extract both directions: (a causes b) and (b causes a). For example PathIE would extract two statements from *myocardial damage caused by ischemia-reperfusion*. Here an entity type filter did not solve the problem because both entities have the type *disease*.

Third, in situations with several entities and clauses within one sentence, PathIE seemed to mess up the original information and extracted wrong statements, e.g., see P3.1, where hyperglycemia did not induce cardiomyopathy. In summary, PathIE could extract statements from complex sentences, but a cleaning step had to be applied afterward to achieve acceptable quality.

Canonicalization. We exported the database statistics for PathIE. We carefully read the extracted verb phrases in cooperation with two domain experts. Verb phrases such as *treats*, *prevents* and *cares* point towards a *treats* relation, which we included into our relation vocabulary. Phrases such as *inhibits* and *down regulates* may stand for a *inhibits* relation. To find more synonyms automatically, we used a Biomedical Word Embedding [31] that we used in the toolbox paper before. Following this procedure, we defined eight relations with 30 synonyms. We repeated the procedure five times and derived a relation vocabulary of 60 entries. The relation vocabulary was a mixture of verb phrases and keywords that indicated a relation in the text. In sum, we had six sessions of two hours each to build the final relation vocabulary.

However we noticed that PathIE extractions were problematic when not filtered. Relations like *treats* and *inhibits* also include entity types that we had not expected, e.g., two diseases in treats. We formulated entity type constraints for eight relations to remove such problematic statements. The relations *treats* and *inhibits* looked more helpful because they only contained relevant entity types. We tried to filter relations like *induces* between diseases. Some extractions were correct, but many extractions mixed up the relation's direction (a causes b instead of b causes a). In the end, PathIE was not very helpful for extracting such directed relations due to its poor quality. We stopped the cleaning here, but a more advanced cleaning would be helpful to handle such situations.

Application Costs. We spent most of our time designing entity and relation vocabularies and analyzing the retrieved results. The creation of suitable vocabularies took as around one week in sum. The execution of the toolbox scripts was quite simple; See our GitHub Repository. To measure the runtime for PubPharm, we applied the PathIE-based pipeline on around 12 million PubMed abstracts (PubMed subset about drugs). The procedure could be completed within one week: Entity detection took two days for the complete PubMed collection (33 million abstracts). PathIE took five days and cleaning took one day. Hence, such an extraction workflow is realizable for PubPharm with moderate costs.

Generalizability. We already know that OpenIE and PathIE have worse performance than supervised methods; See the benchmarks in the original toolbox paper. However we could design a suitable extraction workflow with an acceptable amount of time (a few weeks of cooperation with nine sessions with experts). OpenIE6 had a very poor recall, and filtering remained unclear. Thus, they were not of interest for PubPharm's purposes.

PubPharm is currently using the PathIE extractions in their narrative retrieval service [13]. Here recall is essential to find a suitable number of results to answer queries. Although the quality of PathIE is only moderate, the quality seems to be sufficient for such a retrieval service. Here the statement should hint that the searched information is expressed within the document, e.g., that a *metformin treatment* is contained. The main advantage of a retrieval service is that the original sentences can be shown to users to explain where the statements were extracted. In summary, if users are integrated into the process, and the statements' origin is shown, these PathIE allow novel applications like the retrieval service.

Nevertheless, we encounter several issues: First, PathIE extracts wrong statements if several entities are contained in a sentence. Next the undirected extractions of PathIE are often problematic if no additional cleaning can be performed (e.g., relations between diseases). Although these issues must be faced somehow, PathIE allowed us an extraction workflow that we could not have realized using supervised methods due to the lack of training data. We would not recommend PathIE for building a knowledge graph due

Table 5: Pollux OpenIE6 example extractions. On the left the corresponding entity filter is shown (subject, partial and exact).

Political Sciences	**Partial**	PS1.1	Stalin wanted all 16 **Soviet**[S] **(NORP)** Republics *to have*[P] separate seats in **UN General Assembly**[O] **(ORG)** but only 3 were given Russia Ukraine Belarus.
		PS1.2	This paper seeks to understand why the **United States**[S] **(GPE)** *treated*[P] **Japan**[O] **(GPE)** and Korea *differently*[P] in the revisions of bilateral nuclear cooperation agreements.
	Subject	PS2.1	Based on these features, the article suggests that **China**[S] **(GPE)** is poised *to become*[P] **a true global power**[O].
		PS2.2	Prior to the introduction of the Transparency Register the **European Parliament**[S] **(ORG)** *had maintained*[P] **a Register of Accredited Lobbyists since 1996**[O] while the European Commission [...].

to many wrong extractions that would lead to transitive errors when performing reasoning on the resulting graph.

What is missing? In this pharmaceutical case study we focused on relations between pharmaceutical entities. PathIE completely ignored the surrounding context of statements, e.g., dose and duration information of therapies. The coherence of statements was also broken down, e.g., drug, dosage form, disease, and target group of treatments were split into four separate statements. The desired goal would be to retain all relevant information within a single statement. However PathIE is restricted to binary relations. A future enhancement of PathIE would be desirable to retain all connected entities in a sentence. PubPharm's retrieval service bypassed the problem by using document contexts, i.e., statements from the same document belong together. The service uses abstracts, and this approximation would not have been possible for full texts because a full-text document might contain several different contexts.

3.3 Political Sciences

We applied the toolbox to 10k abstracts from political sciences.

Entity linking. The field of political sciences displays some distinct differences compared to the biomedical field and encyclopedias like Wikipedia. A notable difficulty lies in the lack of well-curated vocabularies for the domain. This can be mitigated in two ways: by using NER as implemented by Stanza [21] or by constructing/deriving entity vocabularies from general-purpose knowledge bases like Wikidata. We investigated both approaches.

Stanza NER yielded ca. eight tags per document. The extracted mentions seemed sensible, e.g., entities like *USA*, *Bush* or *the Cold War* were extracted. However Stanza NER also displayed some drawbacks, e.g., it was sensitive to missing uppercase letters for identifying names. Such restrictions can be problematic in practice due to bad metadata (abstracts in upper case).

For the second approach we selected wars (Q198), coup d'états (Q45382) and elections (Q40231) as seed events, since those are likely to be subject of debate in political science articles. Furthermore we inductively utilized Wikidata's subclass property (P279) to receive all subclasses of all seed events. We used the SPARQL endpoint to export the corresponding vocabularies by asking for the English label and alias labels for the seed events, all instances of the seed events, and their subclasses. In total, we collected 2.9k wars, 904 coups, and 79.7k election entries. An evaluation of the toolbox's entity linker showed good performance on wars while coup d'états and elections were rarely linked sensible. However we increased the linking quality by applying simple rules, e.g., the entity label must contain the term *election*. We derived 3.7k entities in sum.

Information Extraction. Due to the lack of comprehensive entity vocabularies, we focused on OpenIE6 in this case study and omitted PathIE. OpenIE6 yielded 147.2k (no filter), 28.6k (partial), 128 (exact) and 7.3k (subject) extractions. Subject phrases tended to be short (only 32.0% were complex), and object phrases tended to be long (74.3% complex) again, like in the previous case studies. 93.2% of all sentences were estimated to be complex. We randomly sampled 100 extractions of each filter for further analysis. Again, extractions from small sentences looked helpful, while long sentences led to long object phrases. We picked some interesting results and display them in Table 5.

Exact entity filter. Again the exact entity filter decreased the number of extractions drastically (from 147.2k to 128). But extractions seemed plausible, e.g., *Alexander Lukashenko is president of Belarussian[SIC]* from *Focus on the career and policies of the first Belarussian president, Alexander Lukashenko, elected in 1994.* Another correct extraction was *United States prepares to exit* from *As the United States prepares to exit Afghanistan [...].*

Partial entity filter. In PS1.1, the extraction *Soviet to have UN General Assembly* was wrong because the context about Stalin and separate seats was missed. The extraction in PS1.2, *United States treated differently Japan*, was not helpful because *Korea* was missed. Again, the context that this statement was investigated in that article was lost. We found the extractions of the partial filter not helpful: Either they mixed up the original information or decisive context was missed.

Subject entity filter. The extraction PS2.1 showed a correct extraction, but then the information that the statement was suggested by an article was missed. Although the sentence of P2.2 was quite complex, OpenIE6 extracted useful information about the *European Parliament*: *European Parliament had maintained a Register of Accredited Lobbyists since 1996.*

We skipped the canonicalization procedure here because we already knew that canonicalizing OpenIE6 verb phrases remains unclear (see Wikipedia case study). The exact filter yielded fewer extractions, partial filtering resulted in incorrect statements, and PathIE could not be applied due to the lack of vocabularies. And extractions from the subject filter could hardly be canonicalized to precise relations if the object phrase contains large sentence parts.

Application Costs. The application costs for the political domain seemed higher compared to the other two case studies. The lack of curated vocabularies necessitates the creation of such. As demonstrated, this can hardly be done automatically but requires domain knowledge. We exported some vocabularies from Wikidata but we missed many entities in the end. In sum, we had four sessions, each

1.5 hours, with a domain expert to analyze the results. The case study took us five person-days in sum.

Generalizability. Due to the lack of available benchmarks, we restricted our evaluation to a qualitative level. As another difficulty, simple fact statements, e.g., *Joe Biden is the president of the USA* hardly carried new or relevant information. Still disputed claims, viewpoints, or assessments like *the UK aims to position itself as an independent power after Brexit* might be the subject of study. This often resulted in long clauses for the subjects and objects that are hard to map to the already sparsely recognized named entities. But the subject entity filter allowed us to retain that *UK aims to position itself as an independent power after Brexit* as a suitable extraction. We plan to proceed from here by extracting semi-structured information via the subject filter.

What's Missing. Additionally the context of a statement is often highly relevant. In the example the statement loses its information if the context *after Brexit* is omitted. Observations were similar to the Wikipedia case studies: Either the object phrases retained the context but could hardly be handled by filtering methods. Or the object phrases were short and missed information.

4 DISCUSSION

In the following we discuss how suitable unsupervised extraction workflows are in digital libraries by considering technical and conceptual limitations. Furthermore we give best practices on what to do and when supervision is necessary.

4.1 Technical Toolbox Limitations

The toolbox filtered verb phrases by removing non-verbs (stop words, adverbs, etc.) and verbs like *be* and *have*. Here negations in verb phrases were lost, too. We implemented a parameter to make this behavior optional. Next we implemented the subject entity filter that was useful in Wikipedia and political sciences. Here a statement's subject must be linked to an entity, but the object can keep the original information. Then the results could be used as a semi-structured knowledge base, e.g., showing all actions of *Albert Einstein* or *positions* that the *EU* has taken.

In addition, the dictionary-based entity linker fails to resolve short and ambiguous mentions. These wrongly linked mentions cause problems in the cleaning step (entity-based filters). Here more advanced linkers would be more appropriate to improve the overall quality. A coreference resolution is also missing, i.e., resolving all pronouns and mentions that refer to known entities.

PathIE is currently restricted to binary relations but might be extended to extract more higher-ary relations, e.g., by considering all connected entities via a verb phrase or a particular keyword like treatment. A suitable cleaning would be possible if the relation arguments could be restricted to entity types.

4.2 Restrictions of Unsupervised Extraction

The first significant restriction of unsupervised methods is their focus on and thus restriction to grammatical structures. Suppose the example: *The German book Känguru-Chroniken was written by Marc-Uwe Kling*. Here unsupervised methods may not extract that the language of the work is German.

In common relation extraction benchmarks such relations do appear and can be learned and inferred by modern language models [4, 15]. However we argue that such extractions require high domain knowledge, typically unavailable in unsupervised extraction methods. Similar examples could be made in specialized domains like pharmacy (treatments, inhibitions, etc.). Moreover it is not possible to integrate this knowledge into unsupervised models by design: The model would need training data to infer such rules and, thus, be supervised. We do not expect unsupervised models with access to comprehensive domain-specific knowledge soon.

Our case studies showed that OpenIE6 extracts noun phrases in two ways: Either noun phrases are short and miss relevant information from the sentence. These phrases are easier to handle but may be unhelpful in the end. Or the noun phrases are long and complex but retain the original information. Handling complex phrases requires more advanced cleaning methods.

The toolbox canonicalization procedure for relations considers only the verb phrases, not the surrounding context. Verb phrases like *uses*, *publish*, and *prevent* could refer to a plethora of relations. In the end more advanced methods are required for a suitable canonicalization quality. Especially canonicalizing OpenIE6 verb phrases to precise relations was not really possible.

4.3 Application and Costs

Although we observed several issues and limitations, these methods can be used to implement services in digital libraries. We summarize the measured runtimes and computed estimations for the corresponding collections in Table 6.

Consider PubPharm for a good example: PathIE could enable a graph-based retrieval service with moderate costs [13]. Around nine sessions with experts and moderate development time were necessary to implement a workflow. The computation of PathIE took 2 min on our sample and was estimated to take 4.6 days for the whole PubMed collection. Indeed, PubPharm could perform the complete extraction workflow in one week.

Our current cooperation with Pollux revealed that OpenIE6 could bring more structure in this domain. We will continue our work with Pollux by focusing on research questions that we would like to answer with semi-structured information derived from OpenIE6 with subject entity filtering.

On our server with an Nvidia GTX 1080 TI, the computation of OpenIE6 took 55.4 min on the Pollux sample and is estimated to take five days for the complete collection. For Wikipedia the sample took 53.6 min, and all English articles would require 98.8 days. Note that we used a single GPU which is already five years old. Hence the workflow can be accelerated with a modern GPU and parallelized by utilizing multiple GPUs. In addition, OpenIE6 can also be restricted to sentences that contain at least two entities. Here the runtime is decreased from 55.4 to 22.4 min (Pollux) and 53.6 to 41.4 min (Wikipedia).

4.4 Best Practices

Subsequently we give some advice that we can deduce from our case studies. OpenIE6 handles short and simple sentences well. Here the exact entity filter will produce suitable extractions but decrease the recall drastically. The partial entity filter improves the recall

Table 6: The table summarizes the measured runtimes for the samples and gives an estimation for the whole collection.

		Wikipedia		Pharmacy		Political Sciences	
		Sample	*Estimation*	*Sample*	*Estimation*	*Sample*	*Estimation*
Entity Det.	*NER*	10.5 min	19.4 days	-	-	10.1 min	21.6 hours
	EL	0.6 min	1.2 days	1.2 min	2.8 days	0.7 min	1.4 hours
Extraction	*PathIE*	2.6 min	4.7 days	2.0 min	4.6 days	-	-
	OpenIE6	53.6 min	98.8 days	74.0 min	98.8 days	55.4 min	5.0 days
Cleaning		< 1 hour	<1 day	< 1 hour	<1 day	< 1 hour	< 1 day

but often messes up the original information. We recommend two strategies for long and complex sentences:

First, do not use the exact or partial entity filter because important information can be missed. Use the subject entity filter to retrieve precise entities as subjects and the original information in object phrases. This filter allows the construction of semi-structured knowledge bases, e.g., positions that were taken by the *EU* or actions that *Albert Einstein* has done. Another option is to use no filter, but then, the extractions are not cleaned in any way.

Second, PathIE can find specialized relations that are expressed by keywords. But PathIE requires directed relations that must be cleaned by entity type constraints. Detecting such relations via PathIE is fast and probably cheaper than training supervised extraction models. However PathIE will fail if several entities of the same type are mentioned within a sentence, e.g., side effects of treatments. Here supervised methods are required to achieve suitable quality.

5 RELATED WORK

The main goal of information extraction (IE) is the extraction of structured information from unstructured or semi-structured information such as texts, tables, figures, and more [9, 16, 17, 28]. In the following we give an overview of challenges and research trends in IE from texts.

Current Trends. Modern IE research mainly focuses on improving the extraction accuracy, which is typically measured on benchmarks [3, 9]. Indeed, previous evaluations have shown that IE methods already produce good results, but the research is still ongoing [3, 4, 9, 12, 20]. Primarily driven by the development of modern language models like BERT [4], IE has made a huge step forward.

However these systems rely on supervised learning and thus need large-scale training data that cannot be reliably transferred across domains. In brief, although supervised methods are up to the job with reasonable quality, their practical application comes at high costs. The expenses for supervision lead to the design of zero-shot, semi-supervised, and distant supervised extraction methods (see [28] for a good overview).

Open Information Extraction. Instead of designing extraction systems for each domain, methods like unsupervised information extraction (OpenIE) are proposed to change the game [20]. OpenIE aims to extract knowledge from texts without knowing the entity and relation domains a-priori [20, 28]. While supervised (closed) methods focus on domain-specific and relevant relations and concepts, open methods are more flexible and may be applied across domains [20, 28]. Vashishth proposed CESI to canonicalize OpenIE extractions by clustering noun and verb phases with the help of side information [25]. However CESI was analyzed for short phrases that refer to precise entities. In addition studies have shown that OpenIE methods may struggle to handle scientific texts well because sentences are often long and domain-specific vocabulary terms are used [5]. While research in both directions (open and closed) is still ongoing, some works bridge the gap between both worlds: Kruiper et al. propose the task of Semi-Open Relation extraction [14], i.e., they use domain-specific information to filter irrelevant open information extractions. Similarly, we showed that domain-specific filtering of OpenIE outputs could yield helpful results [12].

Information Extraction in Digital Libraries. Digital libraries are interested in practical IE workflows to allow novel applications; See this tutorial at JCDL2016 [29]. IE can allow literature-based discovery workflows, which have been studied on DBpedia [24]. The extraction of entities and relations is therefore challenging. That is why modern approaches build upon language models and supervision for a reliable extraction [22]. These language models require extensive computational resources for training and application [4, 15]. Good examples for IE are DBpedia [2] that was harvested from Wikipedia infoboxes or the SemMedDB, which is a collection of biomedical statements harvested from PubMed [8, 30]. Hristovski et al. have used the SemMedDB to perform knowledge discovery [6]. Nevertheless the construction of SemMedDB required biomedical experiences to define hand-written rules for the extraction. In contrast to the previous works, our work focused on nearly-unsupervised extraction workflows that do not rely on training data for the extraction phase.

6 CONCLUSION

In this paper we have studied nearly-unsupervised extraction workflows for a practical application in digital libraries. We focused on three different domains to generalize our findings, namely the encyclopedia Wikipedia, pharmacy, and political sciences. First, the scalability of the investigated methods was acceptable for our partners. Second, unsupervised extraction workflows required intensive cleaning and canonicalization to result in precise semantics. Thus they do not work out-of-the-box and reliably canonicalize OpenIE verb phrases remains an open issue. Although such cleaning can be exhausting, the pharmaceutical case study yielded a novel retrieval service. Such a service would not have been possible when training data must have been collected for each relation. In addition, not filtering complex object phrases can allow the construction of semi-structured knowledge bases or enrich the original texts, e.g., show all actions of Albert Einstein. In conclusion, unsupervised extraction workflows are worth studying in digital libraries. They come with limitations and require cleaning, but they entirely bypass the lack of training data in the extraction phase.

ACKNOWLEDGMENT

Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): PubPharm – the Specialized Information Service for Pharmacy (Gepris 267140244).

REFERENCES

[1] Giuseppe Attardi. 2015. WikiExtractor. https://github.com/attardi/wikiextractor.

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer, Busan, Korea, 722–735.

[3] Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. 2019. CaRB: A Crowdsourced Benchmark for Open IE. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 6262–6267. https://doi.org/10.18653/v1/D19-1651

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[5] Paul Groth, Mike Lauruhn, Antony Scerri, and Ron Daniel Jr. 2018. Open Information Extraction on Scientific Text: An Evaluation. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 3414–3423. https://aclanthology.org/C18-1289

[6] Dimitar Hristovski, Andrej Kastrin, Dejan Dinevski, and Thomas C Rindflesch. 2015. Constructing a Graph Database for Semantic Literature-Based Discovery. *Studies in health technology and informatics* 216 (2015), 1094.

[7] Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. 2019. Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. In *Proceedings of the 10th International Conference on Knowledge Capture* (Marina Del Rey, CA, USA) *(K-CAP '19)*. Association for Computing Machinery, New York, NY, USA, 243–246. https://doi.org/10.1145/3360901.3364435

[8] Halil Kilicoglu, Dongwook Shin, Marcelo Fiszman, Graciela Rosemblat, and Thomas C. Rindflesch. 2012. SemMedDB: a PubMed-scale repository of biomedical semantic predications. *Bioinformatics* 28, 23 (10 2012), 3158–3160. https://doi.org/10.1093/bioinformatics/bts591

[9] Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. 2020. OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 3748–3761. https://doi.org/10.18653/v1/2020.emnlp-main.306

[10] Hermann Kroll, Judy Al-Chaar, and Wolf-Tilo Balke. 2021. Open Information Extraction in Digital Libraries: Current Challenges and Open Research Questions. In *Proceedings of the Workshop on Digital Infrastructures for Scholarly Content Objects (DISCO 2021) co-located with ACM/IEEE Joint Conference on Digital Libraries 2021(JCDL 2021), Online , September 30, 2021 (CEUR Workshop Proceedings, Vol. 2976)*, Wolf-Tilo Balke, Anita de Waard, Yuanxi Fu, Bolin Hua, Jodi Schneider, Ningyuan Song, and Xiaoguang Wang (Eds.). CEUR-WS.org, Online, 14–18. http://ceur-ws.org/Vol-2976/short-1.pdf

[11] Hermann Kroll, Jan-Christoph Kalo, Denis Nagel, Stephan Mennicke, and Wolf-Tilo Balke. 2020. Context-Compatible Information Fusion for Scientific Knowledge Graphs. In *Digital Libraries for Open Knowledge*, Mark Hall, Tanja Merčun, Thomas Risse, and Fabien Duchateau (Eds.). Springer International Publishing, Cham, 33–47. https://doi.org/10.1007/978-3-030-54956-5_3

[12] Hermann Kroll, Jan Pirklbauer, and Wolf-Tilo Balke. 2021. A Toolbox for the Nearly-Unsupervised Construction of Digital Library Knowledge Graphs. In *ACM/IEEE Joint Conference on Digital Libraries, JCDL 2021, Champaign, IL, USA, September 27-30, 2021*, J. Stephen Downie, Dana McKay, Hussein Suleman, David M. Nichols, and Faryaneh Poursardar (Eds.). IEEE, Champaign, IL, USA, 21–30. https://doi.org/10.1109/JCDL52503.2021.00014

[13] Hermann Kroll, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. 2021. Narrative Query Graphs for Entity-Interaction-Aware Document Retrieval. In *Towards Open and Trustworthy Digital Societies - 23rd International Conference on Asia-Pacific Digital Libraries, ICADL 2021, Virtual Event, December 1-3, 2021, Proceedings (Lecture Notes in Computer Science, Vol. 13133)*, Hao-Ren Ke, Chei Sian Lee, and Kazunari Sugiyama (Eds.). Springer, Online, 80–95. https://doi.org/10.1007/978-3-030-91669-5_7

[14] Ruben Kruiper, Julian Vincent, Jessica Chen-Burger, Marc Desmulliez, and Ioannis Konstas. 2020. In Layman's Terms: Semi-Open Relation Extraction from Scientific Texts. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 1489–1500. https://doi.org/10.18653/v1/2020.acl-main.137

[15] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (09 2019), 1234–1240. https://doi.org/10.1093/bioinformatics/btz682

[16] Ying Liu, Kun Bai, Prasenjit Mitra, and C. Lee Giles. 2007. TableSeer: Automatic Table Metadata Extraction and Searching in Digital Libraries. In *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries* (Vancouver, BC, Canada) *(JCDL '07)*. Association for Computing Machinery, New York, NY, USA, 91–100. https://doi.org/10.1145/1255175.1255193

[17] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. Association for Computational Linguistics, Baltimore, Maryland, USA, 55–60.

[18] David Mendez, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, and al. 2018. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research* 47, D1 (11 2018), D930–D940. https://doi.org/10.1093/nar/gky1075

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. ICLR, Scottsdale, Arizona, USA.

[20] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A Survey on Open Information Extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 3866–3878. https://aclanthology.org/C18-1326

[21] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Online, 101–108. https://doi.org/10.18653/v1/2020.acl-demos.14

[22] Santosh Tokala Yaswanth Sri Sai, Prantika Chakraborty, Sudakshina Dutta, Debarshi Kumar Sanyal, and Partha Pratim Das. 2021. Joint Entity and Relation Extraction from Scientific Documents: Role of Linguistic Information and Entity Types. In *Proceedings of the 2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE 2021) co-located with JCDL 2021, Virtual Event, September 30th, 2021 (CEUR Workshop Proceedings, Vol. 3004)*, Chengzhi Zhang, Philipp Mayr, Wei Lu, and Yi Zhang (Eds.). CEUR-WS.org, Online, 15–19. http://ceur-ws.org/Vol-3004/paper2.pdf

[23] Tim Schardelmann and Wolfgang Otto. 2018. POLLUX – von der Bedarfsanalyse zur technischen Umsetzung. *Bibliotheksdienst* 52, 3-4 (2018), 225–234. https://doi.org/10.1515/bd-2018-0029

[24] Menasha Thilakaratne, Katrina Falkner, and Thushari Atapattu. 2020. *Information Extraction in Digital Libraries: First Steps towards Portability of LBD Workflow*. Association for Computing Machinery, New York, NY, USA, 345–348. https://doi.org/10.1145/3383583.3398607

[25] Shikhar Vashishth, Prince Jain, and Partha Talukdar. 2018. CESI: Canonicalizing Open Knowledge Bases Using Embeddings and Side Information. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1317–1327. https://doi.org/10.1145/3178876.3186030

[26] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[27] Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2013. PubTator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Res.* 41, Web Server issue (July 2013), W518–22.

[28] Gerhard Weikum, Xin Luna Dong, Simon Razniewski, and Fabian M. Suchanek. 2021. Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases. , 108–490 pages. https://doi.org/10.1561/1900000064

[29] Kyle Williams, Jian Wu, Zhaohui Wu, and C. Lee Giles. 2016. Information Extraction for Scholarly Digital Libraries. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries* (Newark, New Jersey, USA) *(JCDL '16)*. Association for Computing Machinery, New York, NY, USA, 287–288. https://doi.org/10.1145/2910896.2925430

[30] Rui Zhang, Michael J. Cairelli, Marcelo Fiszman, Graciela Rosemblat, Halil Kilicoglu, Thomas C. Rindflesch, Serguei V. Pakhomov, and Genevieve B. Melton. 2014. Using semantic predications to uncover drug–drug interactions in clinical data. *Journal of Biomedical Informatics* 49 (2014), 134–147. https://doi.org/10.1016/j.jbi.2014.01.004

[31] Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. 2019. BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific data* 6, 1 (2019), 1–9.

B.6. IJDL 2023a: A discovery system for narrative query graphs: entity-interaction-aware document retrieval

---

IJDL'23a

Hermann Kroll, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. "A discovery system for narrative query graphs: entity-interaction-aware document retrieval". International Journal on Digital Libraries (IJDL) 2023. DOI: https://doi.org/10.1007/s00799-023-00356-3

International Journal on Digital Libraries
https://doi.org/10.1007/s00799-023-00356-3

A discovery system for narrative query graphs: entity-interaction-aware document retrieval

Hermann Kroll[1] · Jan Pirklbauer[1] · Jan-Christoph Kalo[2] · Morris Kunz[1] · Johannes Ruthmann[1] · Wolf-Tilo Balke[1]

Received: 21 July 2022 / Revised: 19 January 2023 / Accepted: 16 March 2023

Abstract
Finding relevant publications in the scientific domain can be quite tedious: Accessing large-scale document collections often means to formulate an initial keyword-based query followed by many refinements to retrieve a *sufficiently complete, yet manageable* set of documents to satisfy one's information need. Since keyword-based search limits researchers to formulating their information needs as a set of unconnected keywords, retrieval systems try to guess each user's intent. In contrast, distilling short narratives of the searchers' information needs into simple, yet precise entity-interaction graph patterns provides all information needed for a precise search. As an additional benefit, such graph patterns may also feature variable nodes to flexibly allow for different substitutions of entities taking a specified role. An evaluation over the PubMed document collection quantifies the gains in precision for our novel entity-interaction-aware search. Moreover, we perform expert interviews and a questionnaire to verify the usefulness of our system in practice. This paper extends our previous work by giving a comprehensive overview about the discovery system to realize narrative query graph retrieval.

Keywords Narrative information access · Narrative queries · Graph-based retrieval · Digital libraries

1 Introduction

PubMed, the world's most extensive digital library for biomedical research, consists of about 34 million publications and is currently growing by more than one million publications each year. Accessing such an extensive collection by simple means such as keyword-based retrieval over publication texts is a challenge for researchers, since they simply cannot read through hundreds of possibly relevant documents, yet cannot afford to miss relevant information in retrieval tasks. Indeed, there is a dire need for retrieval tools tailored to specific information needs in order to solve the above conflict. For such tools, deeper knowledge about the particular task at hand and the specific semantics involved is essential. Taking a closer look at the nature of scientific information search, interactions between entities can be seen to represent a short narrative [15]—a short story of interest: how or why entities interact, in what sequence or roles they occur, and what the result or purpose of their interaction is [6, 15]. This article is an extended version of our previous article [18].

Indeed, an extensive query log analysis on PubMed in [10] clearly shows that researchers in the biomedical domain are often interested in interactions between entities such as drugs, genes, and diseases. Among other results, the authors report that (a) on average significantly more keywords are used in PubMed queries than in typical Web searches, (b) result set sizes reach an average of (rather unmanageable) 14,050 documents, and (c) keyword queries are on average 4.3 times refined and often include more specific information about

✉ Hermann Kroll
kroll@ifis.cs.tu-bs.de

✉ Jan Pirklbauer
j.pirklbauer@tu-bs.de

✉ Jan-Christoph Kalo
j.c.kalo@vu.nl

✉ Morris Kunz
morris.kunz@tu-bs.de

✉ Johannes Ruthmann
j.ruthmann@tu-bs.de

✉ Wolf-Tilo Balke
balke@ifis.cs.tu-bs.de

[1] Institute for Information Systems, TU Braunschweig, Mühlenpfordtstr. 23, 38106 Braunschweig, Lower Saxony, Germany

[2] Knowledge Representation and Reasoning Group, VU Amsterdam, De Boelelaan 1111, 1081 HV Amsterdam, The Netherlands

Published online: 24 April 2023

the keywords' intended semantic relationships, e.g., *myocardial infarction AND aspirin* may be refined to *myocardial infarction prevention AND aspirin*. Given all these observations, native support for entity-interaction-aware retrieval tasks can be expected to be extremely useful for PubMed information searches and is quite promising to generalize to other kinds of scientific domains, too. However, searching scientific document collections curated by digital libraries for such narratives is tedious when restricted to keyword-based search, since the same narrative can be paraphrased in countless ways [1, 10].

Therefore, we introduce the novel concept of *narrative query graphs for scientific document retrieval* enabling users to formulate their information needs as entity-interaction queries explicitly. Complex interactions between entities can be precisely specified: simple interactions between two entities are expressed by a basic query graph consisting of two nodes and a labeled edge between them. Of course, by adding more edges and entity nodes, these basic graph patterns can be combined to form arbitrarily complex graph patterns to address highly specialized information needs. Moreover, narrative query graphs support *variable nodes* supporting a far broader expressiveness than keyword-based queries. As an example, a researcher might search for treatments of *some disease* using *simvastatin*. While keyword-based searches would broaden the scope of the query far in excess of the user intent by just omitting any specific disease's name, narrative query graphs can focus the search by using a variable node to find documents that describe treatments of *simvastatin* facilitated by an entity of the type *disease*.

In contrast to query languages for knowledge graphs, our discovery system does not match the query against a single knowledge graph. Instead, we must on-the-fly match the query against several document graphs, i.e., the document itself stays in the focus of the system. And moreover, if variables are used in searches, the result lists require novel visualizations, e.g., clustering document result lists by possible node substitutions to get an entity-centric literature overview. Since our document graphs are extracted from texts with automated methods, we provide provenance information to explain why a document matches the query.

Whereas our previous article [18] focused on benefits of the overall retrieval, this article extends the previous work by describing the extraction workflow in more detail, and the overall discovery system with its key features. In addition, we utilize an Open IE system for a retrieval quality comparison. We also point out limitations that have to be faced in the future to further improve this kind of retrieval. In summary, our contributions are:

1. We proposed narrative query graphs for scientific document retrieval enabling fine-grained modeling of users' information needs. Moreover, we boosted query expressiveness by introducing variable nodes for document retrieval.
2. We developed a discovery system that processes arbitrary narrative query graphs over the biomedical literature. As a showcase, the service performs searches on 34 million PubMed titles and abstracts in real time.
3. We extended our previous work by stating details on the extraction quality. In addition, we described system details required to implement narrative query graph retrieval.
4. We evaluated our system in two ways: On the one hand, we demonstrated our retrieval system's usefulness and superiority over keyword-based search on the PubMed digital library in a manual evaluation which included practitioners from the pharmaceutical domain. On the other hand, we performed interviews and a questionnaire with eight biomedical experts who face the search for literature on a daily basis.

2 Related work

Relevant research areas to this work are narrative information access, machine learning for retrieval, graph-based retrieval, document representations, and scholarly knowledge graphs.

2.1 Narrative information access

Narrative query graphs are designed to offer complex querying capabilities over scientific document collections aiming at high precision results. Focusing on retrieving entity interactions, they are a subset of our conceptual overlay model for representing narrative information [15]. Our conceptual model narrative allows users to state their information needs as a complex and nested graph model involving entities, events, literals, and even nested literals. We then understand the narrative as a *logical overlay over knowledge repositories*, i.e., we try to find evidence by binding parts of modeled narrative against real-world data. We discussed suitable methods and the technical challenges to bind against document collections in [16]. Here we are looking for scientific narratives that may require combining several statements. We already know that combining statements from different scientific contexts can be a serious threat to the overall result quality [14]. Our proposed discovery system requires that a whole information need must be matched within a small abstract because we assume the context to be stable within it [14, 20].

This work builds upon our previous work [18]. In extension to [18], this paper describes the complete retrieval method and evaluation of narrative query graphs for document retrieval. Therefore, we extend our previous work by giving insights into our data model, corresponding extraction

statistics, and the complete extraction workflow. We also utilize an Open IE system for a retrieval quality comparison. In addition, we describe the discovery system in more detail. Mainly, we discuss our design decisions to engage technical challenges. We also show extensions of our original discovery system: A concept selection picker, user feedback options, and Drug Overviews (Drug-centered overviews generated from the literature). We finally point out limitations that have to be faced in the future to improve this kind of retrieval further.

2.2 Machine learning for retrieval

Modern personalized systems try to guess each user's intent and automatically provide more relevant results by query expansion; see [1] for a good overview. Mohan et al. focus on information retrieval of biomedical texts in PubMed [28]. The authors derive a training and test set by analyzing PubMed query logs and train a deep neural network to improve literature search. Entity-based language models are used to distinguish between a term-based and entity-based search to increase the retrieval quality [33]. Yet, while a variety of approaches to improve result rankings by learning how a query is related to some document [28, 43, 45] have been proposed, gathering enough training data to effectively train a system for all different kinds of scientific domains seems impossible. Specialized information needs, which are rarely searched, are hardly covered in such models.

2.3 Graph-based retrieval

Using graph-based methods for textual information retrieval gained in popularity recently [6, 35, 36, 45], for instance, Dietz et al. discuss the opportunities of entity linking and relation extraction to enhance query processing for keyword-based systems [6], and Zhao et al. demonstrate the usefulness of graph-based document representations for precise biomedical literature retrieval [45]. Kadry et al. also include entity and relationship information from the text as a learning-to-rank task to improve support passage retrieval [12]. Besides, Spitz and Gertz built a graph representation for Wikipedia to answer queries about events and entities more precisely [35]. But in contrast to our work, those approaches focus on unlabeled graphs or include relationships only partially.

2.4 Document representation

Croft et al. proposed a network representation of documents and their corresponding terms [5]. Such a network representation supports effective retrieval because documents and terms can easily be linked and traversed in the retrieval phase. Further, [4] demonstrated that using a network representation can enhance the effectiveness of a retrieval system while allowing the implementation of several search strategies.

France has developed the MARIAN system that allows an effective representation and retrieval of relationships between digital library objects [9], e.g., how library objects are linked. Another example of an early intelligent retrieval system was the CODER system [38]. The system was implemented in a modular fashion allowing to test novel retrieval strategies. Chen has developed an object-oriented model called LEND (Large External object-oriented Network Database) model [3]. This model supports the representation and querying of graph-structured data.

While the research on effective document representations for retrieval has a long-standing tradition and is still ongoing, the previous works focused on retrieving documents based either on textual content or metadata. In contrast, our work is focused on the representation of documents as entity-interaction-aware graphs, i.e., we break down document texts into graphs.

2.5 Scholarly knowledge bases

Several projects aim to capture knowledge about the academic world as graph representations, e.g., the Microsoft Academic Knowledge Graph [8], the Open Research Knowledge Graph [11], and OpenAlex [31]. Another example is GrapAl, a graph database of academic literature that is designed to assist academic literature search by supporting a structured querying language, namely Cypher [2]. GrapAl mainly consists of traditional metadata like authors, citations, and publication information but also includes entities and relationship mentions. However, complex entity interactions are not supported, as only a few basic relationships per paper are annotated.

QKBfly is a search system that extracts facts from text to support question answering [29]. It constructs a knowledge base for ad hoc question answering during query time that provides journalists with the latest information about emergent topics. However, they focus on retrieving relevant facts concerning a single entity. In contrast, we focus on document retrieval for complex entity interactions, i.e., we match structured queries against documents to retain the original contexts.

In contrast to the previous works, this paper introduces a complete discovery system involving extraction, retrieval, user interface design, effectiveness evaluation, and user studies.

3 Narrative query graphs

Entities represent things of interest in a specific domain: drugs and diseases are prime examples in the biomedical

domain. An entity $e = (ID, type)$, where id is a unique identifier and $type$ the entity type. To give an example, we may represent the drug *simvastatin* by its identifier and entity type as follows: $e_{simvastatin} = (D019821, Drug)$. Typically, entities are defined by predefined ontologies, taxonomies, or controlled vocabularies, such as NLM's MeSH or EMBL's ChEBI. We denote the set of known entities as $\mathcal{E}$. Since we aim to find entity interactions in texts, we need to know where entities are mentioned. In typical natural language processing, each sentence is represented as a sequence of tokens, i.e., single words. Therefore, an **entity alignment** maps a token or a sequence of tokens to an entity from $\mathcal{E}$ if the tokens refer to it.

Entities might also be classes as well, e.g., the entity *diabetes mellitus* (Disease) refers to a class of specialized diabetes diseases such as *DM type 1* and *DM type 2*. Thus, these classes can be arranged in subclass relations, i.e., *DM type 1* is the subclass of general *diabetes mellitus*. We define the following function to derive the set of all subclasses of an entity: $subclasses(e) = \{e_i \mid e_i \text{ is subclass of } e\}$. If an entity e is not a class or does not have any subclasses, the function does simply return e.

We call an interaction between two entities a **statement** following the idea of knowledge representation in the Resource Description Framework (RDF) [26]. Hence, we define a **statement** as triple (s, p, o) where $s, o \in \mathcal{E}$ and $p \in \Sigma$. Σ represents the set of all interactions we are interested in. We focus only on interactions between entities, unlike RDF, where objects might be literals too. For example, a *treatment* interaction between *simvastatin* and *hypercholesterolemia* is encoded as $(e_{simvastatin}, treats, e_{hypercholesterolemia})$. We call a set of extractions from a single document a **document graph**.

Document graphs support narrative querying, i.e., the query is answered by matching the query against the document's graph. Suppose a user formulates a query like $(e_{simvastatin}, treats, e_{hypercholesterolemia})$. In that case, our system retrieves a set of documents containing the searched statement. Narrative query graphs may include typed variable nodes as well. A user might query $(e_{simvastatin}, treats, ?X(Disease))$, asking for documents containing *some* disease treatment with *simvastatin*. Hence, all documents that include *simvastatin* treatments for diseases are proper matches. Formally, we denote the set of all variable nodes as $\mathcal{V}$. Variable nodes consist of a name and an entity type to support querying for entity types. We also support the entity type *All* to query for arbitrary entities. We write variable nodes by a leading question mark. Hence, a narrative query graph might include entities stemming from $\mathcal{E}$ and variable nodes from $\mathcal{V}$. Formally, a **fact pattern** is a triple $fp = (s, p, o)$ where $s, o \in (\mathcal{E} \cup \mathcal{V})$ and $p \in \Sigma$. A **narrative query graph** q is a set of fact patterns similar to SPARQL's basic graph patterns [30]. When executed, the query produces one or more matches μ by binding the variable symbols to actual entities, i.e., $\mu : \mathcal{V} \rightarrow \mathcal{E}$ is a partial function. If several fact patterns are queried, all patterns must be contained within a document forming a proper query answer. Suppose queries include entities that are classes and have subclasses. In that case, the query will be expanded to also query for these subclasses, i.e., direct and transitive subclasses. We do this by applying the *subclasses* function on every entity in the query.

4 Document graphs

The discovery system requires a transformation of documents' texts into a document graph representation. This step involves entity linking, information extraction, cleaning, and loading. It extracts document graphs from text and stores them in a structured repository. Then the system takes narrative query graphs as its input and performs graph pattern matching. All document graphs that match the query are returned to the users. In this section, we describe all relevant details about the extraction process.

4.1 Document graph extraction

Linking entities and extracting statements from texts form the essential core of mining document graphs. Therefore, we analyzed a plethora of different domain-specific methods like supervised annotations tools (e.g., TaggerOne [24] and GNormPlus [40]). For the extraction phase, we analyzed supervised extraction tools that aim to reduce the need for training data (e.g., Snorkel [32] and DeepDive [34]). However, all of these supervised methods still require training data and are thus specialized for a certain domain. Although their quality is often very high, we went for a different approach: unsupervised linking and extraction. Our goal was to design and utilize methods that deliver sufficient quality and could still be transferred to another domain. With such a set of methods realizing the service in a different domain seems not too far-fetched.

Our efforts yielded a toolbox that we shared as open source[1]: *A Toolbox for the Nearly-Unsupervised Construction of Digital Library Knowledge Graphs* [17]. The toolbox includes methods for unsupervised entity linking, interfaces to unsupervised extraction methods, and cleaning methods to obtain a sufficient quality. We call it nearly unsupervised because the toolbox requires the design of two different vocabularies: (1) An entity vocabulary including all entities of interest. Each entry consists of an unique entity id, an entity type, an entity name, and a list of synonyms. (2) A relation vocabulary including all relations of interest. Each entry consists of a relation and a set of synonyms. For details

[1] https://github.com/HermannKroll/KGExtractionToolbox.

Table 1 Number of entries in our entity vocabulary

Entity type	#Distinct entries	#Terms
Chemical	146	1,850
Disease	5051	57,295
Drug	45,200	69,767
Dosage form	136	6,891
Excipient	12,951	132,704
Lab method	528	5,742
Method	2,512	23,182
Plant family	2,818	2,818
Vaccines	161	1032
Sum	69,503	300,134

about the actual extraction quality, we refer the reader to our original toolbox paper for a quantitative evaluation in the biomedical domain [17] and our follow-up work on a qualitative analysis for three corpora: pharmaceutical literature, the Wikipedia encyclopedia, and political sciences literature [19]. In brief, our main findings were: First, entity and relation vocabularies are a fixed requirement to apply the toolbox. Second, the quality clearly lags behind supervised entity linking and information extraction methods. Third, the canonicalization of verb phrases to precise relations is still an open issue in some cases. Although missing vocabularies, limited extraction quality (especially recall), and open canonicalization issues must be tackled in the future, we still argue that nearly unsupervised workflows are worth studying in digital libraries because they completely bypass training data in the extraction phase [19].

4.2 Pharmaceutical entity linking

For our retrieval system we designed an entity vocabulary that comprises *chemicals, drugs, diseases, dosage forms, excipients, plant families, lab methods, methods, and vaccines*. We derived vocabulary entries from the biomedical specialized database ChEMBL [27], the Medical Subject Headings (MeSH),[2] and Wikidata [37]. In cooperation with two pharmaceutical domain experts, we manually selected suitable subsets of the previous vocabularies and manually formulated missing entities such as specialized dosage forms (e.g., nanoparticles). In summary, we derived 69,503 distinct entities with 300,134 terms. A list of all entity types and their corresponding vocabulary size is shown in Table 1.

We then evaluated our entity linking quality for our work [17]. For chemicals and diseases, we selected two biomedical benchmarks: BioCreative V CD-R and NCBI Disease. We used the given vocabularies for these benchmarks and applied our dictionary-based entity linker. In addition, we randomly sampled 50 entity annotations for drug, dosage forms, and plant families. We presented these annotations to two pharmaceutical domain experts. Together they decided for each text span if it was linked correctly to the given entity. However, we could thus only compute the precision for these three entity types. The results are shown in Table 2.

Diseases could be linked with a precision between 55.1 and 82.8%. The recall was between 62.0 and 63.3%. For chemicals, we obtained a precision of 76.6% and 78.7%. In our sample-based evaluation, we obtained a precision of 90% for drugs, 82% for dosage forms and plant families, and 74% for excipients. We evaluated the linker against state-of-the-art supervised methods in a previous publication; see [17] for more details.

To give a few more insights: We decided not to link trade names of drugs. These trade names included words like *horse*, *man*, *power*, etc., which were often linked incorrectly. For our discovery system, the main issue was that a few frequently but wrongly linked entities would be annoying for users to handle. We handled this issue by applying two strategies:

1. We went through the 500-top-frequently tagged entities and removed often wrongly linked entities from our vocabularies.
2. We applied a special cleaning rule for plant families like *paris* because they were of high interest for our purposes but often linked wrongly.

We checked whether one of 82 regular expressions (e.g., *Traditional Medicine* or *phytotherap**) could be matched against the same abstract. We kept the linked plant families only if at least one of these expressions could be successfully matched. In summary, dictionary-based entity linkers do have their limitations. But we did not need training data for the linking step and the quality was sufficient for us to continue.

In addition to our entity linking workflow, we integrated annotations from the PubTator Central service.[3] This service is hosted by the National Library of Medicine (NLM) and allowed us to retrieve annotations for diseases, chemicals, genes, and species. We analyzed the annotation service in cooperation with our domain experts. For our goal, we found the chemical annotations to be too general. That is why we integrated only diseases, genes, and species annotations. Details about PubTator Central can be found in [39, 42].

4.3 Pharmaceutical inf. extraction

We had to extract statements between the detected entities for the actual document graph representation. Although super-

[2] https://meshb.nlm.nih.gov.

[3] https://www.ncbi.nlm.nih.gov/research/pubtator/.

Table 2 Evaluation of our entity linking step: linking of chemicals and diseases was tested on two established biomedical benchmarks (BioCreative V CD-R [41] and NCBI disease [7]). For drugs, dosage forms, excipients, and plant families, we performed a manual evaluation of 50 random-sampled annotations

Entity type	Benchmark	Precision	Recall	F1
Chemical	BioCreative V CD-R	76.6%	78.7%	77.6%
Disease	BioCreative V CD-R	82.8%	62.0%	70.9%
Disease	NCBI disease	74.5%	55.1%	63.3%
Drug	Sample	90.0%	–	–
Excipient	Sample	74.0%	–	–
Dosage form	Sample	82.0%	–	–
Plant family	Sample	82.0%	–	–

vised extraction methods would have likely achieved a better extraction quality, we decided to build upon unsupervised extraction methods. The quality of existing open information extraction like OpenIE 6 sounded promising [13], but we found that open information extraction methods highly lack recall when processing biomedical texts; see the evaluation in [17]. That is why we developed a recall-oriented extraction technique **PathIE** in [17] that flexibly extracts interactions between entities via a path-based method. The central idea was to take sentences in which at least two different entities have been detected. Then, the shortest path between the entities in the grammatical structure of the sentence was computed. All verb phrases and keywords (that have been specified in the relation vocabulary) were considered for extraction. PathIE then yielded triples consisting of two entities and a predicate (either a verb phrase or a given keyword like treatment).

PathIE yielded many synonymous predicates (treats, aids, prevents, etc.) that represent the relation *treats*. The toolbox implemented a canonicalization procedure to unify synonymous predicates to precise relations. The procedure works as follows: Given a pre-designed relation vocabulary, all terms that appear directly in the vocabulary are mapped to the corresponding relation. In addition, we used the optional word embedding feature to also canonicalize similar verb phrases, i.e., verb phrases that were similar to entries in the vocabulary were also mapped to the corresponding relation.

The pharmaceutical relation vocabulary had to have precise semantics and was built with the help of two domain experts. The relation vocabulary included 60 entries (10 relations plus 50 synonyms) for the cleaning step. As a biomedical word embedding, we used the pre-trained word embedding from [44]. Then, we applied the toolbox canonicalization procedure. The cleaning allowed users to formulate their queries based on a well-curated vocabulary of entity interactions in the domain of interest. To increase the quality of extractions, we introduced type constraints by providing fixed domain and range types for each interaction. Extracted interactions that did not meet the interaction's type constraints were removed. For example, the interaction *treats* is typed, i.e., the subject must be a drug, and the object must be a disease or species. Some interactions in our vocabulary like *induces* or *associated* are more general and thus were not annotated with type constraints. We found those type constraints worked well if the relations are directed, e.g., a *treats* relation between a drug and a disease [19]. If relations are not directed, PathIE often messes up the direction by design, e.g., a causes b instead of b causes a.

Table 3 CDR2015 benchmark evaluation [41]. The table reports the extraction quality for CoreNLP OpenIE, PathIE, and best reported baselines

Method	Prec. (%)	Rec. (%)	F1 (%)
CoreNLP OpenIE	64.9	5.8	10.6
PathIE	50.8	31.7	39.1
Best precision	90.5	80.8	85.4
Best recall	86.1	86.2	86.1

The following experiment has already been reported in [17]. To test our extraction pipeline, we utilized the benchmark of [41]. The benchmark provides abstracts and entity annotations and requires extracting *induce* relations between chemicals and diseases. We loaded the abstracts and annotations. Then, we applied PathIE and the canonicalization with our relation vocabulary. For comparison, we applied the Stanford CoreNLP method [25] from our toolbox. The results are reported in Table 3. The table lists the precision, recall and F1 score when extracting statements with our toolbox using either CoreNLP OpenIE or PathIE. In addition, we included the workshop's best-performing systems [41] concerning precision and recall in Table 3.

In brief, PathIE achieved an F1 score of 39.1%, whereas supervised methods achieved an F1 score between 85.4% and 86.1%. CoreNLP achieved a precision of 64.9% but a recall of only 5.8%. Due to the low recall, we decided to use PathIE for our retrieval system. However, we report on another comparison between PathIE and CoreNLP for the actual retrieval in our evaluation section. We refer the reader for more details about the toolbox's extraction quality to our previous publications [17, 19].

Table 4 Number of extracted statements per relation

Relation	#Statements
Associated	182,258,817
Method	153,040,391
Compares	19,552,314
Induces	12,012,245
Treats	9,938,585
Administered	7,098,069
Decreases	4,299,302
Interacts	4,002,765
Inhibits	1,741,788
Metabolizes	112,822
Sum	394,057,098

For our service, we applied three special rules:

1. Instead of removing statements, we mapped all statements that hurt their type constraints to the *associated* relation since both entities were still in *some way* associated in the sentence.
2. All statements including an entity of type *method* or *lab method* were mapped to the relation *method*.
3. All statements including an entity of type *dosage form* were mapped to the relation *administered*.

We applied the first rule to allow users to search for *arbitrary* relations between entities. Rules 2 and 3 were applied to have special relations for methods and dosage forms. In summary, we extracted 394 M statements and ten unique relations. Statistics are reported in Table 4.

5 Discovery system

In the following section, we describe our discovery system for entity-interaction-aware document retrieval. On the one hand, the system must be capable of answering narrative query graphs. On the other hand, querying in this way requires suitable interfaces for a suitable user experience. Moreover, our discovery allows users to integrate variables into their searches which asks for novel visualization in the user interface. Our discovery system is freely accessible.[4] A systematic overview of the whole system is depicted in Fig. 1. In the following, we report on the current system's version (July 2022).

5.1 Discovery content

We integrated the complete NLM Medline collection (about 34 M publications), i.e., the content of the PubMed search engine. Therefore, we obtained the titles and abstracts plus entity annotations from the PubTator service. In addition, we loaded metadata for the publications, e.g., authors, journal, publication year, etc. We obtained the metadata from the NLM's official XML dumps. In joint cooperation with ZB MED and the Robert Koch-Institute in Germany, we integrated about 45k pre-prints from PreView [22, 23] (ZB MED service) for COVID-19 questions. In both cases, we loaded each publication's titles, abstracts, and metadata (authors, journal, etc.). We did not consider full texts. We then applied our entity linking, information extraction, and cleaning workflow to transform each document's text into a document graph. Note that we concatenated a document's title and

[4] http://www.narrative.pubpharm.de.

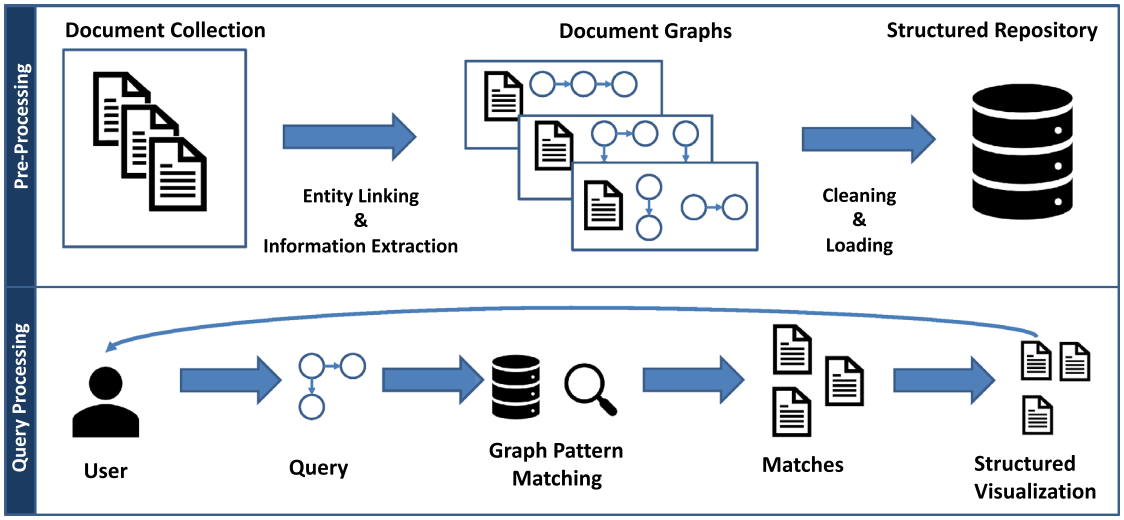


Fig. 1 System overview: document graphs are extracted from texts, cleaned, indexed, and loaded into a structured repository. Narrative query graphs are then matched against this repository to retrieve the respective documents

Table 5 Statistics about the content of our system

Name	#Docs	#Graphs
PubMed	34 M	19.5 M
COVID-19 Pre-Prints	45k	24.6k

abstract to derive a single text for the graph transformation. In addition, we could not extract a document graph for all documents since we neither detected entities nor interactions in them. For instance, we extracted *19.5 M* graphs from *34 M* PubMed documents and *24.6k* from *45k* COVID-19 preprints. The statistics are reported in Table 5. We developed scripts to update the service content at periodic intervals. At the moment, the discovery system cannot search for documents from which we cannot extract a single statement. In the future, a more flexible query model could allow a search for that documents, i.e., by rewriting a narrative query graph to a set of keywords if no match could be found otherwise.

5.2 Data representation

In the design of our discovery system, we had two central requirements: (1) process narrative query graphs and (2) deliver a suitable user experience. In early talks with domain experts, we found that explainability was relevant, i.e., visualizing why documents should match their information needs. With that in mind, the question of how we should represent our data was raised. In an early phase, we decided to store our data in a relational database because they are well supported (reliable software and interfaces) and our data could be broken in a relational fashion. For example, the service returns document titles, sentences, entity annotations, and extraction information to explain matches to the user. In this way a central document table allowed us to join the corresponding information if necessary. An overview of our relational schema is shown in Fig. 2.

The document table stores the title and abstract for each document. Each document is identified by an ID and the corresponding collection (e.g., PubMed). The tag table stores entity annotations, the predication table stores the extracted statement, and the document metadata tables stores information about a publication's authors, journals, etc. To explain matches to the user, we integrated a sentence table to link an extracted statement to its sentence origin. We split sentences and statements to reduce redundancy - several statements might have been extracted from the same sentences. Sentences are identified by an MD5-hash for each document collection. To accelerate the actual retrieval of documents' metadata, we created a materialized view *metadata service* which contains titles and metadata of documents in which at least a single statement was extracted. On the one hand, titles and metadata can, in this way, be queried from the same table. On the other hand, the number of documents is reduced from 34.4 to 19.5 M entries. In other words, we did not extract a single statement in around 15 M documents. Some database statistics (July 2022) like the number of tuples and size on disk of relevant tables are reported in Table 6. We used Postgres V10 as a relational database implementation. The database (incl. indexes and materialized views) consumed roughly 300GB of disk space in sum.

5.3 Document retrieval

As a reminder of Sect. 3, a narrative query graph consists of fact patterns following simple RDF-style basic graph patterns. Our discovery system automatically translates these narrative query graphs into a structured query language: They are translated into SQL statements for querying the underlying relational database. A single fact pattern requires a selection of the extraction table with suitable conditions to check the entities and the interaction. Multiple fact patterns require self-joining of the extraction table and adding document conditions in the where clause, i.e., the facts matched against the query must be extracted from the same document. In practice, joining the predication table with itself was not fast enough when many rows were selected to answer a fact pattern.

That is why we computed an inverted index. The inverted index mapped subject-predicate-object tuples to a denormalized attribute: This attribute then stored a document ID plus the predication IDs in a JSON format, e.g., the document IDs *1* and the predication IDs *2* and *3*. The predication IDs were required to explain matches to users, i.e., which sentence and why the sentence matched the fact pattern. For subjects and objects we used two attributes: the corresponding entity ID and entity type. The type was helpful to accelerate queries with variables that search for a specific entity type (e.g., drug or disease). We created indexes for the subject, predicate, and object attributes. The inverted index had 34 million tuples and consumed 14 GB of disk space (incl. indexes). Having that index, a fact pattern required only a single selection on it. We developed an in-memory and hash-based matching algorithm that quickly combines the results.

Another issue to think about were ontological subclass relations between entities. For example, querying for treatments of *Diabetes Mellitus* would require to also search for the subclasses *Diabetes Mellitus Type 1* and *Diabetes Mellitus Type 2*. Query rewriting was necessary to compute complete results for queries that involve entities with subclasses [21]. Therefore, we utilized the Medical Subject Headings (MeSH) Ontology and the Anatomical Therapeutic Chemical Classification System (ATC). ATC was used to support querying for classes of drugs. We rewrote queries that include entities with subclasses to also query for these

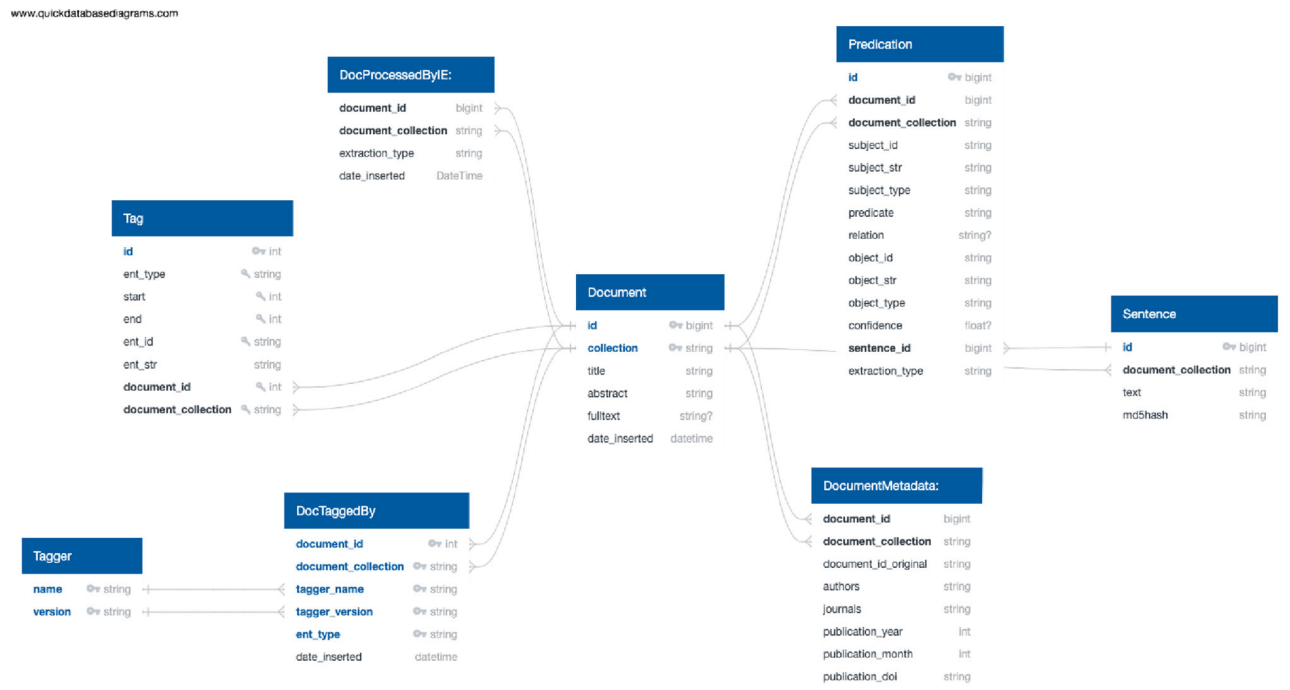


Fig. 2 Database schema overview: document is the central table storing titles and abstracts. Tables on the left side store information about the entity linking. Tables on the right side store information about the extraction process. The document metadata stores information for the service

Table 6 Database statistics (July 2022) of our underlying relational database. We report the consumed disk space of relevant database tables (size reflects the pure data while size* includes indexes as well)

Table name	#Tuples	Size	Size*
Document	34.4 M	32 GB	35 GB
Document metadata	34.4 M	8.1 GB	10 GB
Metadata service	19.5 M	6.6 GB	7.2 GB
Tag	524.2 M	44 GB	91 GB
Predication	394 M	61 GB	95 GB
Sentence	67.3 M	17 GB	19 GB

subclasses. If an entity was also a superclass, then we also searched for all subclasses. We rewrote the SQL statement precisely in the following way: Instead of searching for a single entity, we searched with an *IN* expression. We allowed all subclasses plus the given entity.

In brief, the query translation works as follows:

1. The user inputs a string through the query builder in the form of a list of subject–predicate–object tuples.
2. We translate each subject and object to a set of corresponding entities, i.e., all entities that have the given term (subject/object) as one of its synonyms.
3. We expand each entity by all subclasses, i.e., we apply the *subclass* function to each entity. The intermediate query representation is now a list of fact patterns. A fact pattern is a triple consisting of a set of entities as the subject, a predicate, and a set of entities as the object.
4. We translate each fact pattern into a SQL statement:

If a subject/object is only a single entity, we directly add a simple comparison in the WHERE clause. If a subject/object is a set of entities, we add an *IN* statement to check whether the entity is in that set of entities. To accelerate the translation, we maintained an in-memory index mapping terms to a set of entities, including all of their subclasses if applicable.

Due to the long-standing development of databases, querying our index was performed very quickly by suitable indexes. Besides, we implemented some optimization strategies to accelerate the query processing, e.g., matching fact patterns with concrete entities first and fact patterns with variable nodes afterward. We remark on our system's query performance in our evaluation.

5.3.1 Remarks

Why did we not build upon graph databases? We thought about utilizing graph databases for the query processing. Our main motivation to stick to relational databases was for simplicity reasons: On the one hand, we were familiar with relational architectures. On the other hand, we had to store data about the documents and the extraction information. This way we can identify new documents that must be processed, update suitable tables and indexes, and so on. In

addition, the service had to return document and provenance information to the user, i.e., titles, journal, author, and why documents match a user query. For simplicity, we decided to store all data in a single database and only maintain a single one. Moreover, the overall query performance was sufficient for us. For future work, analyzing graph databases like the RDF database Virtuoso could be of interest for our discovery system.

5.4 Architecture

Our service was realized as a web service split into two components: a backend for the query processing and a frontend for user inputs. We implemented the backend as a REST service based on the Django framework in Python. The frontend was implemented by utilizing the Bootstrap and the jQuery Framework. The data exchange between both sides was realized with JSON.

Transferring the results between the backend and frontend turned out to be a challenge. Search engines typically only transfer parts of the result lists. If users move to the next result page, these page results are transferred to the frontend. The problem for us was that we did not have typical result lists in every case. For instance, results for searches with variables had to be aggregated on the backend side before transmission. In brief, a *simple* result list in our system might be composed of several nested lists, i.e., documents that share the same variable substitution. Thus, implementing a *lazy loading* for the next pages was challenging. This feature would have required a complex caching architecture in the backend, i.e., store the result lists and allow the frontend to load specific parts dynamically. An alternative would have been to recompute the same query with corresponding page/list positions. Both options were not suitable for us because they would have consumed too many resources in the backend.

That is why we decided to transfer the complete result object once between the backend and frontend. The JSON contained the basic structure (simple list or nested list), all document information (ID, title, authors, journals, etc.), and provenance information. For provenance information we only transferred IDs of the predication table, i.e., we dynamically loaded provenance information from the backend if the user asked for it. The frontend then dynamically visualizes parts of the results and allows the user to jump in the lists. Depending on the number of results and the usage of variables, the result size can vary between a few KB up to several MB. Very large queries like *Drug treats Disease* would even require to transfer of a few hundred MBs. But the vast majority of queries that contain at least a single entity (and not only variables) require at maximum a few MBs. The Django framework supports sending the data in a compressed format if the browser supports it. We enabled that option to decrease the transmission size. However, we are aware that the transmission size is an issue in practice.

5.5 User interface

In the following, we present a user interface resulting from joint efforts by the University Library, the Institute for Information Systems, and two pharmaceutical domain experts who gave us helpful feedback and recommendations. In contrast to graph query interfaces for SPARQL queries, we wanted to create a user interface that is easy to use and does not require to learn an additional query language. Furthermore, we supported the user with a query builder and suitable result visualization on the frontend side. In an early prototype phase, we tested different user interfaces to formulate narrative query graphs, namely

1. A simple text field,
2. A structured query builder, and
3. A graph designer tool.

We found that our users preferred the structured query builder, which allows them to formulate a query by building a list of fact patterns. For each fact pattern, the users had to enter the query's subject and object. The service assists the user by suggesting about three million terms (entity names plus synonyms). Then, they could select an interaction between both in a predefined selection. Variable nodes could be formulated, e.g., by writing *?X(Drug)* or just entering the entity-type like *Drug* in the subject or object field.

When users start their search, the service sends the query to the backend and visualizes the returned results. The returned results are sorted by their corresponding publication date in descending order. The service represents documents by a document ID (e.g., PubMedID), a title, a link to the digital library entry, metadata (authors, journal, etc.), and provenance information. Provenance includes the sentence from which the matching fact was extracted. We highlight the linked entities (subject and object) and their interaction (text term plus mapping to the interaction vocabulary). Provenance may be helpful for users to understand why a document is a match. If a query contains multiple fact patterns, we attach a list of matched sentences in the visualization. Visualizing document lists is comparable to traditional search engines, but handling queries with variable nodes requires novel interfaces. In the next subsection, we will discuss such visualizations for queries including variable nodes. A screenshot of the running system is shown in Fig. 3

5.6 Retrieval with variable nodes

Variable nodes in narrative query graphs may be restricted to specific entity types like *Disease*. We also allowed a

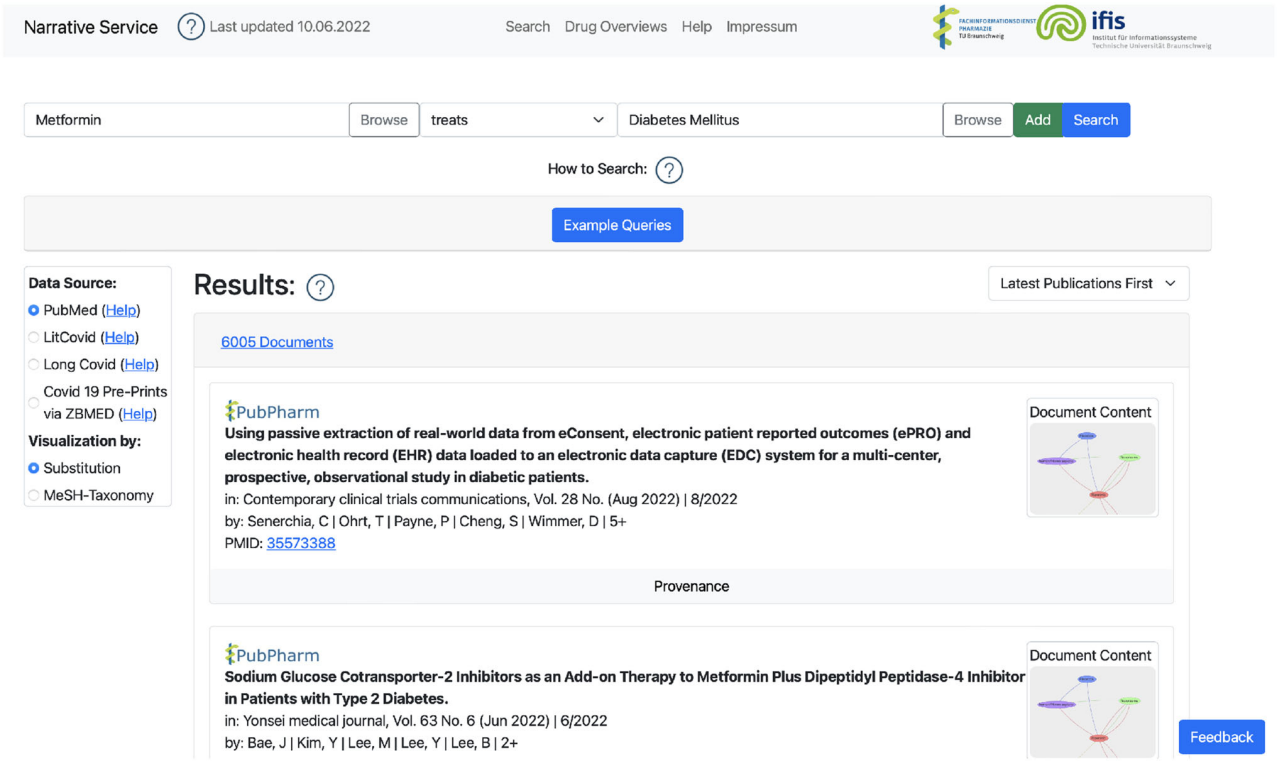


Fig. 3 A screenshot of our user interface: The query builder is shown on the top. Users can formulate their queries by adding more patterns and then start their search. On the left side, several filter options are shown. In the center/bottom, the result list is visualized. Each result is represented by metadata and a Provenance button to explain the match

general type *All* to support querying for arbitrary entities. For example, a user might formulate the query (*Simvastatin*, *treats*, *?X(Disease)*). Several document graphs might match the query with different variable substitutions for $?X$. A document d_1 with the substitution $\mu_1(?X) = hypercholestorelemia$ as well as a document d_2 with $\mu_2(?X) = hyperlipidemia$ might be proper matches to the query. How should we handle and present these substitutions to the users? Discussions with domain experts led to the conclusion that aggregating documents by their substitution seems most promising. Further, we present two strategies to visualize these document result groups in a user interface: *substitution-centric* and *hierarchical visualization*. A general overview of both visualizations is shown in Fig. 4. We implemented the aggregation and ranking on the backend side: The frontend sends the selected visualization to the backend. The backend then calculates the required data representation and sends it to the frontend. The frontend finally visualizes the computed representation.

Substitution-centric visualization. Given a query with a variable node, the first strategy is to aggregate by similar variable substitutions. We retrieve a list of documents with corresponding variable substitutions from the respective document graphs. Different substitutions represent different groups of documents, e.g., one group of documents might cover the treatment of *hypercholestorelemia* while the other group might deal with *hypertriglyceridemia*. When computing the results, an in-memory hash map is created that maps each variable substitution to a set of document ids. These groups are sorted in descending order by the number of documents in each group. Note that a document may have multiple substitutions, and hence, may appear in several groups. Hence, variable substitutions shared by many documents appear at the top of the list by default. Since the lists may become very long, we divided them into pages so that the user can jump to less frequent parts of the result list. In addition, the users may also sort the groups in ascending order (rare substitutions first). Our system visualizes a document group as a collapsible list item. A user's click can uncollapse the list item to show all contained documents. Provenance information is used to explain why a document matches the query, i.e., the system displays the sentences in which a query's pattern was matched. Provenance may be especially helpful when working with variable nodes.

Hierarchical visualization. Entities are arranged in taxonomies in many domains. Here, diseases, dosage forms, and methods are linked to MeSH (Medical Subject Heading) descriptors arranged in the MeSH taxonomy. The hierar-

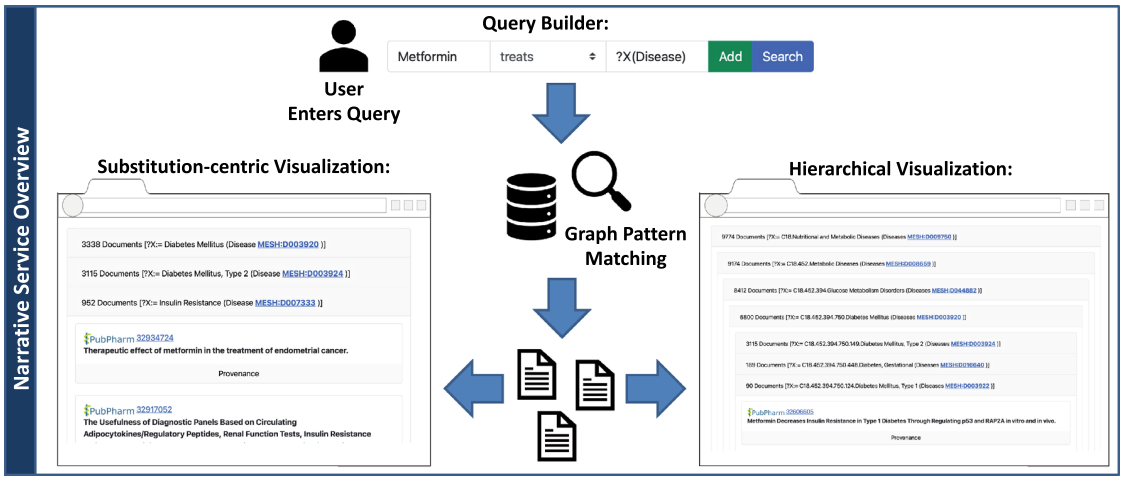


Fig. 4 A schematic overview of our service implementation. A query builder helps the users to formulate their information needs. If the narrative query involves variable nodes, the results can be visualized in a substitution-centric visualization (left side) or in a hierarchical visualization (right side)

chical visualization aims at showing document results in a hierarchical structure. For example, *hypercholestorelemia* and *hypertriglyceridemia* share the same superclass in MeSH, namely *hyperlipidemias*. All documents describing a treatment of *hypercholestorelemia* as well as *hypertriglyceridemia* are also matches to *hyperlipidemias*. On the backend side, we implemented an algorithm that works as follows:

1. Aggregate all documents by their variable substitution. Note that a document may have multiple substitutions, and hence, may appear in several groups.
2. Create an empty MeSH-tree structure.
3. Attach a set of documents to the corresponding tree position, i.e., the entity's position in that tree.
4. Forward the number of documents to all predecessor nodes to update their document count.
5. Prune all nodes that do not have documents attached in their node or all successor nodes to bypass the need to show the whole MeSH taxonomy. Our service visualizes this hierarchical structure by several nested collapsible lists, e.g., *hyperlipidemias* forms a collapsible list. If a user's click uncollapses this list, then the subclasses of *hyperlipidemias* are shown as collapsible lists as well. In this way, users can walk through the tree structure till they find document entries. These document entries are visualized in the same way as in our user interface when no variables are used.

6 Retrieval evaluation

The following evaluations of our prototype are based on an older version (January 2021). In contrast to the content of the current version, which covers the complete Medline collection and COVID-19 pre-prints, the older version was focused on pharmaceutical users. Therefore, we selected a PubMed Medline subset that includes drug and excipient annotations. We annotated the whole Medline collection with our entity linking component, yielding 302 million annotations. Around six million documents included a drug or excipient annotation. Performing the extraction and cleaning workflow on around six million documents yielded nearly 270 million different extractions. Hence, the prototype version in January 2021 included about six million documents. In the following evaluation, we will thus call it prototype because we refer to the version of January 2021. The differences to the current system version were: (1) The content was smaller (not the complete NLM Medline and no pre-prints), (2) the entity vocabularies were older (older versions of MeSH and ChEMBL, and the entity types method, lab method and vaccine were missing), and (3) missing improvements in the user interface (improved document visualization, faster rendering, and faster loading).

Subsequently, we analyze our retrieval prototype concerning two research questions: *Do narrative query graphs offer a precise search for literature? And, do variable nodes provide useful entity-centric overviews of literature?* We performed three evaluations to answer these questions:

1. Two pharmaceutical experts created test sets to quantify the retrieval quality (100 abstracts and 50 full-text papers). Both experts are highly experienced in pharmaceutical literature search.
2. We performed interviews with eight pharmaceutical experts who search for literature in their daily research. Each expert was interviewed twice: Before testing our

prototype to understand their information need and introducing our prototype. After testing our prototype, to collect feedback on a qualitative level, i.e., how they estimate our prototype's usefulness.
3. Finally, all eight experts were asked to fill out a questionnaire. The central findings are reported in this paper.

6.1 Retrieval evaluation

After having consulted the pharmaceutical experts, we decided to focus on the following typical information needs in the biomedical domain:

I1: Drug-Disease treatments (*treats*) play a central role in the mediation of diseases.
I2: Drugs might decrease the effect of other drugs and diseases (*decreases*).
I3: Drug treatments might increase the expression of some substance or disease (*induces*).
I4: Drug-Gene inhibitions (*inhibits*), i.e., drugs disturb the proper enzyme production of a gene.
I5: Gene-Drug metabolisms (*metabolizes*), i.e., gene-produced enzymes metabolize the drug's level by decreasing the drug's concentration in an organism.

Narrative query graphs can specify the exact interactions a user is looking for. For each information need (I1-5), we built narrative query graphs with well-known entities from the pharmaceutical domain:

Q1: *Metformin treats Diabetes Mellitus (I1)*,
Q2: *Simvastatin decreases Cholesterol (I2)*,
Q3: *Simvastatin induces Rhabdomyolysis (I3)*,
Q4: *Metformin inhibits mtor (I4)*,
Q5: *CYP3A4 metabolizes Simvastatin AND Erythromycin inhibits CYP3A4 (I4/5)*, and
Q6: *CYP3A4 metabolizes Simvastatin AND Amiodarone inhibits CYP3A4 (I4/5)*.

For our evaluation, we wanted to measure our system's precision and recall. The recall was of interest here because we already knew that information extraction (PathIE) could only extract statements between entities if mentioned in the same sentence. That is why we used the entities for each query to search for document candidates on PubMed, e.g., for Q1 we used *metformin diabetes mellitus* as the PubMed query. We kept only documents that were processed in our pipeline. Then, we took a random sample of 25 documents for each query. The experts manually read and annotated these sample documents' abstracts concerning their information needs (true hits/false hits). Besides, we retrieved 50 full-text documents from PubMed Central (PMC) for a combined and very specialized information need (Q5 and Q6). The experts made their decision for PubMed documents by considering titles and abstracts, and for PMC documents, the full texts. We decided to select 25 as the sample size for each query because we had to obtain a manageable set of documents for our manual expert evaluation (in sum 100 abstracts and 50 full texts had to be evaluated). Subsequently, we considered these documents as ground truth to estimate the retrieval quality (precision, recall, and F1). Note that we did consider any ranking for the subsequent evaluation because matching narrative query graphs against document graphs is a binary decision: Either the information is contained or not. Ranking the results of such a ranking would require novel methods that were out of scope for this evaluation. However, we compared our retrieval to two baselines, (1) queries on PubMed and (2) queries on PubMed with suitable MeSH headings and subheadings.

PubMed MeSH baseline PubMed provides so-called MeSH terms for documents to assist users in their search process. MeSH is an expert-designed vocabulary comprising various biomedical concepts (around 26K different headings). These MeSH terms are assigned to PubMed documents by human annotators who carefully read a document and select suitable headings. Prime examples for these headings are annotated entities such as drugs, diseases, etc., and concepts such as study types, therapy types, and many more. In addition to headings, MeSH supports about 76 subheadings to precisely annotate how a MeSH descriptor is used within the document's context. An example document might contain the subheading *drug therapy* attached to *simvastatin*. Hence, a human annotator decided that *simvastatin* is used in drug therapy within the document's context. The National Library of Medicine (NLM) recommends subheadings for entity interactions such as treatments and adverse effects. In cooperation with our experts who read the NLM recommendations, we selected suitable headings and subheadings to precisely query PubMed concerning the respective entity interaction for our queries. We denote this baseline as *MeSH Search*.

Results The corresponding interaction and the retrieval quality (precision, recall, and F1-score) for each query are depicted in Table 7. The sample size and the number of positive hits in the sample (TP) are reported for each query. For instance, the sample size of Q2 was 25, and 16 documents were correct hits with regard to the corresponding information need. The subsequent reported precision, recall and F1 scores are based on the corresponding sample for each document.

The PubMed search was used to construct the ground truth, i.e., was used to retrieve the document lists from which the samples were drawn. That means that the PubMed search achieved a recall of 1.0 in all cases because all samples were subsets of the PubMed search results. The PubMed search yielded a precision of around 0.64 up to 0.76 for abstracts

Table 7 Expert evaluation of retrieval quality for narrative query graphs compared to PubMed and a MeSH-based search on PubMed. Two experts have annotated PubMed samples to estimate whether the information need was answered. Then, precision, recall, and F1-measure are computed for all systems

				PubMed	MeSH search			Narrative QG		
Query	#Hits	#Sample	#TP	Prec.	Prec.	Rec.	F1	Prec.	Rec.	F1
Q1	12.7K	25	19	0.76	0.82	0.47	0.60	**1.00**	0.42	0.59
Q2	5K	25	16	0.64	**0.73**	0.50	0.59	0.66	0.25	0.36
Q3	427	25	17	0.68	0.77	0.59	0.67	**1.00**	0.35	0.52
Q4	726	25	16	0.64	**0.78**	0.44	0.56	0.71	0.31	0.43
Q5	397	25	6	0.24	–	–	–	**1.0**	0.17	0.29
Q6	372	25	5	0.20	–	–	–	**1.0**	0.20	0.33

- denotes no hits

and 0.2 up to 0.24 for full texts. The PubMed MeSH search achieved a moderate precision of about 0.73-−0.82 and a recall of about 0.5 for PubMed titles and abstracts (Q1-Q4). Unfortunately, the relevant MeSH annotations were missing for all true-positive hits for Q5 and Q6 in PMC full texts. Hence, the PubMed MeSH search did not find any hits in PMC for Q5 and Q6. Narrative query graphs (Narrative QG) answered the information need with good precision: Q1 (*treats*) and Q3 (*induces*) were answered with a precision of 1.0 and a corresponding recall of 0.42 (Q1) and 0.47 (Q3). The minimum achieved precision was 0.66, and the recall differed between 0.17 and 0.42. Our prototype could answer Q5 and Q6 on PMC full texts: One correct match was returned for Q5 as well as for Q6, leading to a precision of 1.0.

6.1.1 Comparison to OpenIE

For our prototype, we used PathIE to extract the document graphs. For this comparison, we repeated the extraction on the benchmark documents by utilizing the Stanford CoreNLP OpenIE [25]. We selected the same relation vocabulary and cleaning rules. The results are listed in Table 8. By utilizing OpenIE we could not answer four out of six queries (Q1, Q3, Q5, Q6). A problematic example is the following sentence: *Metformin is the mainstay therapy for type 2 diabetes*. CoreNLP OpenIE extracted the following statement: (Metformin, is, mainstay therapy for type 2 diabetes). First, the object phrase contains more information than just the diabetes disease. Even if we would reduce the phrase to the pure disease *diabetes*, canonicalizing the verb phrase *is* to a *treats* relation would not be possible, simply because for all *is* verb phrases this decision would be wrong. Hence, CoreNLP OpenIE did not yield a suitable *treats* statement here to answer the query. In contrast, PathIE extracted a *treats* statement here because *therapy* was included in the relation vocabulary (a list of special words indicating a relation).

Although we achieved a precision of 1.0 for Q4 and 0.5 for Q2, in both cases the recall was at 0.06. In contrast, PathIE could answer all queries. For Q2 PathIE obtained a higher precision than OpenIE. For Q4 the precision was lower (0.71 instead of 1.0), but the recall was higher (0.31 instead of 0.11). In summary, PathIE was more suitable for our prototype because it could answer more queries and had a better F1 score for all queries.

Table 8 Comparison between CoreNLP OpenIE and PathIE for narrative query graph retrieval (- no hits)

	OpenIE			PathIE		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Q1	–	–	–	1.00	0.42	0.59
Q2	0.50	0.06	0.11	0.66	0.25	0.36
Q3	–	–	–	1.00	0.35	0.52
Q4	1.00	0.06	0.12	0.71	0.31	0.43
Q5	–	–	–	1.0	0.17	0.29
Q6	–	–	–	1.00	0.20	0.33

6.2 User interviews

The retrieval evaluation demonstrated that our system could achieve good precision when searching for specialized information needs. However, the following questions were: How does our prototype work for daily use cases? And, what are the prototype's benefits and limitations in practice? Therefore, we performed two interviews with each of the eight pharmaceutical experts who search for literature in their daily work. All experts had a research background and worked either at a university or university hospital.

First interview We asked the participants to describe their literature search in the first interview. They shared two different scientific workflows that we had analyzed further: (1) Searching for literature in a familiar research area and (2) Searching for a new hypothesis which they might have heard in a talk or read in some paper. We performed think-aloud experiments to understand both scenarios. They shared their screen, showed us at least two different literature searches, and how they found relevant documents answering their information need. For scenario (1), most of them already knew suitable keywords, works, or journals. Hence, they quickly found relevant hits using precise keywords and sorting the results by their publication date. They already had a

good overview of the literature and could hence answer their information need quickly. For scenario (2), they guessed keywords for the given hypothesis. They had to refine their search several times by varying keywords, adding more, or removing some. Then, they scanned titles and abstracts of documents looking for the given hypothesis. We believe that scenario (1) was recall-oriented: They did not want to miss important works. Scenario (2) seemed to be precision-oriented, i.e., they quickly wanted to check whether the hypothesis may be supported by literature. Subsequently, we gave them a short introduction to our prototype. We highlighted two features: The precision-oriented search and the usage of variable nodes to generate entity-centric literature overviews. We closed the first interview and gave them three weeks to use the prototype for their literature searches.

Second interview We asked them to share their thoughts about the prototype: What works well? What does not work well? What could be improved? First, they considered querying with narrative query graphs, especially with variable nodes, different and more complicated than keyword-based searches. Querying with variable nodes by writing *?X(Drug)* as a subject or an object was deemed too cryptic. They suggested that using *Drug*, *Disease*, etc. would be easier. Another point was that they were restricted to a fixed set of subjects and objects (all known entities in our prototype). For example, querying with pharmaceutical methods like *photomicrography* was not supported back then. Next, the interaction vocabulary was not intuitive for them. Sometimes they did not know which interaction would answer their information need. One expert suggested to introduce a hierarchical structure for the interactions, i.e., some general interactions like *interacts* that could be specified into *metabolizes* and *inhibits* if required. On a positive note, they appreciated the prototype's precise search capability. They all agreed that they could find precise results more quickly using our prototype in comparison to other search engines. Besides, they appreciated the provenance information (why the document should be a match) to estimate if a document match answers their information need. They agreed that variable nodes in narrative query graphs offered completely new search possibilities, e.g., *In which dosage forms was Metformin used when treating diabetes?* Such a query could be translated into two fact patterns: (*Metformin*, *administered*, *?X(DosageForm)* and (*Metformin*, *treats*, *Diabetes Mellitus*). The most common administrations are done *orally* or via an *injection*. They agreed that such information might not be available in a specialized database like DrugBank. DrugBank covers different dosage forms for Metformin but not in combination with diabetes treatments. As queries get more complicated and detailed, such information can hardly be gathered in a single database. They stated that the *substitution-centric visualization* helps them to estimate which substitutions are relevant based on the number of supporting documents. Besides, they found the *hierarchical visualization* helpful when querying for diseases, e.g., searching for (*Metformin*, *treats*, *?X(Disease)*). Here, substitutions are shown in an hierarchical representation, e.g., *Metabolism Disorders*, *Glucose Disorders*, *Diabetes Mellitus*, *Diabetes Mellitus Type 1*, etc. They liked this visualization to get a drug's overview of treated disease classes. All of them agreed that searches with variable nodes were helpful to get an entity-structured overview of the literature. Four experts stated that such an overview could help new researchers get better literature overviews in their fields.

6.3 Questionnaire

We asked each domain expert to answer a questionnaire after completing the second interview. The essential findings and results are reported subsequently. First, we asked them to choose between precision and recall when searching for literature. Q1: *To which statement would you rather agree when you search for related work?* The answer options were (rephrased): A1a: *I would rather prefer a complete result list (recall). I do not want to miss anything.* A2a: *I would rather prefer precise results (precision) and accept missing documents*. Six of eight experts preferred recall, and the remaining two preferred precision. We asked a similar question for the second scenario (hypothesis). Again, we let them select between precision and recall (A1a and A1b). Seven of eight preferred precision, and one preferred recall when searching for a hypothesis. Then, we asked Q3: *To which statement would you rather agree for the vast majority of your searches?* Again, seven of eight domain experts preferred precise hits over complete result lists. The remaining one preferred recall. The next block of questions was about individual searching experiences with our prototype (called prototype in the Questionnaire): different statements were rated on a Likert scale ranging from 1 (disagreement) to 5 (full agreement). The results are reported in Table 9. They agreed that the prototype allows to formulate precise questions (4.0 mean rating), and the formulation of questions was understandable (4.0). Besides, provenance information was beneficial for our users (5.0). They could well imagine using our prototype in their literature research (3.9) and searching for a hypothesis (3.4). Still, users were reluctant to actually switch to our prototype for related work searches (2.8). Finally, the result visualization of narrative query graphs with variables was considered helpful (4.5).

6.4 Performance analysis

We measured the performance of our prototype and database on a server, having two Intel Xeon E5-2687W (@3,1 GHz, eight cores, 16 threads), 377 GB of DDR3 main memory, and SDDs as primary storage. Back in January 2021, the

Table 9 Questionnaire results: eight participants were asked to rate the following statements about our prototype on a Likert scale ranging from 1 (disagreement) to 5 (agreement). The mean ratings are reported

Statement about the prototype	Mean
The prototype allows me to formulate precise questions by specifically expressing the interactions between search terms	4.0
The formulation of questions in the prototype is understandable for me	4.0
The displayed text passage from the document (Provenance) is helpful for me to understand why a document matches my search query	5.0
The prototype provides precise results for my questions (I quickly find a relevant match)	3.5
Basically, grouping results is helpful for me when searching for variable nodes	4.5
When searching for related work, I would prefer the prototype to a search using classic search tools (cf. PubPharm, PubMed, etc)	2.8
When searching for or verifying a hypothesis, I would prefer the prototype to a search using classic search tools (cf. PubPharm, PubMed, etc).	3.4
I could imagine using the prototype in my literature research	3.9

preprocessing took around one week for our six million documents (titles and abstracts). We have incrementally improved the performance and can now process the complete Medline collection (34 M documents) in one week. We randomly generated 10 k queries asking for one, two, and three interactions. We measured the query execution time on a single thread (on the January 2021 version). Queries that are not expanded via an ontology took in average 21.9 ms (1-fact) / 52 ms (2-facts) / 51.7 ms (3-facts). Queries that are expanded via an ontology took in average 54.9 ms (1-fact) / 158.9 ms (2-facts) / 158.2 ms (3-facts). However, the query time heavily depends on the interaction (selectivity) and how many subclasses are involved. In summary, our system can retrieve documents within a quick response time for the vast majority of searches.

7 Discussion

In close cooperation with domain experts using the PubMed corpus, our evaluation shows that overall document retrieval can indeed decisively profit from graph-based querying. The expert evaluation demonstrates that our system achieves moderate up to good precision for highly specialized information needs in the pharmaceutical domain. Although the precision is high, our system has only a moderate recall. Moreover, we compared our system to manually curated annotations (MeSH and MeSH subheadings), which are a unique feature of PubMed. Most digital libraries may support keywords and tags for documents but rarely support how these keywords, and primarily, how entities are used within the document's context. Therefore, we developed a document retrieval system with a precision comparable to manual metadata curation but without the need for manual curation of documents.

The user study and questionnaire reveal a strong agreement for our service's usefulness in practice. In summary, the user interface must be intuitive to support querying with narrative query graphs. Further enhancements are necessary to explain the interaction vocabulary to the user. We appreciate the idea of hierarchical interactions, i.e., showing a few basic interactions that can be specified for more specialized needs. Especially the search with variable nodes in detailed narrative query graphs offers a new access path to the literature. The questionnaire showed that seven of eight experts agreed that the vast majority of their searches are precision-oriented. Next, they agreed that they prefer our service over established search engines for precision-oriented searches. The verification of hypotheses seems to be a possible application because precise hits are preferred here. We believe that our service should not replace classical search engines because there are many recall-oriented tasks like related work searches. The recall will always be a problem by design when building upon error-prone natural language processing techniques and restricting extractions to sentence levels. Although the results seem promising, there are still problems to be solved in the future, e.g., we can still improve the extraction and the user interface.

7.1 Technical challenges

We faced five major technical challenges when realizing narrative query graph retrieval:

Retrieval with graphs Graph-based retrieval of literature requires representing texts differently. We decided to extract statements from text, compute an inverted index, and then compute queries against that index. An alternative could be a retrieval with the latest language models that may match query graphs on-the-fly against texts.

Data storage and query processing The processing of queries requires performing an expensive graph-pattern matching. Here we built upon relational databases to store all data within one place. But alternatives like performing queries on a graph database and retrieving Provenance from a different source could be relevant.

Query formulation Formulating information needs as query graphs was unfamiliar and thus challenging for users. Easy-to-use interfaces must be developed and integrated here. An extension could be to formulate natural language questions that are automatically translated to query graphs.

Result list handling Transferring result lists between backend and frontend can be similar to keyword-based retrieval systems. But if variables came into play and result lists became nested, a new way of handling that lists was required. Either we must recompute the query for certain list parts or design a suitable caching and streaming architecture.

Querying with variables Searches with variables required novel methods to transfer and visualize the result lists: On the one hand, the visualization must face a large amount of data in real-time. On the other hand, interfaces should be fast and responsive.

Our discovery system tackled all challenges by finding solutions that worked in practice and delivered a suitable quality.

7.2 Generalizability

Knowledge in the biomedical domain is often entity-centric, e.g., clinical studies involving certain target groups, drug testing, treatments and therapies, method investigations, and much more. Existing thesauri and distinguishing relations between certain entities were essential for realizing access via narrative query graphs. The generalizability of this research is in this way limited to entity-centric domains. In political sciences, information needs may be based upon some school of thought. For example, when searching for statements of that school, special keywords and framing are essential to formulate the actual query. Breaking down such information need to a *simple* entity-interaction pattern does not seem possible.

On the one hand, we have already seen methods suitable for our extraction workflow, and hence, our discovery system would not work well for political sciences if suitable vocabularies were missing [19]. Canonicalizing different predicates to precise relations here is even more challenging. In biomedicine, a drug and a disease might roughly stand in two relations: Either the drug treats the disease or induces the disease. When thinking about possible relations between persons, we are likely to face a high number of possible ones. But on the other hand, although having these restrictions, we have elaborated on the benefits of narrative information access for political sciences [20]. For example, listing people involved in a certain decision, or structuring the literature into action categories, e.g., tackling climate change, could be answered by realizing a similar access path to that domain.

8 Outlook

In addition to steady improvements in the user interface, entity vocabularies, and content updates, we give an outlook on the latest developments of our service.

8.1 Feedback

This work's qualitative and quantitative evaluations show that users can benefit from narrative query graphs in practice. However, we continue the evaluation of our service. For instance, two preliminary evaluations are described in [20]: In joint work with the specialized information services for political sciences (Pollux), we analyzed how well the service can assist research questions in political sciences. In cooperation with ZB MED (infrastructure and research center for information and data in the life sciences) and Robert Koch-Institute in Germany (leading public health institute in Germany), we evaluate how well the service is suitable to answer COVID-9-related questions.

For our daily users, we integrated three feedback options into our system: (1) Users can automatically create a screenshot of our system, mark something in that and write a short text. Then the data is sent automatically to our service. (2) Users can rate visualized substitution groups when searching with variables. Users who explore substitution lists can directly rate if this group is sensible concerning the query. (3) Users can rate Provenance information, i.e., whether the extraction is suitable to answer the query. The options are shown in Fig. 5. All this feedback is stored in our service, and we will further use it to improve the system and extraction methods.

8.2 Concept selection

In our study, we learned that selecting the correct concept (entity/class or variable) can sometimes be challenging. On the one hand, users might not know the correct term for a given entity, e.g., users searched for *diabetes* instead of the correct entity term *Diabetes Mellitus*. On the other hand, users did not know which overviews could be generated, i.e., which variable types we allowed. To deal with this problem, we introduced the so-called Concept Selection View. A screenshot is shown in Fig. 6. Here users can enter a term, and a list of matching concepts (entities/classes/variables) is shown. In addition, we integrated a list of allowed variable types and classes from MeSH and ATC. This view extends the autocompletion function by showing a tree-based visualization of concepts, i.e., we utilized ontologies like MeSH and ATC (a drug taxonomy) to show the different concepts

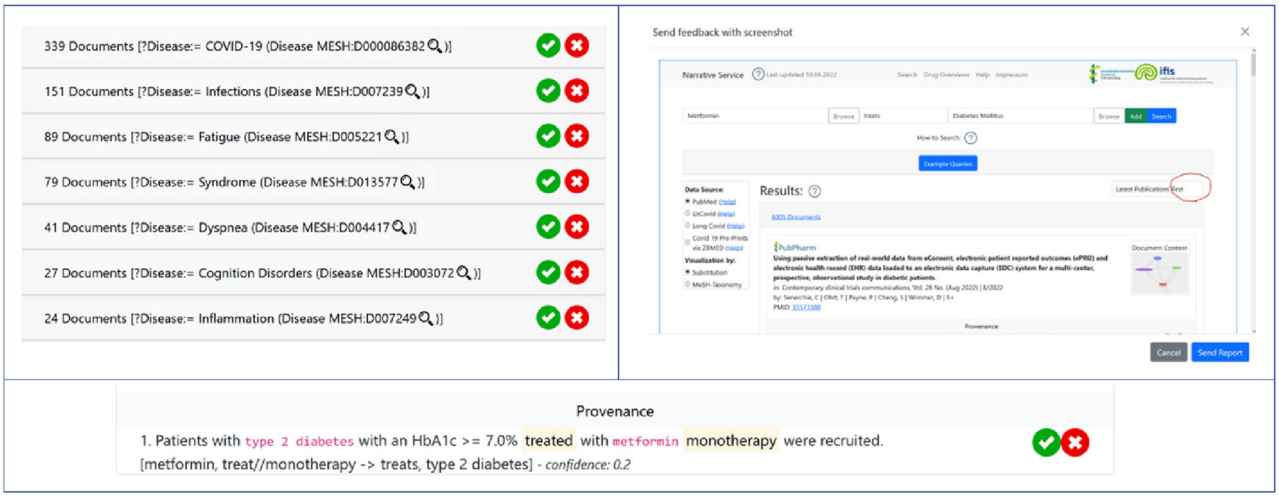


Fig. 5 Feedback options of our service: on the left upper corner, users can rate substitution groups when searching with variables. On the right upper corner, users can create a screenshot, mark something in that, write a short text, and send it to us. At the bottom, users can rate Provenance entries (if the extraction was suitable)

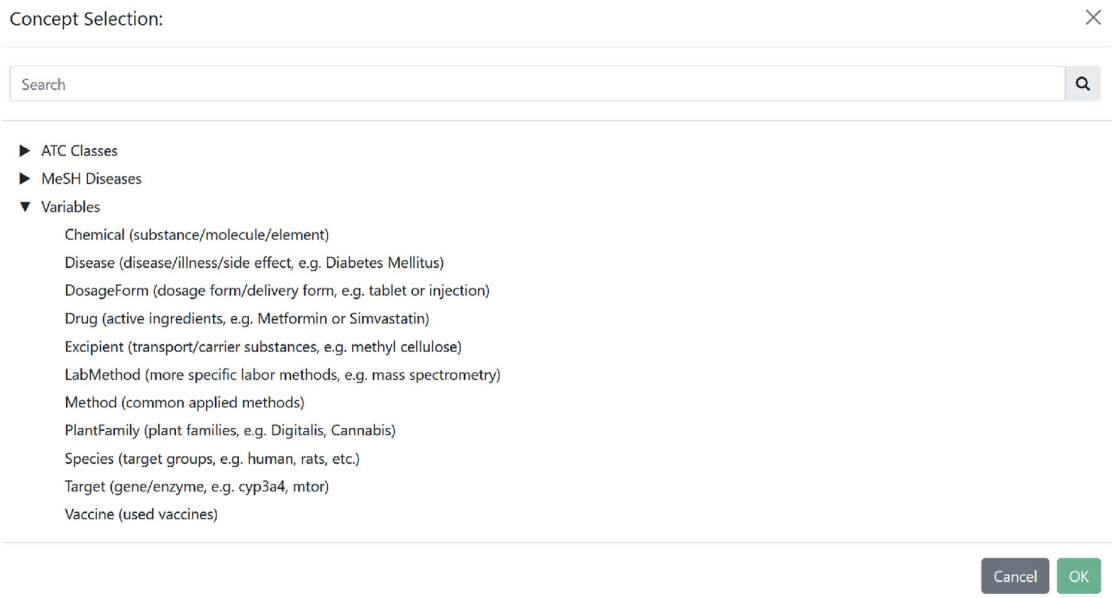


Fig. 6 Concept selection: this view allows users to precisely select their concept (entity/class) or a typed variable (variable of type drug, etc.) in searches. The view has a search field, so the tree-based visualization can be searched in real-time

and their superclass/subclass relationships. If a user selects a concept here, the concept will be copied to the query builder. Users can access the concept selection by clicking on the *Browse* button either in the subject or object of the query builder.

8.3 Document graphs

Our service already allowed the visualization of Provenance information, i.e., the service explains why a document should match a query graph. However, we found it to be useful to allow users to explore our actual document graph representation as well. Therefore, we integrated a Document Graph View. A screenshot is shown in Fig. 7. This view has two components: On the left side, the document's text plus metadata (authors, journals, etc.) is shown. Here we highlight detected entities in the title and abstract in a certain color (the color denotes the entity type). Users have the option to select or deselect certain entity types to focus the visualization on their needs. On the right side, the actual document graph is visu-

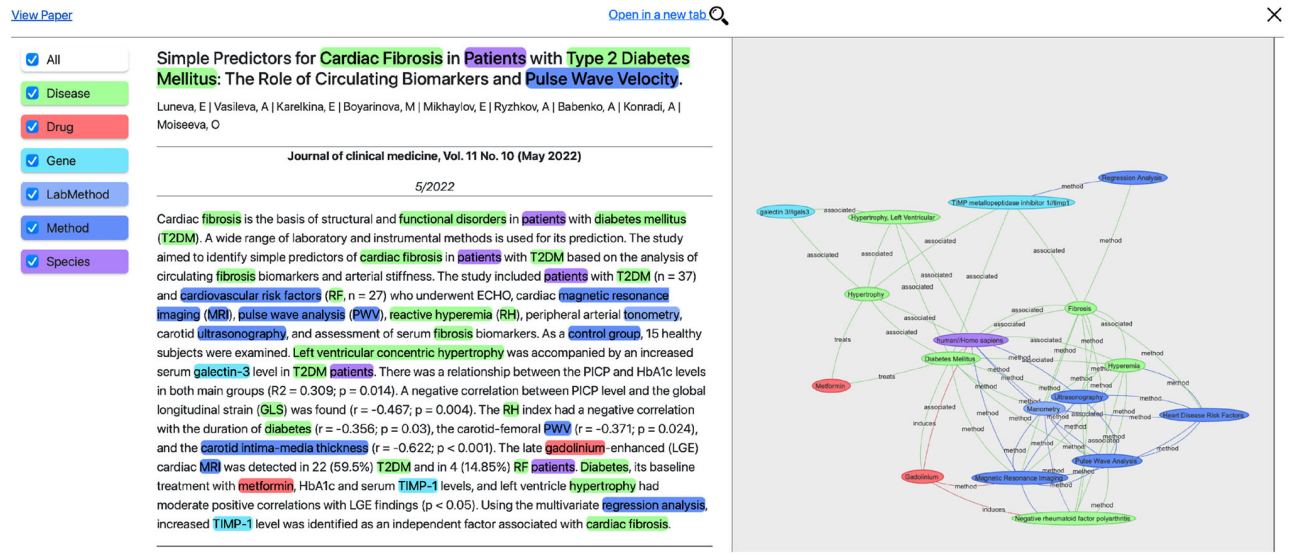


Fig. 7 Document graph visualization: a document's abstract and identified entities are highlighted on the left side. On the right side, interactions between entities (statements) are visualized as a directed-edge labeled and colored graph. Different colors depict different entity types (drugs, diseases, etc)

alized as a directed-edge labeled and colored graph. Colors again denote the entity types. The graph is interactive, and users may move nodes or edges. If two entities are connected via several predicates, we only visualize a single edge and concatenate the predicate labels, e.g., *associated and treats*.

8.4 Drug overviews

Our latest extension is the so-called *Drug Overviews*. A screenshot is shown in Fig. 8. Users have to enter a drug name, and the corresponding overview is generated for them. Therefore, we combine information from our service as well as from curated and specialized databases. On the one hand, we show curated information about the drug like the molecular mass, pKA values, etc. To retrieve the curated information, we utilized the official API of the ChEMBL database [27]. On the other hand, a set of pre-defined narrative query graphs is used to show extractions from the literature. In cooperation with domain experts again, we created these query graphs for different purposes: Showing known indications (treatments) of the drug, showing how the drug is administered (tablet, injection, etc.), the interacting targets (enzymes/gene systems) and more. Then the corresponding results, i.e., the entity groups, are shown in list views. Each entry consists of two components: the entity's name and the number of supporting documents for the given relationship between the searched drug and this entity. Users who click on an entity will be forwarded to our retrieval service, and the corresponding relationship is searched automatically. Another thing to mention are indications: Here, we combine extractions from the literature with information about clinical phases from ChEMBL [27]. Suppose a drug-disease-indication is verified via a clinical trial. In that case, the corresponding phase of the trial is visualized as a roman letter in its entry. Users who click on the trial phase will be forwarded to the corresponding ChEMBL entry.

The difference with existing curated databases is that we can generate these overviews even for the latest drugs. And moreover, we can show associations that may have been reported in the literature but have not been curated in a database. For instance, a drug administration as a nanoparticle might not have worked out in practice. It likely will not appear in a curated database but is shown in our overview. In summary, these overviews allow thus to quickly retrieve information about a drug, even if the drug has not been researched thoroughly.

9 Conclusion

Entity-based information access catering even for complex information needs is a central necessity in today's scientific knowledge discovery. But while structured information sources such as knowledge graphs offer *high query expressiveness* by graph-based query languages, scientific document retrieval is severely lagging behind. The reason is that graph-based query languages allow to describe the desired characteristics of and interactions between entities in sufficient detail. In contrast, document retrieval is usually limited to simple keyword queries. Yet unlike knowledge graphs, scientific document collections offer *contextualized knowledge*, where entities, their specific characteristics, and

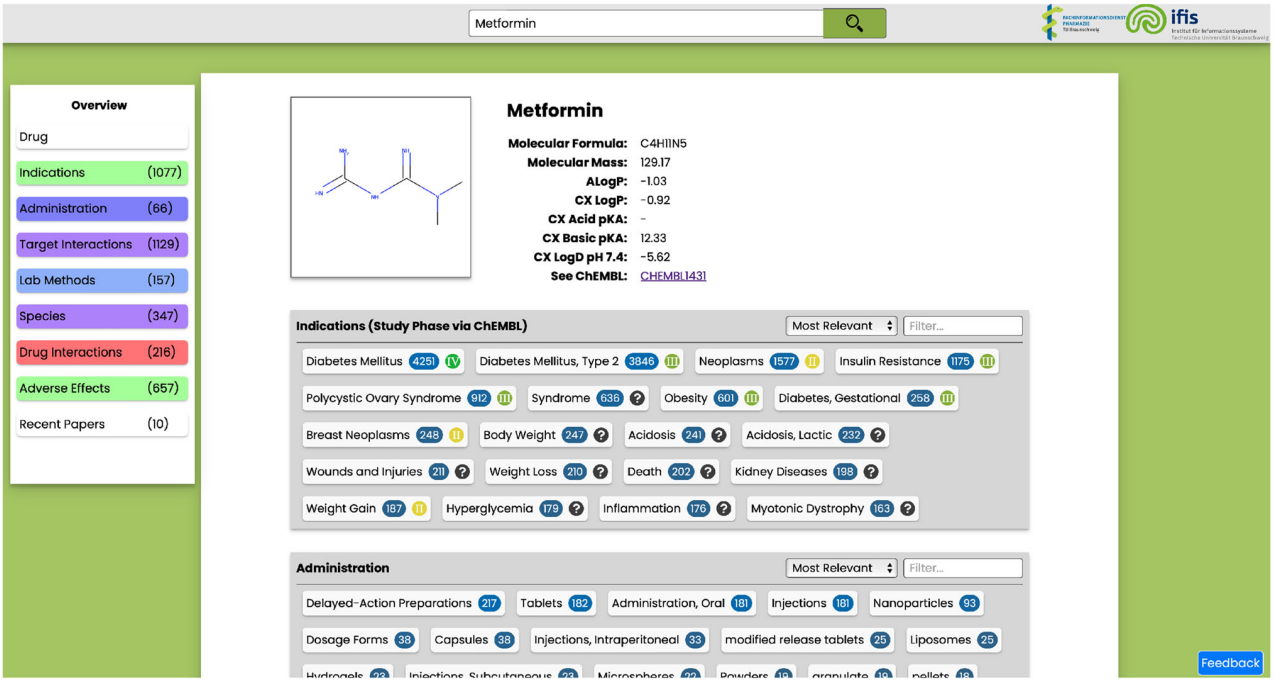


Fig. 8 Drug overviews: users enter a drug name, and then an overview of the drug is generated. Therefore we combine information from our service as well as from curated and specialized databases. A user's click on an entity will then invoke a search in our discovery system

their interactions are connected as part of a coherent argumentation and thus offer a clear advantage [14, 15]. The research presented in this paper offers a novel workflow to bridge the worlds of structured and unstructured scientific information by performing graph-based querying against scientific document collections. Implementing such an access path to a digital library comes with costs for designing extraction workflows and maintaining the actual discovery system. However, nearly unsupervised extraction workflows might be a compromise here: They bypass training data for the extraction but suffer in quality and require suitable vocabularies. If a digital library decides to go that way, novel applications such as the query graph retrieval system, searches with variables, graph visualizations of documents, or overviews of certain entities (here drugs) are not too far-fetched. But as our current workflow is clearly precision-oriented, we plan to improve the recall without having to broaden the scope of queries in future work.

Supplementary information The code of the extraction toolbox can found in our GitHub repository: https://github.com/HermannKroll/KGExtractionToolbox. An archived version of our toolbox can be found in the Software Heritage project: https://archive.softwareheritage.org/swh:1:dir:67c17339a5c800ddb50cb36bda598fb96a200856.

Acknowledgements Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): PubPharm - the Specialized Information Service for Pharmacy (Gepris 267140244).

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Azad, H.K., Deepak, A.: Query expansion techniques for information retrieval: a survey. Inf. Process. Manag. **56**(5), 1698–1735 (2019). https://doi.org/10.1016/j.ipm.2019.05.009
2. Betts, C., Power, J., Ammar, W.: GrapAL: connecting the dots in scientific literature. In: Proceedings of the 57th annual meeting of the association for computational linguistics: system demonstrations. association for computational linguistics, Florence, Italy, pp 147–152, (2019) https://doi.org/10.18653/v1/P19-3025
3. Chen, Q.: An object-oriented database system for efficient information retrieval applications. PhD thesis, (1992) http://hdl.handle.net/10919/27976

4. Croft, W., Parenty, T.J.: A comparison of a network structure and a database system used for document retrieval. Inf. Syst. **10**(4), 377–390 (1985). https://doi.org/10.1016/0306-4379(85)90042-0
5. Croft, W.B., Wolf, R., Thompson, R.: A network organization used for document retrieval. In: proceedings of the 6th annual international acm sigir conference on research and development in information retrieval. association for computing machinery, New York, NY, USA, SIGIR '83, p 178-188, (1983) https://doi.org/10.1145/511793.511820
6. Dietz, L., Kotov, A., Meij, E.: Utilizing knowledge graphs for text-centric information retrieval. In: The 41st international ACM SIGIR conference on research & development in information retrieval. Association for computing machinery, New York, NY, USA, SIGIR '18, p 1387-1390, (2018) https://doi.org/10.1145/3209978.3210187
7. Dogan, R.I., Leaman, R., Lu, Z.: NCBI disease corpus: a resource for disease name recognition and concept normalization. J. Biomed. Inf. **47**, 1–10 (2014). https://doi.org/10.1016/j.jbi.2013.12.006
8. Färber, M.: The microsoft academic knowledge graph: A linked data source with 8 billion triples of scholarly data. In: The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II, Lecture Notes in Computer Science, vol 11779. Springer, pp 113–129, (2019) https://doi.org/10.1007/978-3-030-30796-7_8
9. France, R.K.: Effective, efficient retrieval in a network of digital information objects. PhD thesis, (2001) http://hdl.handle.net/10919/29754
10. Herskovic, J.R., Tanaka, L.Y., Hersh, W., et al.: A day in the life of pubmed: analysis of a typical day's query log. J. Am. Med. Inf. Assoc. **14**(2), 212–220 (2007). https://doi.org/10.1197/jamia.M2191
11. Jaradeh, M.Y., Oelen, A., Farfar, K.E., et al. Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge. In: proceedings of the 10th international conference on knowledge capture, K-CAP 2019, Marina Del Rey, CA, USA, November 19-21, 2019. ACM, pp 243–246, (2019) https://doi.org/10.1145/3360901.3364435
12. Kadry, A., Dietz, L.: open relation extraction for support passage retrieval: merit and open issues. In: proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval. Association for computing machinery, New York, NY, USA, SIGIR '17, p 1149-1152, (2017) https://doi.org/10.1145/3077136.3080744
13. Kolluru, K., Adlakha, V., Aggarwal, S., et al. OpenIE6: iterative grid labeling and coordination analysis for open information extraction. In: Proc. of the 2020 conf. on empirical methods in natural language processing (EMNLP). ACL, pp 3748–3761, (2020) https://doi.org/10.18653/v1/2020.emnlp-main.306
14. Kroll, H., Kalo, J.C., Nagel, D., et al.: Context-compatible information fusion for scientific knowledge graphs. In: Digital Libraries for Open Knowledge, pp. 33–47. Springer (2020)
15. Kroll, H., Nagel, D., Balke, W.T.: Modeling Narrative Structures in Logical Overlays on Top of Knowledge Repositories. In: Dev, T. (ed.) Conceptual Modeling, pp. 250–260. Springer (2020)
16. Kroll, H., Nagel, D., Kunz, M., et al. Demonstrating narrative bindings: linking discourses to knowledge repositories. In: fourth workshop on narrative extraction from texts, Text2Story@ECIR2021, CEUR Workshop Proceedings, vol 2860. CEUR-WS.org, pp 57–63, (2021a) http://ceur-ws.org/Vol-2860/paper7.pdf
17. Kroll, H., Pirklbauer, J., Balke, W.: A toolbox for the nearly-unsupervised construction of digital library knowledge graphs. In: ACM/IEEE joint conference on digital libraries, JCDL 2021, Champaign, IL, USA, September 27-30, 2021. IEEE, pp 21–30, (2021b) https://doi.org/10.1109/JCDL52503.2021.00014
18. Kroll, H., Pirklbauer, J., Kalo, J., et al. Narrative query graphs for entity-interaction-aware document retrieval. In: Towards open and trustworthy digital societies—23rd international conference on Asia-pacific digital libraries, ICADL 2021, Virtual Event, December 1-3, 2021, Proceedings, Lecture Notes in Computer Science, vol 13133. Springer, pp 80–95, (2021c) https://doi.org/10.1007/978-3-030-91669-5_7
19. Kroll, H., Pirklbauer, J., Plötzky, F., et al. A library perspective on nearly-unsupervised information extraction workflows in digital libraries. In: proceedings of the 22nd ACM/IEEE joint conference on digital libraries. Association for computing machinery, New York, NY, USA, JCDL '22, (2022a) https://doi.org/10.1145/3529372.3530924
20. Kroll, H., Plötzky, F., Pirklbauer, J., et al. What a Publication Tells You-Benefits of Narrative Information Access in Digital Libraries. In: Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries. Association for Computing Machinery, New York, NY, USA, JCDL '22, (2022b) https://doi.org/10.1145/3529372.3530928
21. Krötzsch, M., Rudolph, S.: Is your database system a semantic web reasoner? KI-Künstliche Intelligenz **30**(2), 169–176 (2016). https://doi.org/10.1007/s13218-015-0412-x
22. Langnickel, L., Baum, R., Darms, J., et al. COVID-19 preVIEW: semantic search to explore COVID-19 research preprints. In: public health and informatics. IOS Press, Amsterdam, the Netherlands, p 78–82, (2021a) https://doi.org/10.3233/SHTI210124
23. Langnickel, L., Darms, J., Baum, R., et al.: preVIEW: from a fast prototype towards a sustainable semantic search system for central access to COVID-19 preprints. J. EAHIL **17**(3), 8–14 (2021)
24. Leaman, R., Lu, Z.: TaggerOne: joint named entity recognition and normalization with semi-Markov Models. Bioinformatics **32**(18), 2839–2846 (2016). https://doi.org/10.1093/bioinformatics/btw343
25. Manning, C.D., Surdeanu, M., Bauer, J., et al. The stanford CoreNLP natural language processing toolkit. In: proceedings of the 52nd annual meeting of the association for computational linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, system demonstrations. The association for computer linguistics, pp 55–60, (2014) https://doi.org/10.3115/v1/p14-5010
26. Manola, F., Miller, E., McBride, B., et al. RDF primer. W3C recommendation 10(1-107):6 (2004)
27. Mendez, D., Gaulton, A., Bento, A.P., et al.: ChEMBL: towards direct deposition of bioassay data. Nucleic Acids Res. **47**(D1), D930–D940 (2018). https://doi.org/10.1093/nar/gky1075
28. Mohan, S., Fiorini, N., Kim, S., et al. A fast deep learning model for textual relevance in biomedical information retrieval. In: Proceedings of the 2018 world wide web conference. International world wide web conferences steering committee, Republic and Canton of Geneva, CHE, WWW '18, p 77-86, (2018) https://doi.org/10.1145/3178876.3186049
29. Nguyen, D.B., Abujabal, A., Tran, N.K., et al.: Query-driven on-the-fly knowledge base construction. Proc. VLDB Endow **11**(1), 66–79 (2017)
30. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of SPARQL. ACM Trans. Database Syst. (2009). https://doi.org/10.1145/1567274.1567278
31. Priem, J., Piwowar, H., Orr, R.: Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. (2022) https://doi.org/10.48550/ARXIV.2205.01833
32. Ratner, A., Bach, S.H., Ehrenberg, H.R., et al.: Snorkel: rapid training data creation with weak supervision. Proc. VLDB Endow **11**(3), 269–282 (2017)
33. Raviv, H., Kurland, O., Carmel, D.: Document retrieval using entity-based language models. In: Proceedings of the 39th international acm sigir conference on research and development in information retrieval. association for computing machinery, New

York, NY, USA, SIGIR '16, p 65-74, (2016) https://doi.org/10.1145/2911451.2911508
34. Shin, J., Wu, S., Wang, F., et al.: Incremental knowledge base construction using deepdive. Proc. VLDB Endow **8**(11), 1310–1321 (2015)
35. Spitz, A., Gertz, M.: Terms over LOAD: Leveraging named entities for cross-document extraction and summarization of events. In: proceedings of the 39th international acm sigir conference on research and development in information retrieval. Association for computing machinery, New York, NY, USA, SIGIR '16, p 503-512, (2016) https://doi.org/10.1145/2911451.2911529
36. Vazirgiannis, M., Malliaros, F.D., Nikolentzos, G.: GraphRep: boosting text mining, NLP and information retrieval with graphs. In: proceedings of the 27th ACM international conference on information and knowledge management. Association for computing machinery, New York, NY, USA, CIKM '18, p 2295-2296, (2018) https://doi.org/10.1145/3269206.3274273
37. Vrandecic, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014). https://doi.org/10.1145/2629489
38. Weaver, M.T.: Implementing an intelligent information retrieval system: the CODER system, version 1.0. Master's thesis, (1988) http://hdl.handle.net/10919/44097
39. Wei, C.H., Kao, H.Y., Lu, Z.: PubTator: a web-based text mining tool for assisting biocuration. Nucleic Acids Res. **41**(W1), W518–W522 (2013). https://doi.org/10.1093/nar/gkt441
40. Wei, C.H., Kao, H.Y., Lu, Z.: GNormPlus: an integrative approach for tagging genes, gene families, and protein domains. BioMed. Res. Int. **918**, 710 (2015a). https://doi.org/10.1155/2015/918710
41. Wei, C.H., Peng, Y., Leaman, R., et al. Overview of the BioCreative V chemical disease relation (CDR) task. In: proceedings of the fifth biocreative challenge evaluation workshop (2015b)
42. Wei, C.H., Allot, A., Leaman, R., et al.: PubTator central: automated concept annotation for biomedical full text articles. Nucleic Acids Res. **47**(W1), W587–W593 (2019). https://doi.org/10.1093/nar/gkz389
43. Xiong, C., Power, R., Callan, J.: Explicit semantic ranking for academic search via knowledge graph embedding. In: proceedings of the 26th international conference on world wide web. international world wide web conferences steering committee, Republic and Canton of Geneva, CHE, WWW '17, p 1271-1279, (2017) https://doi.org/10.1145/3038912.3052558
44. Zhang, Y., Chen, Q., Yang, Z., et al.: BioWordVec, improving biomedical word embeddings with subword information and MeSH. Sci. Data **6**(1), 52 (2019). https://doi.org/10.1038/s41597-019-0055-0
45. Zhao, S., Su, C., Sboner, A., et al. GRAPHENE: a precise biomedical literature retrieval engine with graph augmented deep learning and external knowledge empowerment. In: proceedings of the 28th ACM international conference on information and knowledge management. Association for computing machinery, New York, NY, USA, CIKM '19, p 149-158, (2019) https://doi.org/10.1145/3357384.3358038

B.7. IJDL 2023b: A detailed library perspective on nearly unsupervised information extraction workflows in digital libraries

IJDL'23b

Hermann Kroll, Jan Pirklbauer, Florian Plötzky, and Wolf-Tilo Balke. "A detailed library perspective on nearly unsupervised information extraction workflows in digital libraries". International Journal on Digital Libraries (IJDL) 2023. DOI: https://doi.org/10.1007/s00799-023-00368-z

International Journal on Digital Libraries
https://doi.org/10.1007/s00799-023-00368-z

A detailed library perspective on nearly unsupervised information extraction workflows in digital libraries

Hermann Kroll[1] · **Jan Pirklbauer**[1] · **Florian Plötzky**[1] · **Wolf-Tilo Balke**[1]

Received: 30 October 2022 / Revised: 9 May 2023 / Accepted: 19 May 2023

Abstract
Information extraction can support novel and effective access paths for digital libraries. Nevertheless, designing reliable extraction workflows can be cost-intensive in practice. On the one hand, suitable extraction methods rely on domain-specific training data. On the other hand, unsupervised and open extraction methods usually produce not-canonicalized extraction results. This paper is an extension of our original work and tackles the question of how digital libraries can handle such extractions and whether their quality is sufficient in practice. We focus on unsupervised extraction workflows by analyzing them in case studies in the domains of encyclopedias (Wikipedia), Pharmacy, and Political Sciences. As an extension, we analyze the extractions in more detail, verify our findings on a second extraction method, discuss another canonicalizing method, and give an outlook on how non-English texts can be handled. Therefore, we report on opportunities and limitations. Finally, we discuss best practices for unsupervised extraction workflows.

Keywords Open information extraction · Extraction workflows · Digital libraries

1 Introduction

This paper is an extended version of our previous work [17] focusing on nearly unsupervised information extraction workflows in digital libraries. Extracting structured information from textual digital library collections enables novel access paths, e.g., answering complex queries over knowledge bases [2, 30], providing structured overviews about the latest literature [9], or discovering new knowledge [8].

However, utilizing information extraction (IE) tools in digital libraries is usually quite cost-intensive, which hampers the implementation in practice. On the one hand, extraction methods usually rely on supervision, i.e., ten thousands of examples must be given for training suitable extraction models [35]. On the other hand, utilizing the latest natural language processing (NLP) tools in productive pipelines requires high expertise and computational resources.

In addition to supervised IE, Open IE methods (OpenIE) have been developed to work out-of-the-box without additional domain-specific training [11, 23]. But why aren't they used broadly in digital library applications? The reason is that OpenIE generates non-canonicalized (not normalized) results, i.e., several extractions describing the same piece of information may be structured in completely different ways (synonymous relations, paraphrased information, etc.). But such non-canonicalized results are generally not helpful in practice, because a clear relation and entity semantics like in supervised extraction workflows is vital for information management and query processing. Since the lack of clear semantics has been recognized as a major issue, cleaning and canonicalization methods have been investigated to better handle such extractions [31]. Still are they ready for application in digital libraries?

In this paper, case studies are used to find out how suitable nearly unsupervised methods are to design reliable extraction workflows. In particular, we analyze extraction and cleaning methods from the perspective of a digital library

✉ Hermann Kroll
kroll@ifis.cs.tu-bs.de

✉ Jan Pirklbauer
j.pirklbauer@tu-bs.de

✉ Florian Plötzky
ploetzky@ifis.cs.tu-bs.de

✉ Wolf-Tilo Balke
balke@ifis.cs.tu-bs.de

[1] Institute for Information Systems, TU Braunschweig, Mühlenpfordtstr. 23, Braunschweig 38106, Lower Saxony, Germany

Published online: 13 June 2023

by assessing the required expertise, domain knowledge, computational costs and result quality.

Therefore, we selected our toolbox for a nearly unsupervised extraction from text published in JCDL 2021 [15]. The toolbox contains interfaces to the latest named entity recognition (NER) and open information extraction methods. In addition, it includes cleaning and canonicalization methods to handle noisy extractions by utilizing domain-specific information. Our corresponding paper [15] advertises the toolbox to considerably decrease the need for supervision and to be transferable across domains; nevertheless, it comes with several limitations:

1. Although we did report on the extraction quality (good precision, low recall), we did **not** report on the **costs of applying the toolbox**, i.e., how much expertise and computational costs are required for a reliable workflow.
2. We applied the toolbox only in the biomedical domain, which lessens the **generalizability of our findings**.
3. Moreover, we did **not** report what is **technically and conceptually missing** in such extraction workflows.
4. We focused on **English** texts and did not analyze workflows for **non-English** texts yet.

In this paper, we address the previous issues by analyzing the toolbox application in three distinct real-world settings from a library perspective: 1. We extracted knowledge about scientists from the online encyclopedia Wikipedia (controlled vocabularies, descriptive writing). 2. We applied the toolbox to the pharmaceutical domain (controlled vocabularies, entity-centric knowledge) in cooperation with the specialized information service for Pharmacy (www.pubpharm.de). 3. We applied the toolbox in Political Sciences (open vocabulary, topic/event-centric knowledge) in cooperation with the specialized information service for Political Sciences [29] (www.pollux-fid.de). For Pharmacy and Political Sciences, we recruited associated domain experts for expertise in the evaluation. We performed these three case studies to answer the following questions:

1. *How much expertise and effort is required to apply nearly unsupervised extractions across different domains?*
2. *How generalizable are these state-of-the-art extraction methods and particularly, how useful are the extraction results?*
3. *What is missing toward a comprehensive information extraction from texts, e.g., for retaining the original information?*

In addition to those questions, we discuss how digital libraries may handle non-English texts with our toolbox. This paper is an extended version of our previous article [17]: For our extension, we (1) give more insights and details for each case study in Sect. 4, (2) investigate the complexity of extracted noun phrases in Sect. 4.4, (3) apply and analyze a second OpenIE tool, namely CoreNLP OpenIE, to generalize our findings in Sect. 4.5, (4) have a close look on an unsupervised canonicalization method for verb phrases in Sect. 5, and (5) dive into machine translation to apply the toolbox on non-English texts, at the example of German in Sect. 6. For a comparison of old and new hardware, we also measured the runtimes on our latest server from 2021 in Sect. 7.5.1.

2 Related work

The main goal of information extraction (IE) is the extraction of structured information from unstructured or semi-structured information such as texts, tables, figures, and more [11, 22, 23, 35]. In the following, we give an overview of challenges and research trends in IE from texts.

Current Trends. Modern IE research mainly focuses on improving the extraction accuracy, which is typically measured on benchmarks [3, 11]. Indeed, previous evaluations have shown that IE methods already produce good results, but the research is still ongoing [3, 5, 11, 15, 26]. Primarily driven by the development of language models like BERT [5], IE has made a step forward.

However, these systems rely on supervised learning and thus need large-scale training data that cannot be reliably transferred across domains. In brief, although supervised methods are up to the job with reasonable quality, their practical application comes at high costs. The expenses for supervision lead to the design of zero-shot, semi-supervised, and distant supervised extraction methods (see [35] for a good overview).

Open Information Extraction. Instead of designing extraction systems for each domain, methods like unsupervised information extraction (OpenIE) are proposed to change the game [26]. OpenIE aims to extract knowledge from texts without knowing the entity and relation domains a-priori [26, 35]. While supervised (closed) methods focus on domain-specific and relevant relations and concepts, open methods are more flexible and may be applied across domains [26, 35].

Canonicalization of OpenIE. Vashishth proposed CESI to canonicalize OpenIE extractions by clustering noun and verb phases with the help of side information [31]. However, CESI was analyzed for short phrases that refer to precise entities. In addition, studies have shown that OpenIE methods may struggle to handle scientific texts well because sentences are often long and domain-specific vocabulary terms are used [7]. While research in both directions (open and closed) is still ongoing, some works bridge the gap between both worlds: Kruiper et al. propose the task of Semi-Open Relation extraction [20], i.e., they use domain-specific information

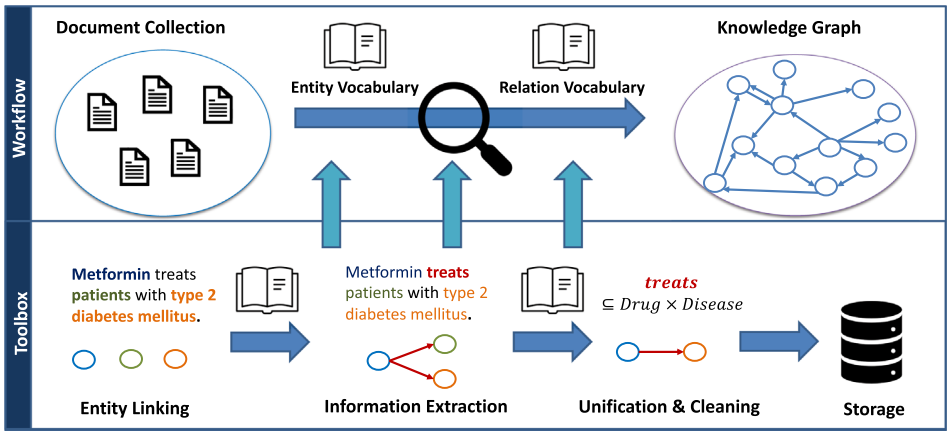


Fig. 1 The Toolbox's Systematic Overview: Entity linking detects concepts/entities, and information extraction extracts relations between them. Then, the output will be cleaned and loaded into a structured repository [15]

to filter irrelevant open information extractions. Similarly, we showed that domain-specific filtering of OpenIE outputs could yield helpful results [15].

Information Extraction in Digital Libraries. Digital libraries are interested in practical IE workflows to allow novel applications; see this tutorial at JCDL2016 [36]. IE can allow literature-based discovery workflows, which have been studied on DBpedia [30]. The extraction of entities and relations is therefore challenging. That is why modern approaches build upon language models and supervision for a reliable extraction [28]. These language models require extensive computational resources for training and application [5, 21]. Good examples for IE are DBpedia [2], which was harvested from Wikipedia infoboxes or the SemMedDB, which is a collection of biomedical statements harvested from PubMed [10, 37]. Hristovski et al. have used the SemMedDB to perform knowledge discovery [8]. Nevertheless, the construction of SemMedDB required biomedical experiences to define hand-written rules for the extraction. In contrast to the previous works, our work focused on nearly unsupervised extraction workflows that do not rely on training data for the extraction phase.

3 Study objectives

In the following, we briefly summarize the nearly unsupervised extraction toolbox, raise research questions for our case studies, and explain why we selected the three domains here. A systematic toolbox overview is shown in Fig. 1. Our main objective here is to analyze unsupervised extraction workflows from a digital library perspective.

3.1 Overview of the toolbox

The extraction toolbox covers three common IE areas: entity detection, information extraction, and canonicalization. We shared our toolbox as open-source software and made it publicly available.[1,2] We focus on this toolbox because it proposed an eased and nearly unsupervised extraction workflow by integrating the latest unsupervised extraction plus suitable cleaning methods.

Nearly Unsupervised. We call an information extraction workflow nearly unsupervised if two conditions hold: 1. No training data are required to train or fine-tune an entity detection or information extraction model. In other words, entities and statements are extracted without supervision. And 2. entity information and a relation vocabulary are used to clean not-normalized extraction outputs, e.g., by filtering OpenIE noun phrases via detected entities or canonicalizing synonymous verb phrases to precise relations. In contrast to pure unsupervised workflows, our workflow requires the design of an entity and relation vocabulary to obtain precise relation semantics, e.g., a treats relation between drugs and diseases.

Entity Detection. The toolbox integrates interfaces to one of the latest NER tools, Stanford Stanza [27]. Stanford Stanza is a pre-trained neural model that can be applied without adapting it to a certain domain. Stanza is capable of detecting 18 general-purpose entity types like *persons*, *organizations*, *countries*, and *dates* in texts; see [27] for a complete overview. In addition, the toolbox supports the linking of custom entity vocabularies via a dictionary-based lookup method. The entity linker supports an abbreviation resolution and handling of short homonymous terms (link if the entity is mentioned with a longer mention in the text).

Information Extraction. The toolbox integrates implements interfaces to OpenIE methods, Stanford CoreNLP [23] and OpenIE6 [11]. Besides, the toolbox includes a self-developed path-based extraction method named PathIE. PathIE extracts statements between entities in a sentence if connected in the grammatical structure via verb phrases or custom keywords (e.g., treatment, inhibition, award, and

[1] https://github.com/HermannKroll/KGExtractionToolbox.

[2] https://archive.softwareheritage.org/swh:1:dir:5b575ac043e2bd61999250564a16a220c88ee5c9.

member of) that can be specified beforehand. The OpenIE methods work entirely without entity information, whereas the PathIE requires entity annotations as starting points (as an input).

Cleaning and Canonicalization. OpenIE and PathIE may produce non-helpful and non-canonicalized (not-normalized) outputs, i.e., synonymous noun and verb phrases that describe the same information. The toolbox supports canonicalizing and filtering such outputs automatically. First, extracted noun phrases can be filtered by entity annotations, i.e., only noun phrases that include relevant entities are kept. Here, three different filters are supported to filter noun phrases: exact (noun phrase matches an entity), partial (noun phrase partially includes an entity), and no filter (keep original noun phrase). We will introduce the subject filter as a new option in our case studies. For convenience, the subject filter requires the extracted subject noun phrase to be a detected entity. And it keeps the object noun phrase as it is. As a recent example, consider the sentence: ***Queen of England*** passed away in **2022** after a long reign in **Balmoral Castle**. Assume that we detected the bold text spans as entities. For the following extraction (Queen of England; passed away; in 2022 after a long reign in Balmoral Castle), filtering will then yield:

No Filter:	Keep the extraction as it is.
Partial Filter:	(Queen of England; passed away; Balmoral Castle) and (Queen of England; passed away; 2022).
Exact Filter:	will not return anything because the object consists of more than the detected entity.
Subject Filter:	(Queen of England; passed away; after a long reign in Balmoral Castle)

Second, an iterative cleaning algorithm is integrated that can canonicalize synonymous verb phrases to precise relations, e.g., *birthplace* or *place of birth* to *born in*. Therefore, users can export statistics about the non-canonicalized verb phrases and build a so-called relation vocabulary. Each entry of this vocabulary is a relation consisting of a name and a set of synonyms. The toolbox utilizes this vocabulary to automatically map synonymous verb phrases to precise relations. Word embeddings are supported in the canonicalization procedure to bypass an exhausting editing of the relation vocabulary. The central idea of word embeddings is that words with a similar context appear close in the vector space [25]. The word embedding is then used to automatically map a new verb phrase to the closest match (most similar) in the vocabulary. Relation type constraints can then be used to filter the extractions further, i.e., a relation type constraint describes which entity types are allowed as subjects and objects. For example, *born in* can be defined as a relation between *persons* and *countries*. Other extractions that hurt these constraints are then removed. We already reported on some challenges of OpenIE extractions, especially on handling noun phrases [14]. In contrast to our previous works, this work analyzes the complete workflow in three domains from a library perspective.

3.2 Study goals

The study goals concern three concrete areas of study: 1. application costs, 2. generalizability, and 3. limitations for a comprehensive IE. However, answering these questions on a purely quantitative level is challenging, e.g., how can the costs be measured? That is why we report our findings as a mixture of quantitative measures (e.g., time spent and runtimes) and qualitative observations (what works well and what does not). We define evaluation criteria for all of the three aspects in the following.

3.2.1 Application costs

We understand everything necessary to implement a workflow with the toolbox as *application costs*. We estimate the application costs in terms of

Data Preparation:	transforming data into toolbox formats (e.g., JSON), working with toolbox outputs (TSV/JSON)
Implementation:	computational costs (runtime and space), scalability, executed steps, effort to choose parameters, encountered issues
Domain Knowledge:	entity and relation vocabulary design, required knowledge for canonicalization

3.2.2 Generalizability

In short, how well are the proposed methods generalizable across domains, and how useful are the results?

Extraction quality:	benchmarks (precision and recall), observations, extraction limitations
Usefulness:	relevance of statements (e.g., non-obvious statements), domain insights, helpfulness for domain experts, usefulness in applications

Information, originally connected in coherent written texts, might be broken into not helpful pieces in the end. For a good example, consider a drug-disease treatment: Here context information like the dose or treatment duration, which could give more information about the statement's validity [13], might get lost. We refer to such information as the **context** of statements, e.g., the surrounding scope in which a

Table 1 The number of documents and sentences is reported for each collection and sample

Collection	Size	Sample	
		#Documents	*#Sentences*
English wikipedia	6.3 M	2373	74.5k
PubMed	33 M	10k	87.1k
Political sciences	1.7 M	10k	66.9k

statement is valid. We already discussed why context information is essential when extracting statements; see [13, 18]. In addition, the connection between statements might get lost, too, e.g., an assumption might lead to a conclusion. We call this the **coherence of statements**. They are crucial for real-world applications, but have they being considered yet?

3.2.3 On context and coherence

Contexts affect the validity of statements, and coherence describes how statements belong together. We evaluate the following criteria:

Contexts: relevance of contexts, which kind of information requires context, how does the context affect the validity of extracted statements, what must be done to retain context

Coherence: complex information that is broken into pieces, which kind of information is broken down, what are the subsequent problems with such a decomposition

3.3 Case study selection

We applied the toolbox in three different domains to generalize the findings in this paper. Here we focused on natural language texts written in the English language. We describe the domains and their characteristics in the following. Table 1 provides statistics about the used data and samples.

3.3.1 Wikipedia

A prime example of an encyclopedia is the free and collaborative Wikipedia. Encyclopedic texts should be written in descriptive and objective language, i.e., wording and framing should not play any role. Wikipedia captures knowledge about certain items (persons, locations, events, etc.), in our understanding, entities. Here controlled ontologies about entities and relations are available; see Wikidata [32] as a good example. However, Wikipedia texts also tend to include very long and complex sentences. For this case study, we focus on knowledge about famous fictional and non-fictional scientists (about 2.4k scientists with an English Wikipedia article and Wikidata entry). This case study was selected because sentences are written objectively, and controlled vocabularies are available for usage.

3.3.2 Pharmaceutical domain

The pharmaceutical domain focuses on entity-centric knowledge, i.e., statements about entities such as drugs, diseases, treatments, and side effects. Many vocabularies and ontologies are curated to describe relevant biomedical entities, e.g., the National Library of Medicine (NLM) maintains the so-called Medical Subject Headings (MeSH).[3] These headings are entities with descriptions, ontological relations (subclasses), and suitable synonyms. In this paper, we select a subset of the most comprehensive biomedical collection, the NLM Medline collection.[4] Medline includes around 35 million publications with metadata (title, abstracts, keywords, authors, publication information, etc.). The specialized information service for Pharmacy was interested in statements about drugs. Therefore, we applied the entity linking step to all Medline abstracts (Dec. 2021) and randomly picked a subset of 10k abstracts that included at least one drug mention.

3.3.3 Political sciences

The Political Sciences domain encompasses a diverse range of content, e.g., publications about topics and events, debates, news, and political analyses. Because of its diversity, this domain does not provide extensive curated vocabularies and ontologies. We argue that entity subsets of knowledge bases like Wikidata [32] or DBpedia [2] might be good starting points to derive some entity vocabularies regarding persons, events, locations, and more. Still, Wikidata and DBpedia are built as general-purpose knowledge bases. They are thus not focused on Political Sciences (in contrast to MeSH for the biomedical domain). Nevertheless, they might be helpful to analyze texts in Political Sciences, which is why we analyze them for a practical application here. In addition, descriptions of entities in Political Sciences tend to be subjective, i.e., they depend on different viewpoints and schools of thought. For example, the accession of Crimea to Russia in 2014 was a highly discussed topic, whether this event could be seen as peaceful secession or as an annexation. In contrast to objective and entity-centric statements in biomedicine, Political Sciences are far more based on the wording and framing of certain events. This case study analyzes how far IE methods can bring structure into these texts and where these methods fail. The specialized information service for Political

[3] https://meshb.nlm.nih.gov/search.

[4] https://www.nlm.nih.gov/medline/medline_overview.html.

Table 2 Corpus and entity detection statistics for our case studies

	Sentences		Entity detection	
	#Sent.	*#with2E*	*#NER*	*#EL*
Wikipedia	74.5k	50.3k	155.0k	113.2k
Pharmacy	87.1k	47.4k	–	232.5k
Pol. Sci.	66.9k	17.6k	80.0k	3.7k

We report the number of sentences, sentences with at least two entities mentions, Stanza NER, and entity linking annotations

Sciences (Pollux) provided us with around three million publications (around 1.3 million English abstracts). Our case study is based on a random sample of 10k abstracts selected from the English subset. In addition, domain experts manually selected five abstracts due to their focus on the diverse topics of the EU, philosophy, international relations, and parliamentarism (Tables 2 and 3).

4 Case studies

For our case studies, we developed scripts, produced intermediate results, and implemented some improvements to the toolbox. The details, used data, and produced results of every case study can be found in our evaluation scripts on GitHub (see the Toolbox GitHub Repository). We included a Readme file[5] to document the following case studies. All our experiments and time measurements were performed on our server, having two Intel Xeon E5-2687W (@3,1GHz, eight cores, 16 threads), 377GB of DDR3 main memory, one Nvidia 1080 TI GTX GPU, and SSDs as storage.

For the first part of this section, we used OpenIE6 to perform the OpenIE extractions because it was the latest OpenIE tool available in the toolbox. To better generalize those findings, we subsequently analyzed the produced noun phrases in detail and compare the results to the CoreNLP OpenIE tool; see Sect. 4.4 and Sect. 4.5.

4.1 Wikipedia case study

This first case study was based on 2.3k English Wikipedia full-text articles about scientists. The conversion of Wikipedia articles was simple: We downloaded the available English Wikipedia dump (Dec. 2021), used the WikiExtractor [1] to retrieve plain texts, and filtered these texts by our scientist's criteria (title must be about a scientist of Wikidata). Next, we developed a Python script to transform the plain texts into a JSON format for the toolbox. The data transformations took half a person-day.

[5] https://github.com/HermannKroll/KGExtractionToolbox/blob/main/README_CASE_STUDIES.md.

4.1.1 Entity linking

In this case study, we focused on statements about scientists, such as works, scientific organizations, and degrees. Therefore, we performed entity linking to identify these concepts and use them to filter the extraction outputs. We derived corresponding entity vocabularies from Wikidata by utilizing the official SPARQL endpoint. We retrieved vocabularies by asking for English labels and alternative labels for the following entity types: *Academia of Sciences*, *Awards*, *Countries*, *Doctoral Degrees*, *Religions* and *Irreligions*, *Scientists*, *Professional Societies*, *Scientific Societies* and *Universities*.

This query returned rows including the entity id, the entity name, and a;-separated list of English alternative labels for the corresponding entity. We adjusted the SPARQL queries to directly download the vocabularies as TSV files in the toolbox format. A first look over this entity vocabulary revealed some misleading labels (e.g., the, he, she, and, or), which we removed. Our final vocabulary included 27,864 distinct entities and 68,668 distinct terms.

We applied the dictionary-based entity linker utilizing our vocabulary to the articles. The linker yielded many erroneously linked entities because of very ambiguous labels in the dictionary, e.g., the mentions *doctor*, *atom*, and *observation* were linked to fictional characters which are scientists regarding the Wikidata ontology. Next, synonyms like *Einstein* were erroneously linked when talking about his family or talking about the term *Einstein* in the sense of *genius*. The linker also ignored pronouns completely, i.e., no co-reference resolution was applied. Especially in Wikipedia articles, pronouns are often used. In addition, we executed the NER tool Stanford Stanza to recognize general-purpose entity types like dates or organizations. A closer look at Stanza's results revealed that short entity names were too ambiguous. That is why we removed all detected entities with less than five characters. This step yielded 155k Stanza NER mentions and 113.2k dictionary-based entity links.

4.1.2 Information extraction

OpenIE6. We applied the OpenIE6 method and the entity filter methods (no filter, partial, exact). We obtained 117.1k (no filter), 317.8k (partial), and 2.9k (exact) extractions. Note that statements can be duplicated for the partial filter if multiple entities are included within the same noun phrase. We exported 100 results for each filter randomly and analyzed them. In the following, we report on some examples of good and bad extractions.

Some interesting results about Albert Einstein are listed in Table 4. OpenIE6 produced correct and helpful extractions when sentences were short and simple (no nested structure, no relative clauses, etc.). When sentences became longer, the tool yielded short subjects but long and complex objects,

Table 3 OpenIE6 extraction and filtering statistics: We report the percentage of complex subjects and objects, the number of extractions computed by the different entity filters (no, partial, exact, subject), and PathIE (number of extractions)

	OpenIE6						PathIE
	C. Subjs. (%)	*C. Objs.* (%)	*#No EF*	*#Part. EF*	*#Exact EF*	*#Subj. EF*	*#Extr.*
Wikipedia	16.2	74.5	177.1k	317.8k	2.9k	80.9 k	1.3 M
Pharmacy	37.8	72.1	207.6k	88.0k	291	15.1k[6]	430.8k
Pol. Sci.	32.0	74.3	147.2k	28.6k	128	7.3k	–

Table 4 OpenIE6 example extractions from the Wikipedia article of Albert Einstein. On the left, the corresponding entity filter is shown (subject, partial and exact). Subject[S], predicate[P] and object[O] are highlighted respectively

Wikipedia	Exact	E1.1	In 1933, while **Einstein**[S] **(Person)** *was visiting*[P] **the United States**[O] **(Country)**, [...]
		E1.2	On 30 April 1905, Einstein completed his thesis, with **Alfred Kleiner**[S] **(Person)**, *[be] Professor*[P] of **Experimental Physics**[O] **(ORG)**, serving as "pro-forma" advisor.
	Partial	E2.1	In a German-language letter to **philosopher**[O] **(Profession)** Eric Gutkind, dated 3 January 1954, **Einstein**[S] **(Person)** *wrote*[P]: [...]
		E2.2	**Einstein**[S] **(Person)** *was elected*[P] a Foreign Member of **the Royal Society**[O] **(Org)** (ForMemRS) in 1921.
	Subject	E3.1	During an address to Caltech's students, **Einstein**[S] **(Person)** *noted*[P] **that science was often inclined to do more harm than good**[O].
		E3.2	**Einstein**[S] **(Person)** *started teaching*[P] himself **calculus at 12**[O], and as a 14-year-old [...]

e.g., a whole subordinate clause like *that science was often inclined to do more harm than good*. See E3.1 in Table 4.

We developed a short script to quantify them to understand better how many subjects and objects were complex. Therefore, we formulated regular expressions to check if a sentence or noun phrase contained multiple clauses split by punctuation (,;:), or words (and, or, that, thus, hence, because, due, etc.). We then counted subjects and objects as complex if they matched one of these regular expressions. In addition, noun phrases that consumed more than 50% of the sentence were considered complex. And if noun phrases consumed more than 20% of the sentence and the sentence itself consisted of multiple clauses (regular expressions again), we denoted the noun phrases as complex. Note that we are aware of the limitations of such a heuristic. That is why we compared this heuristic to other methods in depth in Sect. 4.4. Returning to our sample, 16.2% of subjects and 74.5% of objects were classified as complex. We iterated over these classifications to verify the filter criteria.

Partial Entity Filter. This filter yielded problematic results because much information was lost, e.g., a whole subordinate clause was broken down into a single entity regardless of where the entity appeared in this clause. In some cases, this filtering completely altered the sentence's original information; see E2.2 for a good example. Here the extraction *Einstein was elected the Royal Society* was nonsense because *Foreign Member* was filtered out. In E2.1, the extracted statement missed that the *philosopher* was *Eric Gutkind*, and thus lost relevant information.

Exact Entity Filter. The exact filter was very restrictive because the number of extractions was reduced from 117.9 to 2.9k. However, the extraction seemed to have good quality. In E1.1, the extraction *Einstein was visiting the US* was correct, but the context about the year *1933* was lost. Extraction E1.2 showed that OpenIE6 was capable of extracting implicit statements like *be Professor of*. Again, the surrounding context about the year and Einstein was lost. Other extractions showed that a co-reference resolution would be beneficial to resolve mentions like *his*, *in the same article*, and *these models*.

Subject Entity Filter. We observed many complex object phrases (74.5% in sum). These complex phrases contained more information than a single entity. Filtering them led to many wrongly extracted statements. In contrast, subject phrases were often simple and might stand for a single entity (only 16.2% are complex). Because of these observations, we developed a subject entity filter, i.e., only subjects had to match entities directly. The idea was to identify subjects as precise entities and keep object phrases in their original form to retain all information.

Results. This filter worked as expected: In E3.1 and E3.2, the subject was identified as the Person *Einstein*, whereas the original information was kept in the object phrase. For example, this filtering allowed us to generate a structured overview of Albert Einstein: (excelled, at math from a young age), (published, hundreds of articles throughout his life), and

[6] We wrongly reported 151k in [17].

(attempted, to generalize his theory of gravitation following his research on general relativity).

PathIE. In addition to analyzing OpenIE6, we investigated how useful PathIE is in extracting relations between the relevant entity types, such as scientists and awards. PathIE allowed us to specify keywords that can indicate a relation. In a first attempt, we applied PathIE with a small relation vocabulary of Wikidata. We exported the English labels and alternative labels of eleven Wikidata properties that describe the relations between the given entity types: academic degree, award received, date of birth, date of death, field of work, member of, native language, occupation, religion, and writing language. For example, the entry *award received* had the following synonyms: award received, award won, awarded, awards received, honorary title, honors, honours, medals, prize awarded, prize received, recognition title, win, winner of, award, and awards.

We exported and evaluated 100 randomly selected PathIE extractions. When several entities were detected in long and nested sentences, PathIE yielded many wrong extractions because the corresponding entities were connected via some verb phrases, e.g., *Einstein return Zurich* from *Einstein visited relatives in Germany while Maric returned to Zurich* or *Written languages write Leningrad*. Filtering them by entity types like (Person, Date) or (Person, Award) revealed more helpful extractions, e.g., *Einstein win Nobel Prize* from *Einstein received news that he had won the Nobel Prize in November*.

However, we encountered severe entity linking issues when analyzing the cleaned OpenIE6 and PathIE extractions. On the one hand, ambiguous terms were linked wrongly. On the other hand, fragments of a text span were linked against an entity although the whole text span referred to a single entity, e.g., only linking *Albert Einstein* in the text mention *Albert Einstein's Theory of Relativity was published in 1916*. These issues directly affected the extraction quality. We stopped the extraction part at this point.

4.1.3 Canonicalization

We used our small relation vocabulary to canonicalize the extractions. This procedure did work out for PathIE because it directly extracted the vocabulary entries from the texts. For example, we could retrieve a list of statements that indicate an *award received* relation. However, further cleaning was required to obtain *award received* relations between persons and awards. We analyzed 100 entries for this relation. Although some extraction were correct, 60 of 100 extractions had linked awards that were not helpful, e.g., *awards*, *doctor*, *medal*, *president* and *master*. The remaining 40 extractions displayed six wrongly identified persons. However, the remaining 34 extractions seemed plausible, although some information was missed, like the *Nobel prize's* category.

Next, we used the same relation vocabulary to canonicalize the OpenIE6 extractions. In brief, the canonicalization procedure did not work. The reason was that the extracted verb phrases did not appear directly in the vocabulary, e.g., see the aforementioned terms for *award received*. Thus, we used a pre-trained English Wikipedia word embedding from fasttext[7] to find similar matches in the relation vocabulary. We adjusted the cleaning parameters (how similar terms must be and how often terms must occur) and canonicalized the OpenIE6 verb phrases. However, most verb phrases were mapped wrongly because the vocabulary was relatively small, e.g., *divorce* was mapped to *date of death* because it was the closest match (in terms of vector space similarity).

We then derived a list of 120 Wikidata properties that involved persons (ignoring usernames and identifiers) to find more matches. We repeated the canonicalization and analyzed 100 extractions obtained by the subject entity filter because it retrieved the most helpful results in the previous step. Most of the canonicalized verb phrases were mapped incorrectly, e.g., mapping *start teach* to *educated at* or *begin* to *death of place* was wrong. For a positive example, the verb phrase *publish* was mapped to the relation *notable work* and *write* to *author*, e.g., *Galileo publish* ($\mapsto$ *notable work*) *Dialogue Concerning the Two Chief World Systems*. Although this relation was correct for a few extractions, most of these mappings were problematic, e.g., *Einstein publish* ($\mapsto$ *notable work*) *his own articles describing the model among them.* Here the object phrase did not contain a notable work in the sense of how we would understand it.

In summary, the canonicalization procedure had many problems for OpenIE6 extractions. The main issue was that the canonicalization procedure only considered the verb phrase, not the surrounding context in a sentence. But this surrounding context is essential to determine the relation, e.g., the verb phrase *use* could refer to many different relations depending on a concrete sentence. In addition, the relation vocabulary obtained from Wikidata might be insufficient because it did not contain verb phrases as we would expect them. Wikidata describes relations by using substantives and nouns, e.g., notable work of, notable work by, notably created by for the relation *notable work*. However, such substantives should typically not be included in the verb phrase of an OpenIE extraction because they are not verbs.

4.1.4 Application costs

We spent much of our time understanding the Wikidata ontology and formulating suitable SPARQL queries to retrieve the utilized vocabularies. The corresponding vocabularies could

[7] https://fasttext.cc/docs/en/pretrained-vectors.html.

be exported directly from Wikidata and did not need transformations besides a concatenation of files. We formulated several SQL queries to analyze, clean, and filter entity annotations and extractions in the toolbox's underlying database. In summary, three persons performed this case study within three person-days.

4.1.5 Generalizability

We had a close look at existing Wikipedia relation extraction benchmarks for evaluation. Unfortunately, these benchmarks are often built distantly supervised, i.e., if two entities appear in a sentence, and both entities have a relation in a knowledge base, then this relation is the class that must be predicted for this sentence. In other words, the relation does not have to appear within the sentence. Furthermore, these benchmarks often require domain knowledge, e.g., if a football player started his career at a sports team, then the football player *played for* this team. This additional knowledge is typically not included in OpenIE methods. OpenIE extracts statements based on grammatical patterns in a sentence: For the previous example, the tool would extract that the football player started his career on the sports team but not that he also played for the team. So we did not evaluate the extraction tool on existing benchmarks because we had reason to expect the quality to be low by design. Moreover, mapping verb phrases to precise relations would also be too challenging. In contrast, we wanted to understand how useful the results were for practical applications.

First, an improved entity linking would have solved several issues in our case study. Next, the handling of complex noun phrases was an issue: Although the exact entity filter was too restrictive, it resulted in suitable extractions. The partial entity filter messed up the original information and was thus not helpful. OpenIE6 and the subject entity filter allowed us to retrieve a list of actions performed by Albert Einstein, for example. However, this filtering did not yield a canonicalized knowledge base by design. Our case study has shown that PathIE could extract relations between scientists and awards. Although we could not evaluate the quality in rough numbers, we spent three person-days designing a possible extraction workflow. Here, the toolbox allowed us to retrieve such semi-structured information in an acceptable amount of time.

4.1.6 What is missing?

The handling of complex noun phrases was a significant issue: On the one hand, the decisive context was lost if phrases were broken down into small entities. On the other hand, if phrases were retained in their original form, the context was kept, but the canonicalization remained unclear. To the best of our knowledge, there is no out-of-the-box solution that will solve these issues.

4.2 Pharmaceutical case study

We applied the toolbox to a subset of the biomedical Medline collection for our second case study. The PubMed Medline is available in different formats, among other things, in the PubTator format, which is supported by the toolbox. We downloaded the document abstracts from the PubTator Service [33].

4.2.1 Entity linking

We utilized existing entity annotations (diseases, genes, and species) from the PubTator Central service [33, 34]. In addition, we selected subsets of MeSH (diseases, methods, dosage forms), ChEMBL [24] (drugs and chemicals), and Wikidata [32] (plant families) to derive suitable entity vocabularies. We developed scripts that retrieved relevant entries from these vocabularies. This step required us to export relevant entries from XML and CSV files into TSV files.

We then applied the entity linker and analyzed the results by going through the most frequent annotations. Our first attempt yielded frequently, but obviously wrongly linked words such as *horse*, *target*, *compound*, *monitor*, and *iris*. These words were derived from ChEMBL because they were trade names for drugs. We found such trade names to be very ambiguous and removed them. Our final vocabulary included 69,502 distinct entities and 300,133 distinct terms.

But we also found annotations such as *major*, *solution*, *relief*, *cares*, *aim*, and *advances*. We went through the 500 most tagged entity annotations to remove such words by building a list of ignored words (188 in sum). We repeated the entity linking by ignoring these words and computed 232.5k entity mentions. We did not apply Stanford Stanza NER (persons, organizations, and more) here because we were interested in biomedical entities. The number of detected entities already seemed to be sufficient, so we continued with the extraction.

4.2.2 Information extraction

OpenIE6. The domain experts were interested in statements between entities. That is why we applied OpenIE6 and analyzed the partial and exact entity filter, i.e., we wanted to obtain entities as subjects and objects. We skipped no filter and subject filter here because they would have produced not-canonicalized noun phrases. OpenIE6 extracted 207.6k extractions and filtering them yielded 88k (partial) and 291 (exact) extractions. Our heuristic estimated 37.8% of the extracted subjects, and 72.1% of the objects as complex.

Table 5 PubMed PathIE example extractions. On the left, the canonicalized relation is annotated

Pharmacy	Treats	P1.1	We tested whether short-term, low-dose *treatment*[P] with the fluvastatin and **valsartan**[S] **(drug)** combination could improve impaired arterial wall characteristics in type 1 **diabetes mellitus**[O] **(disease)** patients
		P1.2	We encountered two cases of cerebellar **hemorrhage**[O] **(Disease)** in patients *treated*[P] with **edoxaban**[S] **(Drug)** for PVT after hepatobiliary surgery during the past 2 years
	Inhibits	P2.1	**Anthraquinone**[S] **(Drug)** derivative emodin inhibits tumor-associated angiogenesis through *inhibition*[P] of **extracellular signal-regulated kinase 1**[O] **(Gene)**/2 phosphorylation
		P2.2	Impact of **aspirin**[S] **(Drug)** on the gastrointestinal-sparing effects of **cyclooxygenase-2**[O] **(Gene)** *inhibitors*[P]
	Induces	P3.1	**Hyperglycemia**[O] **(Disease)**-*induced*[P] mitochondrial dysfunction plays a key role in the pathogenesis of diabetic **cardiomyopathy**[S] **(Disease)**
		P3.2	Conclusions H. pylori **Infection**[S] **(Disease)** appears to *cause*[P] decreases in **Vitamin B12**[O] **(Excipient)**[...]

Exact Entity Filter. The exact entity filter produced only 291 extractions out of 87.1k sentences (47.4k sentences with at least two entities). This method was hence too restrictive and not helpful because the remaining extractions were too few for a practical application.

Partial Entity Filter. A closer look at 100 randomly sampled extractions indicated that many noun phrases were complex again. The partial entity filter mixed up the original sentence information by filtering out the important information. For example, consider the following sentence: *Inhibition of P53-MDM2 interaction stabilizes P53 protein and activates P53 pathway*. Here the partial entity filter extracts the statement: (*MDM2*, *stabilizes*, *protein*). This statement mixed up the original information. Our analysis showed that the vast majority of filtered extractions were incorrect. In addition, OpenIE6 is focused on verb phrases to extract statements (here *stabilizes*).

However, many relevant statements are expressed by using special keywords, e.g., *treatment*, *inhibition*, *side effect*, and *metabolism*. That means that these OpenIE methods will usually not extract a statement from clauses like *metformin therapy in diabetic patients* by design. A similar observation was already made in the original toolbox paper, where OpenIE methods' recall was clearly behind supervised methods (5.8% vs. 86.2% and 6.2% vs. 75.9% on biomedical benchmarks) [15]. Supervised extraction methods would address this problem by learning typical patterns of how a treatment can be expressed within a sentence.

PathIE. To integrate such specialized keywords in the extraction process, we applied the recall-oriented PathIE method. In the previous example, the entities *metformin* and *diabetic patients* are connected via the keyword *therapy*. In this way, PathIE extracted a helpful statement. However, we had to build a relation vocabulary to define these specialized keywords. In cooperation with domain experts, we built such a vocabulary by incrementally extracting statements with PathIE, looking at extractions and example sentences to find out what we were missing. In sum, we had three two-hour sessions to build the final relation (eight relations plus 60 terms) vocabulary. The final PathIE step yielded 430.8k extractions and took two minutes to complete. Some interesting results are listed in Table 5. We then iterated over a sample of 100 of these extractions.

PathIE was capable of extracting statements from long and nested sentences, e.g., a treatment statement in P1.1 in Table 5. However, we also encountered several issues with PathIE. If a sentence contains information about treatments' side effects (also linked to diseases), PathIE extracted them wrongly as the treated condition (See P1.2). A similar problem occurred when a drug therapy was used to treat two diseases simultaneously. Here, PathIE yielded six statements (three mirrored): two therapy statements about the drug and each disease, and one therapy statement between both diseases, which is wrong. In example P2.2, PathIE failed to recognize that aspirin *effects* the inhibitors and is not an inhibitor itself.

A second problem was the direction of extracted relations: A *treats* relation could be defined as a relation between *drugs* and *diseases*. If a relation has precise and unique entity types, then an entity type filter can be used to remove all other, and possibly wrong, extractions. Suppose a disease causes another one (think about a disease that causes severe effects). In that case, PathIE would extract both directions: (a causes b) and (b causes a). For example, PathIE would extract two statements from *myocardial damage caused by ischemia-reperfusion*. Here an entity type filter did not solve the problem because both entities have the type *disease*.

Third, in situations with several entities and clauses within one sentence, PathIE seemed to mess up the original information and extracted wrong statements, e.g., see P3.1, where hyperglycemia did not induce cardiomyopathy. In summary, PathIE could extract statements from complex sentences, but a cleaning step had to be applied afterward to achieve acceptable quality.

4.2.3 Canonicalization

We exported the database statistics for PathIE. We carefully read the extracted verb phrases in cooperation with two domain experts. Verb phrases such as *treats*, *prevents*, and *cares* point toward a *treats* relation, which we included in our relation vocabulary. Phrases such as *inhibits* and *down regulates* may stand for a *inhibits* relation. To find more synonyms automatically, we used a Biomedical Word Embedding [38] that we used in our toolbox paper before. Following this procedure, we defined eight relations with 30 synonyms. We repeated the procedure five times and derived a relation vocabulary of 60 entries. The relation vocabulary was a mixture of verb phrases and keywords that indicated a relation in the text. In sum, we had six sessions of two hours each to build the final relation vocabulary.

However, we noticed that PathIE extractions were problematic when not filtered. Relations like *treats* and *inhibits* also include entity types that we had not expected, e.g., two diseases in treats. We formulated entity type constraints for eight relations to remove such problematic statements. The relations *treats* and *inhibits* looked more helpful because they only contained relevant entity types. We tried to filter relations like *induces* between diseases. Some extractions were correct, but many mixed up the relation's direction (a causes b instead of b causes a). In the end, PathIE was not very helpful for extracting such directed relations because of its poor quality. We stopped the cleaning here, but a more advanced cleaning would be helpful to handle such situations.

4.2.4 Application costs

We spent most of our time designing entity and relation vocabularies and analyzing the retrieved results. The creation of suitable vocabularies took us around one week in sum. The execution of the toolbox scripts was quite simple; see our GitHub repository. To measure the runtime for PubPharm, we applied the PathIE-based pipeline on around 12 million PubMed abstracts (PubMed subset about drugs). The procedure could be completed within one week: Entity detection took two days for the complete PubMed collection (33 million abstracts). PathIE took five days, and cleaning took one day. Hence, such an extraction workflow is realizable for PubPharm with moderate costs.

4.2.5 Generalizability

We already know that OpenIE6 and PathIE have worse performance than supervised methods; see the benchmarks in the original toolbox paper. However, we could design a suitable extraction workflow with an acceptable amount of time (a few weeks of cooperation with nine sessions with experts). OpenIE6 had a very poor recall, and filtering remained unclear. Thus, they were not of interest for PubPharm's purposes.

PubPharm is currently using the PathIE extractions in their narrative retrieval service[8] [16, 19]. Here recall is essential to find a suitable number of results to answer queries. Although the quality of PathIE is only moderate, the quality seems to be sufficient for such a retrieval service. Here, the statement should hint that the searched information is expressed within the document, e.g., that a *metformin treatment* is contained. The main advantage of a retrieval service is that the original sentences can be shown to users to explain where the statements were extracted. In summary, if users are integrated into the process, and the statements' origin is shown, PathIE allows novel applications like PubPharm's narrative retrieval service.

Nevertheless, we encountered several issues: First, PathIE extracted wrong statements if several entities were contained in a sentence. Next, the undirected extractions of PathIE were often problematic if no additional cleaning could be performed (e.g., relations between diseases). Although these issues must be faced somehow, PathIE allowed us an extraction workflow that we could not have realized using supervised methods due to the lack of training data. We would not recommend PathIE for building a knowledge graph because of many wrong extractions that would lead to transitive errors when performing reasoning on the resulting graph.

4.2.6 What is missing?

In this pharmaceutical case study, we focused on relations between pharmaceutical entities. PathIE completely ignored the surrounding context of statements, e.g., dose and duration information of therapies. The coherence of statements was also broken down, e.g., drug, dosage form, disease, and target group of treatments were split into four separate statements. The desired goal would be to retain all relevant information within a single statement. However, PathIE is restricted to binary relations. A future enhancement of PathIE would be desirable to retain all connected entities in a sentence. PubPharm's narrative retrieval service bypassed the problem by using document contexts [18], i.e., statements from the same document belong together. The service used abstracts, and this approximation would not have been possible for full texts

[8] www.narrative.pubpharm.de.

Table 6 Pollux OpenIE6 example extractions. On the left, the corresponding entity filter is shown (partial and subject)

Political sciences	Partial	PS1.1	Stalin wanted all 16 **Soviet**[S] **(NORP)** Republics *to have*[P] separate seats in **UN General Assembly**[O] **(ORG)** but only 3 were given Russia Ukraine Belarus
		PS1.2	This paper seeks to understand why the **United States**[S] **(GPE)** *treated*[P] **Japan**[O] **(GPE)** and Korea *differently*[P] in the revisions of bilateral nuclear cooperation agreements
	Subject	PS2.1	Based on these features, the article suggests that **China**[S] **(GPE)** is poised *to become*[P] **a true global power**[O]
		PS2.2	Prior to the introduction of the Transparency Register the **European Parliament**[S] **(ORG)** *had maintained*[P] **a Register of Accredited Lobbyists since 1996**[O] while the European Commission [...]

because a full-text document might contain several different contexts.

4.3 Political sciences

We applied the toolbox to 10k abstracts from Political Sciences.

4.3.1 Entity linking

The field of Political Sciences displays some distinct differences compared to the biomedical field and encyclopedias like Wikipedia. A notable difficulty lies in the lack of well-curated vocabularies for the domain. This can be mitigated in two ways: by using NER as implemented by Stanza [27] or by constructing/deriving entity vocabularies from general-purpose knowledge bases like Wikidata. We investigated both approaches.

Stanza NER yielded ca. eight tags per document. The extracted mentions seemed sensible, e.g., entities like *USA*, *Bush*, or *the Cold War* were extracted. Problematic was that mentions like *Bush* were identified as a *person* and not linked to a specific identifier. However, Stanza NER also displayed some drawbacks, e.g., it was prone to missing uppercase letters for identifying names. Such restrictions can be problematic in practice because of bad metadata, e.g., abstracts in upper case.

For the second approach, we selected wars (Q198), coup d'états (Q45382), and elections (Q40231) as seed events, since those are likely to be the subject of debate in political science articles. Furthermore, we inductively utilized Wikidata's subclass property (P279) to receive all subclasses of all seed events. We used the SPARQL endpoint to export the corresponding vocabularies by asking for the English label and alias labels for the seed events, all instances of the seed events (P31– instance of), and their subclasses. In total, we collected 2.9k wars, 904 coups, and 79.7k election entries. An evaluation of the toolbox's entity linker showed good performance on wars, while coup d'états and elections were rarely linked sensibly. Our vocabulary included 52,454 distinct entities and 59,813 distinct terms.

However, we increased the linking quality by applying simple rules, e.g., the entity label must contain the term *election*. We derived 3.7k entity annotations linked to Wikidata in sum.

4.3.2 Information extraction

OpenIE6. Due to the lack of comprehensive entity vocabularies, we focused on OpenIE6 in this case study and omitted PathIE. OpenIE6 yielded 147.2k (no filter), 28.6k (partial), 128 (exact) and 7.3k (subject) extractions. Subject phrases tended to be short (only 32.0% were complex), and object phrases tended to be long (74.3% complex) again, like in the previous case studies. We randomly sampled 100 extractions of each filter for further analysis. Again, extractions from small sentences looked helpful, while long sentences led to long object phrases. We picked some interesting results and displayed them in Table 6.

Exact entity filter. Again, the exact entity filter decreased the number of extractions drastically (from 147.2k to 128). But extractions seemed plausible, e.g., *Alexander Lukashenko is president of Belarussian[SIC]* from *Focus on the career and policies of the first Belarussian president, Alexander Lukashenko, elected in 1994.* Another correct extraction was *United States prepares to exit* from *As the United States prepares to exit Afghanistan [...].*

Partial entity filter. In PS1.1, the extraction *Soviet to have UN General Assembly* was wrong because the context about Stalin and separate seats was missed. The extraction in PS1.2, *United States treated differently Japan*, was not helpful because *Korea* was missed. Again, the context that this statement was investigated in that article was lost. We found the extractions of the partial filter not helpful: Either they mixed up the original information, or decisive context was missed.

Subject entity filter. The extraction PS2.1 showed a correct extraction, but then the information that the statement was suggested by an article was missed. Although the sentence of PS2.2 was quite complex, OpenIE6 extracted useful information about the *European Parliament*: *European Parliament had maintained a Register of Accredited Lobbyists since 1996*.

4.3.3 Canonicalization

We exported the most extracted verb phrases and analyzed them. The ten most frequently extracted verb phrases (lemmatized) were: be, have, be in, provide, examine, present, offer, focus on, be with, and may. We skipped the canonicalization procedure here because we already knew that canonicalizing OpenIE6 verb phrases remains unclear (see Wikipedia case study). The more so, when words like *be, provide, offer* or *may* could refer to various relations—again depending on the context.

The exact filter yielded fewer extractions, partial filtering resulted in incorrect statements, and PathIE could not be applied due to the lack of vocabularies. And extractions from the subject filter could hardly be canonicalized to precise relations if the object phrase contained large sentence parts (complex object noun phrases).

4.3.4 Application costs

The application costs for the political domain seemed higher compared to the other two case studies. The lack of curated vocabularies necessitates the creation of such. As demonstrated, this can hardly be done automatically but requires domain knowledge. We exported some vocabularies from Wikidata but missed many entities in the end. In sum, we had four sessions, each 1.5 h, with a domain expert to analyze the results. The case study took us five person-days in sum.

4.3.5 Generalizability

Due to the lack of available benchmarks, we restricted our evaluation to a qualitative level. As another difficulty, simple fact statements, e.g., *Joe Biden is the president of the USA* hardly carried new or relevant information. Still disputed claims, viewpoints, or assessments like *the UK aims to position itself as an independent power after Brexit* might be the subject of study. This often resulted in long clauses for the subjects and objects that are hard to map to the already sparsely recognized named entities. But the subject entity filter allowed us to retain that *UK aims to position itself as an independent power after Brexit* as a suitable extraction. We plan to proceed from here by extracting semi-structured information via the subject filter.

4.3.6 What is missing?

Additionally, the context of a statement is often highly relevant. In the example, the statement loses its information if the context *after Brexit* is omitted. Observations were similar to the Wikipedia case studies: Either the object phrases retained the context but could hardly be handled by filtering methods. Or the object phrases were short and missed information.

4.4 On complex noun phrases

In the following, we use different methods to analyze the complexity of OpenIE noun phrases in more depth. We then continue by looking at the CoreNLP OpenIE extraction tool to generalize our previous findings better, especially, if they are just an artifact of OpenIE6. All implemented extensions, developed scripts, and produced and analyzed data can be found in our repository.[9]

In the previous case studies, we used a self-developed heuristic to estimate if an OpenIE noun phrase is complex. The heuristic was based on information about the length of the noun phrase, whether the sentence has multiple clauses and a few regular expressions. In the following, we applied a bunch of different methods to analyze the complexity of noun phrases in more detail.

Basically, our methods can be grouped into two categories: (1) Part-of-Speech (POS) tag-based and (2) character length-based methods. A POS tag aligns a word of a sentence to a certain part of speech, e.g., nouns, pronouns, adjectives, and more. The evaluation here was based on utilizing such POS tags. For example, we analyzed how many noun phrases contained verbs. Therefore, we used the Universal POS tags.[10] We classified whether an OpenIE noun phrase felt into one of the following categories:

1. Has an adposition (ADP),
2. Has a conjunction (CCONJ),
3. Has nouns only (NOUN, PROPN, PART, DET, NUM, PUNCT),
4. Has nouns and pronouns only (same as for nouns + PRON),
5. Has nouns, pronouns, and adjectives only (same as for nouns + PRON + ADJ), and
6. Has a verb (VERB).

Our motivation for complex noun phrases was that they should include more than a single concept, e.g., a whole sentence fragment or a composition of concepts. That is why we analyzed adpositions to count how many noun phrases

[9] https://github.com/HermannKroll/KGExtractionToolbox/blob/main/README_IJDL2023.md.

[10] https://universaldependencies.org/u/pos/.

contain words like *of*, *in*, *during*, etc., which may indicate a composition of concepts. We also counted conjunctions for the same reason. We also focused on nouns, i.e., we counted how many noun phrases only consisted of nouns. Note that we allowed the following tags for nouns: PROPN to also allow proper nouns, PART to allow fragments like ' in nouns, DET to allow words like *the, a, an, etc.*, NUM to allow numbers (e.g., 3 cats) and PUNCT to allow abbreviations (e.g., St. Paul). In two additional categories, we also allowed nouns and pronouns as well as nouns, pronouns, and adjectives. For comparison, we also counted verbs in noun phrases, which may indicate a relation between concepts. In brief, we understand a noun phrase consisting of nouns only as not being complex. Conjunctions, adpositions, or verbs in noun phrases may likely hint toward a complex concept. To derive POS annotation, we applied the NLP Spacy[11] tool in version 3.1.4. We downloaded the English model (*en_core_web_sm*) for our subsequent analysis.

The second evaluation category was based on character length. The motivation was to understand better the ratio between the length of a noun phrase and the overall sentence length. We assumed long noun phrases to be complex, especially if they were longer than half of the sentence's length, for example. Therefore, we computed the length for each noun phrase and each sentence by counting the corresponding characters. Hence, we counted how many noun phrases were longer than 30%, 40%, 50%, 60%, and 70% of the sentence.

4.4.1 Results

The evaluation results of our noun phrases extracted by OpenIE6 are reported in Table 7. First, extracted subjects were less complex for all methods and all domains. This reflected our previous findings that OpenIE6 subjects seemed less complex. And that objects were rather often complex. For example, 84.2% of all OpenIE6 subjects extracted from Wikipedia consisted of nouns, pronouns, and adjectives only. In other words, 15.8% subjects were thus more *complex* than a single noun. Our initial heuristic estimated 16.2% of the Wikipedia subjects to be complex. This argument also applies to Pharmacy and Political Sciences. Our heuristic estimated around 37.8% (Pharm.) and 32.1% (Pol.) to be complex. The noun+pronoun+adjective estimation revealed that around 42.1% (Pharm.) and 34.4% (Pol.) contained more information than a single noun. Broadly a third of all OpenIE6 objects in all three domains contained a verb. Concerning the noun phrase length, between 25.5 and 28.6% of the objects were longer than 40% of the sentence. Indeed, between 15.1 and 18.2% were longer than 50% of the sentence. This quantified our qualitative impression that many extracted noun phrases consisted of whole sentence fragments.

To better generalize our findings here, we applied another OpenIE tool, namely CoreNLP OpenIE on our data. This method is older (2014) than OpenIE6 (2020) and may likely have different properties. After execution, we obtained 545k extractions for Wikipedia, 930k for PubMed, and 569k for Political Sciences. The first observation was that CoreNLP OpenIE extracted way more statements than OpenIE6 (545k vs. 179k, 930k vs. 210k, and 569k vs. 150.7k). A quick investigation revealed that CoreNLP OpenIE extracted several *similar* statements from sentences, e.g., five extractions from *The quick brown fox jumped over the lazy dog*. Here, the tool extracted three different versions of the subject (quick brown fox, brown fox, fox), the verb phrase *jumped over*, and the two objects (lazy dog and dog) – yielding six extractions in sum. For this example, OpenIE6 extracted a single extraction: (The quick brown fox; jumped; over the lazy dog). Additional filtering might be beneficial here. However, how to do so is challenging, e.g., keeping just the longest extraction in terms of noun phrase length may conflict with the exact or subject entity filtering later on. If the fox was an entity and we just kept the quick brown fox as the only subject, our filtering methods would not produce a result here. But keep this property in mind for the following investigations.

The noun phrase complexity of CoreNLP OpenIE is reported in Table 7. In brief, this method extracted less complex noun phrases for subjects and objects for all three domains measured by our heuristic. A closer look at the other estimation methods revealed that those supported our findings. The ratios of pure noun phrases consisting of nouns or nouns+adjectives+pronouns were clearly above the ratios of OpenIE6.

In our previous manual evaluation [14], we manually counted the complexity of noun phrases for biomedical and new articles. The findings back then revealed that between 53 (biomedicine) and 68% (news) of OpenIE6 extracted objects were classified as complex by raters. For CoreNLP OpenIE, in contrast, we estimated 25% (biomedicine) and 20% (news) as complex objects. Concluding from both findings (this paper) and our previous study [14], the main takeaway here is that complex noun phrases are a frequent issue that must be faced in practice. Although less frequent for CoreNLP OpenIE than for OpenIE6, they are still there. Handling such complex noun phrases by canonicalizing methods like entity filters still remains open.

4.5 CoreNLP OpenIE

In the first case study, we investigated the noun phrase complexity of CoreNLP OpenIE in comparison to OpenIE6. Although the tool seemed to have less noun phrase complexity, how useful are its extractions in practice? First, we

[11] https://spacy.io/.

Table 7 Evaluation of the OpenIE noun phrase complexity: Different methods and features are used to estimate how *complex* an OpenIE noun phrase is. Therefore, the method, its features, and the results for subjects and objects, as well as for all three domains, are reported. Note that the OpenIE6 complexity is based on 179k tuples for Wikipedia, 210k for PubMed, and 150.7k for Political Sciences. For CoreNLP OpenIE, the results are based on 545k tuples for Wikipedia, 930k for PubMed, and 569k for Political Sciences

Method	Features	Wikipedia		Pharmacy		Pol. sciences	
		Subj. (%)	Obj. (%)	Subj. (%)	Obj. (%)	Subj. (%)	Obj. (%)
OpenIE6							
Our heuristic	Mixed	16.2	74.5	37.8	72.1	32.1	74.4
Has adposition	POS Tags	10.5	77.3	30.4	79.6	24.8	76.3
Has conjunction	POS Tags	0.3	2.3	1.7	4.5	1.9	6.0
Has nouns only	POS Tags	43.0	9.2	32.5	6.3	37.9	7.2
Has nouns+pronouns only	POS Tags	76.0	10.6	40.9	6.5	50.3	7.8
Has n.+pron.+adj. only	POS Tags	84.2	15.1	57.9	12.0	65.6	13.3
Has verb	POS Tags	5.7	29.3	16.7	33.4	13.1	36.9
> 30%-of-Sentence	Char. Length	4.3	43.1	14.1	40.4	10.8	41.9
> 40%-of-Sentence	Char. Length	1.7	28.6	6.7	25.5	5.0	27.3
> 50%-of-Sentence	Char. Length	0.7	18.2	2.9	15.1	2.1	17.0
> 60%-of-Sentence	Char. Length	0.2	11.4	1.2	8.5	0.8	10.1
> 70%-of-Sentence	Char. Length	< 0.1	6.5	0.4	4.0	0.3	5.3
CoreNLP OpenIE							
Our heuristic	Mixed	3.1	46.8	4.0	53.0	2.8	53.0
Has adposition	POS Tags	0.3	45.4	0.9	52.1	0.3	52.6
Has conjunction	POS Tags	0.1	0.1	< 0.1	0.1	< 0.1	< 0.1
Has nouns only	POS Tags	45.2	28.0	50.8	19.3	53.6	20.3
Has nouns+pronouns only	POS Tags	80.1	31.1	59.5	19.6	66.6	21.4
Has n.+pron.+adj. only	POS Tags	91.2	41.3	82.3	31.5	88.2	33.1
Has verb	POS Tags	7.2	32.4	15.3	40.3	10.8	37.5
> 30%-of-Sentence	Char. Length	0.5	19.5	1.1	22.8	0.8	20.3
> 40%-of-Sentence	Char. Length	0.1	10.6	0.2	11.6	0.2	10.6
> 50%-of-Sentence	Char. Length	< 0.1	5.1	< 0.1	4.9	< 0.1	4.8
> 60%-of-Sentence	Char. Length	< 0.1	2.1	< 0.1	1.6	< 0.1	1.7
> 70%-of-Sentence	Char. Length	< 0.1	0.6	< 0.1	0.3	< 0.1	0.4

had a close look at existing NLP benchmarks [3, 6, 11]. In brief, OpenIE6 outperformed CoreNLP OpenIE. We made a similar observation when quantifying how much information these tools keep in practice; see [14]. These findings were expected because the CoreNLP OpenIE is way older and less advanced than OpenIE6.

However, our entity filtering approaches have revealed that handling complex noun phrases remained unclear because either the exact filter yielded too less extractions in practice, or the partial filter mixed up the original sentence's information. Due to a less noun phrase complexity when using CoreNLP OpenIE, we formulated the questions: *1. Does the partial entity filter obtain a better overall quality? 2. Does the exact entity filter yield a sufficient number of extractions in practice? 3. Should we switch back to CoreNLP in combination with entity filtering?*

Table 8 We report the number of CoreNLP OpenIE extractions computed by the different entity filters (no, partial, exact, subject) for our three domains

CoreNLP OpenIE				
Ent. Filter	*#No*	*#Part.*	*#Exact*	*#Subj.*
Wikipedia	544k	171k	36k	272k
Pharmacy	929k	466k	7.7k	112k
Pol. Sci.	568k	11.2k	1.2k	30k

4.5.1 Extraction and filtering

We applied the CoreNLP OpenIE method to our previous case study data by using the same entity annotations for filtering as we used for OpenIE6. The resulting numbers of extractions for each entity filter (no, partial, exact, and subject) are reported in Table 8.

First, the overall number of extractions without filtering was higher than using OpenIE6. We commented on this finding in the previous subsection. For the exact filter, the number of remaining extractions was higher than in the OpenIE6 setting: This time we obtained 36k, 7.7k, and 1.2k extractions instead of 2.9k, 291, and 128 extractions. However, how useful were these extractions? So, we (two authors) performed a qualitative evaluation of the filtered results. We randomly sample 50 extractions for each filter (partial, exact, and subject) and each domain, i.e., 450 in total.

4.5.2 Wikipedia

Partial Filter. The results of this filter were similar to our findings for OpenIE6. We saw some good extractions like (Dahleh, is, professor) from *Munther A. Dahleh [...] is the William Coolidge Professor [...]*. However, we also saw many situations in which the partial filter mixed up the original information, e.g., (Birkeland, was born to, Birkeland) from *Birkeland was born in Christiania (Oslo today) to Reinart Birkeland and Ingeborg [...]*, or (Alexander von Humboldt, is, German) from *Alexander von Humboldt is also a German ship named after the scientist [...].*

Exact Filter. Although the exact filter yielded better extractions, the question was how useful were the extractions in the end. Suppose the following three examples: 1. (Schuenemeyer, is president of, Colorado) from *Schuenemeyer is President of Southwest Statistical Consulting, Cortez, Colorado*. 2. (Niebur, was, president) from *Niebur [...] was president of the National Association of Graduate*. 3. (Wegelin, succeeded langhans as, director) from *[...] Wegelin succeeded Langhans as director of the Anatomical institute*. In all cases, the extraction was syntactically correct. However, the extractions were not useful. Schuenemeyer is not the president of the state of Colorado. He is the president of an organization in Colorado. The organization/affiliation of Niebur's presidency was missed, too. Wegelin indeed succeeded Langhans as a director, but in which position?

Subject Filter. This filter yielded the original object phrases that were extracted by CoreNLP OpenIE, e.g., (Faruque, maintained, active research team) from *Faruque maintained an active research team in icddr [...]*, or (Thoguluva Shesadri Chandrasekar, is, Indian gastroenterologist) from *Gastroenterologist Thoguluva Shesadri Chandrasekar (born 1956) is an Indian gastroenterologist [...].* However, we found that these object phrases were shorter than for OpenIE6, and hence, did contain less information.

4.5.3 Pubmed

Partial Filter. Similarly to the Wikipedia findings, we found it hard to evaluate extractions like (Patients, is with, Disease) from *We identified 8 patients (7 with ALS and 1 with SMA) with motor neuron disease [...].* Although the extraction might be rated as correct, it was not very helpful. The information about the number of patients and which concrete disease was missed. Another extraction was (Injection site reactions, were considered by, Patients) from *Local injection site reactions, including swelling [...], were considered mild or moderate by the patients [...].* The extraction missed how the reactions were considered. So is it correct? Likely yes, but useless.

Exact Filter. (Granulomas, presence of, lymphadenopathy) from *Years later, the presence of pathologic submandibular lymphadenopathy was identified and biopsied, revealing non-caseating granulomas* was a wrong extraction. In contrast, the following three extractions looked correct: 1. (Preterm birth, is contributor to, infant death) from *Preterm birth (PTB) is the largest contributor to infant death in sub-Saharan Africa [...].* 2. (abpa, is usually associated with, respiratory diseases) from *Allergic bronchopulmonary aspergillosis (ABPA) [...] is usually associated with underlying respiratory diseases such as asthma or cystic fibrosis.* 3. (Testicular cancer, affect, men) from *Testicular cancer and Hodgkin's disease are among the most common malignancies to affect young men of reproductive age.* However, much information was still lost: Which men are affected?

4.5.4 Pollux

Partial Filter. Again, the partial filter was problematic, e.g., consider the extraction (Woodward, once again pulls back, Washington) from *Woodward once again pulls back the curtain on Washington [...].* Alternatively, consider: (Switzerland, member of, UN) from *Prior to its full membership in the United Nations, Switzerland was an active observer and even an active member of many specialized UN agencies.* The first extraction missed what was pulled back, and the second one was problematic, too: Here, Switzerland was a member of specialized UN agencies. So UN was detected as an entity, but the rest was missed.

Exact Filter. Extractions like (Putin, is more isolated after, nearly a decade) from *After nearly a decade in power, Putin is more isolated than ever* looked syntactically correct. Another one was (Chaldeans, is in, Iraq) from *[...] experienced by the Chaldeans in Iraq in the last two decades.* We observed many (s, *is in*, o) extractions based on the word *in*. In addition, we also observed problems with '-based extractions like (Nkrumah, of, Ghana) from *The Case of Nkrumah's Ghana.*

Subject Filter. Analogous to our previous observations, the subject filter yielded results of mixed quality. We observed extractions like (South African Defence force, facilitated, relocation of about 4000 bushmen from military bases) from *In March 1990 the now defunct South African Defence Force facilitated the relocation of about 4000 bushmen from military bases [...]* which correctly repeated the gist of the

original sentence but omitted context information (e.g., when the relocation happened and that the defence force was defunct). Yet again, short object phrases can lead to rather useless extractions, e.g., (Spain, is second most important country in, terms) from *In the case of Wind Energy, and in terms of production, Spain is the second most important country […].*

4.5.5 Results

We observed that CoreNLP OpenIE indeed extracted less complex noun phrases than OpenIE6. However, these less complex noun phrases also mean that less context and coherence of the sentence was kept. The partial filter still mixed up with the original information or broke down information into pieces. The exact filter retrieved a higher number of extraction, but the overall quality seemed to be lower than in the OpenIE6 setting, likely because the CoreNLP tool itself had a lower extraction quality. The subject filter still seemed to work out: Subjects were linked to entities, and objects remained not filtered. However, we would still recommend using OpenIE6 for subject filtering. On the one hand, OpenIE6 had a better overall extraction quality (see NLP benchmarks). On the other hand, OpenIE6 extracted longer noun phrases as objects, i.e., more information is kept in that objects. The key takeaway here was that our previous findings for OpenIE6 also applied for OpenIE, allowing a better generalization of our overall findings.

4.6 A remark on quantification

A good question is why our evaluation was mainly qualitative instead of quantitative in nature. On the one hand, existing NLP benchmarks already report on the pure extraction quality and, likely, have a better quality than we would achieve. On the other hand, our goal was to discuss the challenges of information extraction workflows in digital libraries. For example, although the extraction (Patients, is with, Disease) might be seen as syntactically correct, it still does not seem useful in practice. And even worse, our workflow relied on the quality of entity detection, information extraction, filtering and canonicalization, so that each step might lead to subsequent errors. As an example, we quantified the CoreNLP OpenIE extractions of Wikipedia for the partial filter. We would rate 17 of 50 as correct. However, twelve of them were about persons, and six of them had wrongly identified entities. And even worse, some of the correct ones had only partial person names tagged, so just Einstein or Turing, instead of their full names. For the exact filter on Wikidata, we would rate 43 of 50 as correct— but 21 of them had wrongly linked entity types (Washington as a location instead of a person). In the end, we found the quantification too challenging, and the resulting numbers could still be wrong and hence, misleading in the end. That is why we focused on a qualitative study to show the opportunities and drawbacks of such inf. extraction workflows.

5 Advanced canonicalization

Our initial verb phrase canonicalization approach was based on designing a relation vocabulary, i.e., define relations plus a set of synonyms. Such a design can be challenging, as our case studies showed. Canonicalizing verb phrases without considering their sentence contexts remained unclear. Subsequently, we discuss another verb phrase canonicalization based on clustering.

Vashishth proposed CESI to canonicalize OpenIE extractions by clustering noun and verb phases with the help of side information [31]. We wanted to investigate how useful this idea is in practice, i.e., clustering verb phrases that would not require the design of a relation vocabulary. Therefore, we implemented an additional canonicalization method into our toolbox that works as follows: 1. All verb phrases of the extractions are retrieved. 2. These verb phrases are embedded by word embedding that must be given as input. 3. Clustering is performed, and the results are shown to the user.

However, by implementing the last step, we followed the procedure of CESI.[12] They used agglomerative clustering to bypass the need for a pre-given number of clusters. However, a threshold must be provided for splitting the actual clusters. And especially this threshold caused issues for us: How to select a *suitable* threshold?

Here, we used the same Wikipedia Word Embedding as in our case studies before. And, we used the OpenIE6 extractions again. Using the default threshold of 0.429 (see CESI implementation) yielded 351 clusters for 1062 distinct verb phrases from Wikipedia. One cluster, for example, contained the verbs *stand* and *sit*. Another cluster contained the verb phrases *be take, take over, to take, take up, take on, have take, have take over, to take up*. One cluster even contained 629 different verb phrases. We obtained 380 clusters for distinct 1145 different verb phrases from our Political Sciences sample. Alternatively, a threshold of 0.5 yielded 150 and 165 clusters. A threshold of 0.6 yielded 20 and 33.

First, verb phrases need eventually be better cleaned (removing words like *be, to, up, on, by, etc.*) for a practical application. Second, selecting a suitable threshold is challenging. In the end, such a clustering approach did not solve the overall problem that we faced in our case studies. Verb phrases like *use* require the sentence's context information to be reliably canonicalized because they could refer to many different relations. However, such a clustering might give first ideas of which relations could be hidden in the text. So

[12] https://github.com/malllabiisc/cesi/blob/master/src/cluster.py.

it could be used to create a relation vocabulary. Nevertheless, we already developed a script in the original toolbox to export which verb phrases appear most frequently across the collection.

6 Non-English texts

Digital libraries cover a large quantity of texts in different languages. This is especially true for national libraries, e.g., the German National Library or the Royal Library of the Netherlands. In such cases, there is a need for information extraction tools supporting those languages. However, besides some notable exceptions (CoreNLP), most tools are not capable of dealing with non-English texts. They are thus limited in usage for such cases. This is because, besides huge advances in natural language processing in the last decade, there is a clear lack of research in this area regarding texts in languages other than English; see [4] for a good discussion. Thus, other solutions are needed to adapt to non-English texts.

One solution for a couple of languages might be to utilize machine translation for the documents. There is work in the direction of translating training data to train OpenIE systems for other languages [12]. Our idea here was to translate the non-English text into English and apply the toolbox on top of the translation. This approach did not require to adjust the actual methods or retrain NLP models. And, if possible, it would allow utilizing the toolbox's methods on a larger variety of languages since modern machine translation systems support a myriad of languages. That is why we investigated if we can handle Non-English texts (here: German texts) by using automated machine translation. According to this idea, we formulated our research question:

Could machine translation be a solution to handle non-native English texts? And if, how well does the workflow apply here?

6.1 Content

For this small case study, we again focused on the previous three domains: Wikipedia, Pharmacy, and Political Sciences. We manually selected the Wikipedia articles of five famous scientists (Albert Einstein, Alan Turing, Max Weber, Sir. Roger Penrose, and Fritz Jakob Haber). We downloaded the English and German abstracts of these articles. We used the English abstracts for comparison, i.e., the basic idea was to compare sentences from the original English article and from the German-to-English translated one that contain a *similar* information. We were aware that Wikipedia articles might have different levels of detail in different languages. For Pharmacy, we asked a domain expert to provide us with ten pharmaceutical articles that contain an English and a German abstract. We downloaded four articles from *Krankenhauspharmazie*, three from *Phytotherapie*, and three from *Die Pharmazie*. For Political Sciences, we randomly sampled ten articles from the Pollux dump that contained an English and a German abstract. We used the English abstract for Pharmacy and Political Sciences to compare the extractions. The articles should—at best—contain the same information in both languages, i.e., the German-to-English translated version should be similar to the actual English hand-written version.

6.2 Translation service

For the translation, we used the known online service DeepL.[13] DeepL is free-to-use for documents up to 5,000 characters. Additionally, it offers a simple online API and can be adapted for practical scenarios. Note that the English, German, and German-To-English translated abstracts are available in our toolbox repository.

6.3 Statistics

We applied the same extraction workflow as we did for our main case studies, i.e., we used the same entity vocabularies as we used for the corresponding domain in our OpenIE6 case study. We did not adjust any vocabulary for this investigation.

Statistics about this case study's data are listed in Table 9. The Wikipedia articles contained 82 sentences, whereas the German-to-English translated version only contained 55 articles. For Pharmacy, the original English articles contained 14 sentences more, and for Political Sciences, the difference was three. For Wikipedia, 58 of 82 (70%) English sentences contained two entities comparable to the translated version, whereas 37 of 55 (67%) sentences contained at least two entities. The reason might be the different levels of detail in the English and German articles.

For Pharmacy, the number of sentences was decreased by 19%, the number of sentences with two entities by 16%, and the number of detected entities by 21%. For Political Sciences, the numbers of sentences with two entities, NER tags and EL tags were equal except for an entity linking problem: DeepL translated a German fragment to 1980s and 1990s, which were wrongly linked to a plethora of different Wikidata entities: 421 wrong links in total. For the subsequent analysis, we applied the same workflow as in our previous case studies, i.e., applied OpenIE6 with the no filter option; see Table 10 for statistics.

For the subsequent qualitative analysis, we (two authors) evaluated the pure OpenIE6 extractions (i.e., no filtering) to analyze how much information is kept from the original German sentences and how these extractions compare to the original English version. Table 11 shows a comparison of

[13] http://deepl.com.

Table 9 Statistics of our Non-English Case-Study. The numbers of sentences (#Sent.), sentences with at least two detected entities (#with2E), and the number of NER and EL tags are shown. T. denotes the German-to-English translations. *Note that 421 are wrongly linked entities

	Sentences		Entity Det.	
	#Sent.	*#with2E*	*#NER*	*#EL*
Wiki.	82	58	157	143
Wiki. T.	55	37	86	78
Pharm.	89	44	–	147
Pharm. T.	75	38	–	121
Pol. S.	70	11	27	3
Pol. S. T.	67	11	27	424*

Table 10 Translation Case Study: We report the number of extractions obtained from applying OpenIE6 with different entity filters (no, partial, exact, subject)

OpenIE6				
Ent. Filter	*#No*	*#Part.*	*#Exact*	*#Subj.*
Wikipedia	229	200	4	71
Wikipedia T.	119	78	–	17
Pharmacy	201	66	–	10
Pharmacy T.	180	68	–	8
Pol. Sci.	161	6	–	11
Pol. Sci. T.	167	4	–	8

OpenIE6 extractions from English and German-to-English translated texts. In addition, we show the original German texts.

6.4 Wikipedia

Again, remember that Wikipedia abstracts may differ in the levels of detail between the English and German versions. First, extracting information from the first sentence of Wikipedia, the description of who the scientist was, usually worked very well. For example, the extracted statements for Albert Einstein only differed by the word *theoretical* in the object because it was not mentioned in the German text. We made a similar observation for the other four scientists: If their descriptions were the same in English and German, then the translated version resulted in the same statements. Small derivations like that Alan Turing was described as a mathematician, philosopher, computer scientist, logician, and theoretical biologist in the English Wikipedia. In contrast, the translated text yielded the extraction that Turing was a logician, mathematician, cryptanalyst, and computer scientist.

Another sentence about the famous work of Einstein, see Table 11, yielded that Einstein is known for developing the theory of relativity from the English Wikipedia. In the German version, however, the information was stated in a nested version, i.e., the translated version was: *Einstein's main work, the theory of relativity, [...]* which did not yield a statement that he is known for his theory. So small changes in the formulation were decisive in whether a statement was extracted.

Another interesting finding was about Max Weber's occidental rationalism and the disenchantment of the world. The translation for this sentence worked very well, but the final extraction then yielded different statements than the English version, mainly because the formulation was quite different. However, that his work was developed by the unity of a leitmotif was still correctly extracted.

The German statement about Albert Einstein: *Für seine Verdienste um die Theoretische Physik, [...], erhielt er den Nobelpreis des Jahres 1921, der ihm 1922 überreicht wurde* was well translated into English: *For his services to theoretical physics, [...], he was awarded the Nobel Prize of 1921, which was presented to him in 1922.* OpenIE6 yielded the correct extraction that *he received the 1921 Nobel Prize*. The English article stated that *He received the 1921 Nobel Prize in Physics [...].* For this sentence, the extraction also contained the information that *he received the 1921 Nobel Prize* **in Physics**.

In brief, we observed many well-translated sentences and hence, many extractions that were comparable to the original English version, except for minor changes between the different articles.

6.5 Pharmacy

The first statement (see Table 11) about the head and neck region tumors yielded similar extractions except for some slight formulation derivations. In particular, the domain-specific terms were well translated here. This was also reflected by the number of detected entities which was quite close between the English and the German-to-English translated versions.

An interesting finding was the second statement. Although the German sentence was well translated and close to the original English version, OpenIE6 extracted two statements for the English version and only one for the translated version. The difference was based on a missing comma in front of the last *and* in the translated sentence. We manually added the comma and OpenIE6 yielded two statements again. Another finding was about formulations in the articles. The English abstracts tended to use the active formulation *we show*, whereas the German abstracts, and hence, the translated version tended to use the passive style like *it has been shown*. OpenIE6 extracted statements from the active version, but not from the passive version.

Overall, we observed many useful extractions from the translated version, and these extractions were close—except for some formulations—to the original English extractions.

Table 11 Comparison of OpenIE6 extractions from English and German-to-English translated texts

English	German-To-English Trans	German
Wikipedia		
Albert Einstein[S] was a **German-born theoretical physicist**[O]	**Albert Einstein**[S] was a **German-born physicist**[O]	Albert Einstein war ein gebürtiger deutscher Physiker
Einstein[S] is *best known*[P] **for developing the theory of relativity**[O]	**Einstein's main work**[S], the theory of relativity, *made*[P] **him world famous**[O]	Einsteins Hauptwerk, die Relativitätstheorie, machte ihn weltberühmt
Weber's main intellectual concern[S] *was*[P] in **understanding the processes of rationalisation**[O], secularisation, and the ensuing sense of "disenchantment	Even though his work is fragmentary in character, **it**[S] *was nevertheless developed*[P] **from the unity of a leitmotif**[O]: occidental rationalism and the disenchantment of the world it brought about	Auch wenn sein Werk fragmentarischen Charakter hat, wurde es dennoch aus der Einheit eines Leitmotivs entwickelt: des okzidentalen Rationalismus und der damit bewirkten Entzauberung der Welt
Pharmacy		
Tumors in the head[S] and neck region *include*[P] a **heterogeneous group of carcinomas whose treatment has advanced in recent years**[O]	**Tumors in the head**[S] and neck region *comprise*[P] a **heterogeneous group of carcinomas for whose therapy progress has been observed in recent years**[O]	Tumoren im Kopf-Hals-Bereich umfassen eine heterogene Gruppe von Karzinomen, für deren Therapie in den letzten Jahren Fortschritte beobachtet werden konnten
This work[S] *focuses*[P] **on radiation therapy**[O], a treatment option with possible short- and long-term complications, and **the resulting consequences for the patients' quality of life**[O]	**The focus here**[S] *will be*[P] **on radiation treatment**[O], a treatment option with potential short- and long-term complications and the resulting consequences for patients' quality of life	Im Vordergrund soll hier die Strahlenbehandlung stehen, eine Behandlungsoption mit möglichen kurz- und langfristig auftretenden Komplikationen sowie den daraus folgenden Konsequenzen für die Lebensqualität der Patienten
Political sciences		
Beginning with the **Mont Pelerin Society**[S], *founded*[P] by **the Austrian economist and philosopher Friedrich v. Hayek in 1947**[O], [...]	Starting with the **Mont Pelerin Society (MPS)**[S], *founded by*[P] **Friedrich v. Hayek in 1947**[O], [...]	Ausgehend von den Mont Pelerin Society (MPS), die 1947 von Friedrich v. Hayek gegründet wurde, [...]
In his view **they**[S] would finally *lead to*[P] ***'The Road to Serfdom'***[O], that is the title of his famous book published in 1944	**They**[S] *would lead*[P] to the ***'road of servitude'***[O] according to the title of his book published in 1944	Sie würden auf den 'Weg der Knechtschaft' führen, so der Titel seines 1944 veröffentlichten Buches

6.6 Political sciences

An example statement about the funding of the Mont Pelerin Society can be found in Table 11. Here, OpenIE6 yielded nearly the same statement for both versions, the English and German-to-English translation. The only difference was the detail, e.g., that Friedrich v. Hayek was an Austrian economist and philosopher, which was not included in the German text.

Another example about the *Road of Serfdom*, a famous book, revealed problems with the translation. The German word *Knechtschaft* was translated into *servitude*, which was not the correct title of the book (*Serfdom*). However, the extraction that *they lead to the road of Serfdom/servitude* was similar. OpenIE6 although extracted correctly that the *famous book* or *his book* was published in *1944* for both versions.

Another long English sentence was: *This article deals with the role of policy learning for the genesis of Austrian art policy during the 1980ies and early 1990ies and seeks to utilize the conclusion drawn from this analysis for the further development of the concept of policy learning.* The German version *Dieser Artikel befasst sich mit der Rolle des Policy Learning für die Genese der österreichischen Kunstpolitik in den 1980er und frühen 1990er Jahren und versucht, die Schlussfolgerungen aus dieser Analyse für die Weiterentwicklung des Konzepts des Policy Learning zu nutzen.* was translated in *This article addresses the role of policy learning in the formation of Austrian reproductive technology policy during the 1980s and early 1990s and seeks to make findings in this regard useful for a further development of the*

conception of policy learning. OpenIE6 then extracted four extractions for each sentence, respectively. Three of these statements were nearly identical except for some wording. The fourth statement differed in the level of detail in the object phrase: *This article seeks* (*to utilize the conclusion* vs. *to utilize the conclusion drawn from this analysis for the further development of the concept of policy learning*).

7 Discussion

In the following, we discuss how suitable nearly unsupervised extraction workflows are in digital libraries by considering technical and conceptual limitations. Furthermore, we give best practices on what to do and when supervision is necessary.

7.1 Toolbox improvements

The toolbox filtered verb phrases by removing non-verbs (stop words, adverbs, etc.) and verbs like *be* and *have*. Here negations in verb phrases were lost, too. We implemented a parameter to make this behavior optional. Next, we implemented the subject entity filter that was useful in Wikipedia and Political Sciences. Here a statement's subject must be linked to an entity, but the object can keep the original information. In particular, when subject noun phrases were short and object noun phrases were complex, the subject filter could be used to construct a semi-structured knowledge base, e.g., showing all actions of *Albert Einstein* or *positions* that the *EU* has taken. In addition, we implemented a clustering-based canonicalization procedure like proposed by [31].

7.2 Technical toolbox limitations

In addition, the dictionary-based entity linker fails to resolve short and ambiguous mentions. These wrongly linked mentions cause problems in the cleaning step (entity-based filters). Here, more advanced linkers would be more appropriate to improve the overall quality. A co-reference resolution is also missing, i.e., resolving all pronouns and mentions that refer to known entities. PathIE is currently restricted to binary relations but might be extended to extract more higher-ary relations, e.g., by considering all connected entities via a verb phrase or a particular keyword like treatment. A suitable cleaning would be possible if the relation arguments (subject and object) could be restricted to entity types.

7.3 Restrictions of unsupervised IE

The first significant restriction of unsupervised methods is their focus on and thus restriction to grammatical structures. Suppose the example: *The German book Känguru-Chroniken was written by Marc-Uwe Kling*. Here unsupervised methods may not extract that the language of the work is German.

In common relation extraction benchmarks, such relations appear and can be learned and inferred by modern language models [5, 21]. However, we argue that such extractions require high domain knowledge, typically unavailable in unsupervised extraction methods. Similar examples could be made in specialized domains like Pharmacy (treatments, inhibitions, etc.). Moreover, it is not possible to integrate this knowledge into unsupervised models by design: The model would need training data to infer such rules and, thus, be supervised. We do not expect unsupervised models with access to comprehensive domain-specific knowledge soon. And even if applying such a model in a new domain with new types of relations would then again require a re-training of that model, e.g., for treatment relations in Pharmacy.

Our case studies showed that OpenIE6 extracts noun phrases in two ways: Either noun phrases are short and miss relevant information from the sentence. These phrases are easier to handle but may be unhelpful in the end. Or the noun phrases are long and complex but retain the original information. Indeed, our analysis in Sect. 4.4 revealed that many noun phrases, especially objects, were complex. Handling complex phrases requires more advanced cleaning methods. Although CoreNLP OpenIE extracted less complex noun phrases, the overall problem of how to handle such noun phrases still remained.

The toolbox canonicalization procedure for relations considers only the verb phrases, not the surrounding context. Verb phrases like *uses*, *publish*, and *prevent* could refer to a plethora of relations. In the end, more advanced methods are required for a suitable canonicalization quality. Even clustering-based methods will not solve this issue by design, if the sentence context is not considered. Especially, canonicalizing OpenIE6 verb phrases to precise relations was not really possible.

7.4 Handling non-english texts

Although our case study in Sect. 6 was preliminary, it showed the potential of modern machine translation. Even complicated and nested sentences were well translated, and the information extraction method yielded similar extractions in all three domains. Instead of acquiring cost-intensive training data to train information extraction models for non-English languages, translating such languages to English could be a suitable alternative here. However, performing translations could still be challenging if languages are underrepresented.

7.5 Application and costs

Although we observed several issues and limitations, these methods can be used to implement services in digital

Table 12 The table summarizes the measured runtimes for the samples and gives an estimation for the whole collection

		Wikipedia		Pharmacy		Political sciences	
		Sample	Estimation	Sample	Estimation	Sample	Estimation
2013 Server – Nvidia GTX 1080 TI & 2xCPU (8/16) & 377GB DDR3 Memory							
Entity Det.	*NER*	10.5 min	19.4 days	–	–	10.1 min	21.6 h
	EL	0.6 min	1.2 days	1.2 min	2.8 days	0.7 min	1.4 h
Extraction	*PathIE*	2.6 min	4.7 days	2.0 min	4.6 days	–	–
	OpenIE6	53.6 min	98.8 days	74.0 min	170.0 days[14]	55.4 min	5.0 days
	CoreNLP	6.7 min	12.2 days	7.3 min	16.6 days	5.0 min	11 h
Cleaning		< 1 h	< 1 day	< 1 h	< 1 day	< 1 h	< 1 day
2021 Server – Nvidia A40 & 2xCPU (24/48) & 2TB DDR4 Memory							
Entity Det.	*NER*	4.0 min	7.4 days	–	–	3.8 min	8.2 h
	EL	4.2 sec	3.1 h	9.0 sec	8.3 h	5.9 sec	14.4 min
Extraction	*PathIE-32*	1.5 min	2.7 days	1.2 min	2.9 days	–	–
	PathIE-96	2.3 min	4.3 days	2.6 min	5.9 days	–	–
	OpenIE6	18.6 min	34.3 days	26.2 min	60.1 days	19.6 min	1.8 days
	CoreNLP	3.3 min	6.1 days	3.3 min	7.5 days	2.3 min	5.0 h
Cleaning		< 1 h	< 1 day	< 1 h	< 1 day	< 1 h	< 1 day

libraries. We summarize the measured runtimes and computed estimations for the corresponding collections in Table 12.

Consider our PubPharm project, for example: PathIE could enable a graph-based retrieval service with moderate costs [16]. Around nine sessions with experts and moderate development time were necessary to implement a workflow. The computation of PathIE took 2 min on our sample and was estimated to take 4.6 days for the whole PubMed collection. Indeed, PubPharm could perform the complete extraction workflow in one week.

Our current cooperation with Pollux revealed that OpenIE6 could bring more structure to this domain. We will continue our work with Pollux by focusing on research questions that we would like to answer with semi-structured information derived from OpenIE6 with subject filtering.

On our server with an Nvidia GTX 1080 TI, the computation of OpenIE6 took 55.4 min on the Pollux sample and is estimated to take five days for the complete collection. For Wikipedia the sample took 53.6 min, and all English articles would require 98.8 days. Note that we used a single GPU from 2016. Hence the workflow can be accelerated with a modern GPU and parallelized by utilizing multiple GPUs. In addition, OpenIE6 can also be restricted to sentences that contain at least two entities. Here the runtime was decreased from 55.4 to 22.4 min (Pollux) and 53.6 to 41.4 min (Wikipedia). CoreNLP OpenIE took way less time than OpenIE6, i.e., was estimated to take 12.2 days for the complete Wikipedia, 16.6 days for the PubMed corpus, and 11 h for the Political Sciences corpus.

7.5.1 Server 2021

As an extension, we measured the runtime performance on our latest server from 2021. In contrast to our old server, this had two Intel(R) Xeon(R) Gold 6336Y CPU @ 2.40GHz (24 cores and 48 threads each), 2TB DDR4 main memory, and nine Nvidia A40 GPUs with 48GB memory. Note that we only utilized a single GPU for this comparison. Again, the runtimes are reported in Table 12. The main finding here was that the runtime was decreased in GPU-intensive tasks (NER or OpenIE6) by a factor of about three. In CPU-intensive tasks (EL + PathIE + CoreNLP OpenIE), we utilized all CPU threads (96). The entity linking runtime decreased clearly and was estimated to take less than a half day for all three domains. CoreNLP OpenIE achieved a speedup of about a factor of two. And for PathIE, we made an unexpected observation: Utilizing all 96 threads took about double the time than utilizing only 32 threads. PathIE utilizes the Java Stanford CoreNLP tool for generating sentence dependency parses, which might not scale well or might have resource-limited boundaries (e.g., I/O from disk).

7.6 Best practices

Subsequently, we give some advice that we can deduce from our case studies. OpenIE6 handles short and simple sentences well. Here the exact entity filter will produce suitable extrac-

[14] We wrongly reported 98.8 days in [17].

tions but decrease the recall drastically. The partial entity filter improves the recall but often messes up the original information. We recommend two strategies for long and complex sentences:

First, do not use the exact or partial filter because important information can be missed. Use the subject filter to retrieve precise entities as subjects and the original information in objects. This filter allows the construction of semi-structured knowledge bases, e.g., positions that were taken by the *EU* or actions that *Albert Einstein* has done. Another option is to use no filter, but then, the extractions are not cleaned in any way.

Second, PathIE can find specialized relations that are expressed by keywords, e.g., treatment and therapy. But PathIE requires directed relations that must be cleaned by entity type constraints. Detecting such relations via PathIE is fast and probably cheaper than training supervised extraction models. However, PathIE will fail if several entities of the same type are mentioned within a sentence, e.g., side effects of treatments. Here supervised methods are required to achieve suitable quality. Another limitation of PathIE and our canonicalization procedures is that a verb phrase/keyword must refer to a single relation. A verb phrase like *use* that refers to a plethora of different relations could, in this way, hardly be canonicalized, regardless of whether we used a relation vocabulary-based or a clustering-based approach. For such cases, the context of the sentence, and thus, supervision is necessary to extract the underlying relation reliably.

8 Conclusion

In this paper, we have studied nearly unsupervised extraction workflows for a practical application in digital libraries. We focused on three different domains to generalize our findings, namely the encyclopedia Wikipedia, Pharmacy, and Political Sciences. First, the scalability of the investigated methods was acceptable for our partners. Second, unsupervised extraction workflows required intensive cleaning and canonicalization to result in precise semantics. Thus they do not work out-of-the-box, and reliably canonicalizing OpenIE verb phrases remains an open issue because contexts are not considered by relation vocabulary/clustering methods. Although such cleaning can be exhausting, the pharmaceutical case study yielded a novel retrieval service. Such a service would not have been possible when training data must have been collected for each relation. In addition, not filtering complex object phrases can allow the construction of semi-structured knowledge bases or enrich the original texts, e.g., show all actions of Albert Einstein. In conclusion, unsupervised extraction workflows are worth studying in digital libraries, even if, the library contains non-English texts. Those workflows come with limitations and require cleaning, but they entirely bypass the lack of training data in the extraction phase.

Supplementary information

The code of the extraction toolbox and the case study can be found in our GitHub repository.[15] An archived version can be found in the Software Heritage.[16]

Acknowledgements Supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): PubPharm - Specialized Information Service for Pharmacy (Gepris 267140244). We would also like to thank Pollux—Specialized Information Service for Political Science for providing the data for our case study, and Wolfgang Otto (GESIS) for supporting our evaluation.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Attardi, G.: Wikiextractor. https://github.com/attardi/wikiextractor (2015)
2. Auer, S., Bizer, C., Kobilarov, G., et al.: Dbpedia: A nucleus for a web of open data. In: The Semantic Web, pp 722–735. Springer Berlin Heidelberg, (2007). https://doi.org/10.1007/978-3-540-76298-0_52
3. Bhardwaj, S., Aggarwal, S., Mausam, M.: CaRB: A crowdsourced benchmark for open IE. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp 6262–6267. (2019).https://doi.org/10.18653/v1/D19-1651
4. Blasi, D., Anastasopoulos, A., Neubig, G.: Systematic inequalities in language technology performance across the world's languages. In: Proceedings of the 60th Annual Meeting of the ACL, pp 5486–5505. (2022). https://doi.org/10.18653/v1/2022.acl-long.376
5. Devlin, J., Chang, M.W., Lee, K., et al.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of

[15] https://github.com/HermannKroll/KGExtractionToolbox.

[16] https://archive.softwareheritage.org/swh:1:dir:5b575ac043e2bd61999250564a16a220c88ee5c9.

the ACL: Human Language Technologies, vol. 1, pp. 4171–4186 (2019). https://doi.org/10.18653/v1/N19-1423 https://doi.org/10.18653/v1/N19-1423
6. Gashteovski, K., Yu, M., Kotnis, B., et al.: BenchIE: A framework for multi-faceted fact-based open information extraction evaluation. In: Proceedings of the 60th Annual Meeting of the ACL, pp 4472–4490, (2022). https://doi.org/10.18653/v1/2022.acl-long.307
7. Groth, P., Lauruhn, M., Scerri, A., et al.: Open information extraction on scientific text: An evaluation. In: Proceedings of the 27th International Conference on Computational Linguistics, pp 3414–3423, (2018). https://aclanthology.org/C18-1289
8. Hristovski, D., Kastrin, A., Dinevski, D., et al.: Constructing a graph database for semantic literature-based discovery. Stud. Health Technol. Inform. **216**, 1094 (2015)
9. Jaradeh, M.Y., Oelen, A., Farfar, K.E., et al.: Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge. In: Proceedings of the 10th International Conference on Knowledge Capture. ACM, K-CAP '19, pp. 243-246. (2019). https://doi.org/10.1145/3360901.3364435
10. Kilicoglu, H., Shin, D., Fiszman, M., et al.: SemMedDB: a PubMed-scale repository of biomedical semantic predications. Bioinformatics **28**(23), 3158–3160 (2012). https://doi.org/10.1093/bioinformatics/bts591
11. Kolluru, K., Adlakha, V., Aggarwal, S., et al.: Openie6: iterative grid labeling and coordination analysis for open information extraction. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, pp. 3748–3761. (2020). https://doi.org/10.18653/v1/2020.emnlp-main.306
12. Kolluru, K., Mohammed, M., Mittal, S., et al.: Alignment-augmented consistent translation for multilingual open information extraction. In: Proceedings of the 60th Annual Meeting of the ACL, pp 2502–2517. (2022). https://doi.org/10.18653/v1/2022.acl-long.179
13. Kroll, H., Kalo, J.C., Nagel, D., et al.: Context-compatible information fusion for scientific knowledge graphs. In: Digital Libraries for Open Knowledge. Springer International Publishing, pp. 33–47. (2020). https://doi.org/10.1007/978-3-030-54956-5_3
14. Kroll, H., Al-Chaar, J., Balke, W.: Open information extraction in digital libraries: Current challenges and open research questions. In: Proceedings of the Workshop on Digital Infrastructures for Scholarly Content Objects (DISCO) co-located JCDL 2021, CEUR Workshop Proceedings, vol. 2976. CEUR-WS.org, pp. 14–18. (2021a). http://ceur-ws.org/Vol-2976/short-1.pdf
15. Kroll, H., Pirklbauer, J., Balke, W.: A toolbox for the nearly-unsupervised construction of digital library knowledge graphs. In: ACM/IEEE Joint Conference on Digital Libraries, JCDL 2021. IEEE, pp. 21–30. (2021b). https://doi.org/10.1109/JCDL52503.2021.00014
16. Kroll, H., Pirklbauer, J., Kalo, J., et al.: Narrative query graphs for entity-interaction-aware document retrieval. In: Towards Open and Trustworthy Digital Societies - 23rd International Conference on Asia-Pacific Digital Libraries, ICADL 2021., Lecture Notes in Computer Science, Vol 13133. Springer, pp. 80–95. (2021c). https://doi.org/10.1007/978-3-030-91669-5_7
17. Kroll, H., Pirklbauer, J., Plötzky, F., et al.: A library perspective on nearly-unsupervised information extraction workflows in digital libraries. In: Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries. ACM, JCDL '22, (2022a). https://doi.org/10.1145/3529372.3530924
18. Kroll, H., Plötzky, F., Pirklbauer, J., et al.: What a publication tells you-benefits of narrative information access in digital libraries. In: Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries. ACM, JCDL '22, (2022b). https://doi.org/10.1145/3529372.3530928
19. Kroll, H., Pirklbauer, J., Kalo, J.C., et al.: A discovery system for narrative query graphs: entity-interaction-aware document retrieval. Int. J. Digit. Libr. (2023). https://doi.org/10.1007/s00799-023-00356-3
20. Kruiper, R., Vincent, J., Chen-Burger, J., et al.: In layman's terms: semi-open relation extraction from scientific texts. In: Proceedings of the 58th Annual Meeting of the ACL, pp. 1489–1500. (2020). https://doi.org/10.18653/v1/2020.acl-main.137
21. Lee, J., Yoon, W., Kim, S., et al.: BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2019). https://doi.org/10.1093/bioinformatics/btz682
22. Liu, Y., Bai, K., Mitra, P., et al.: Tableseer: automatic table metadata extraction and searching in digital libraries. In: Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries. ACM, JCDL '07, p 91-100, (2007). https://doi.org/10.1145/1255175.1255193
23. Manning, C.D., Surdeanu, M., Bauer, J., et al.: The stanford corenlp natural language processing toolkit. In: Proceedings of the 52nd Annual Meeting of the ACL, ACL 2014. ACL, pp 55–60, (2014). https://doi.org/10.3115/v1/p14-5010
24. Mendez, D., Gaulton, A., Bento, A.P., et al.: ChEMBL: towards direct deposition of bioassay data. Nucleic Acids Res. **47**(D1), D930–D940 (2018). https://doi.org/10.1093/nar/gky1075
25. Mikolov, T., Chen, K., Corrado, G., et al.: Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings, (2013). http://arxiv.org/abs/1301.3781
26. Niklaus, C., Cetto, M., Freitas, A., et al.: A survey on open information extraction. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 3866–3878. (2018). https://aclanthology.org/C18-1326
27. Qi, P., Zhang, Y., Zhang, Y., et al.: Stanza: A python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the ACL: System Demonstrations, pp. 101–108. (2020). https://doi.org/10.18653/v1/2020.acl-demos.14
28. Sai, STYS., Chakraborty, P., Dutta, S., et al.: Joint entity and relation extraction from scientific documents: Role of linguistic information and entity types. In: Proceedings of the 2nd Workshop on EEKE co-located with JCDL 2021, CEUR Workshop Proceedings, vol 3004. CEUR-WS.org, pp. 15–19. (2021). http://ceur-ws.org/Vol-3004/paper2.pdf
29. Schardelmann, T., Otto, W.: Pollux - von der bedarfsanalyse zur technischen umsetzung. Bibliotheksdienst **52**(3–4), 225–234 (2018). https://doi.org/10.1515/bd-2018-0029
30. Thilakaratne, M., Falkner, K., Atapattu, T.: Information Extraction in Digital Libraries: First Steps towards Portability of LBD Workflow, ACM, pp. 345-348. (2020). https://doi.org/10.1145/3383583.3398607
31. Vashishth, S., Jain, P., Talukdar, P.: Cesi: Canonicalizing open knowledge bases using embeddings and side information. In: Proceedings of the 2018 World Wide Web Conference. WWW S. Committee, WWW '18, pp. 1317-1327. (2018). https://doi.org/10.1145/3178876.3186030
32. Vrandecic, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014). https://doi.org/10.1145/2629489
33. Wei, C., Kao, H., Lu, Z.: Pubtator: a web-based text mining tool for assisting biocuration. Nucleic Acids Res. **41**(W1), 518–522 (2013). https://doi.org/10.1093/nar/gkt441
34. Wei, C., Allot, A., Leaman, R., et al.: Pubtator central: automated concept annotation for biomedical full text articles. Nucleic Acids Res. 47(W1):W587–W593. (2019). https://doi.org/10.1093/nar/gkz389

35. Weikum, G., Dong, X.L., Razniewski, S., et al.: Machine knowledge: creation and curation of comprehensive knowledge bases. Foundations and Trends in Databases (2021). https://doi.org/10.1561/1900000064
36. Williams, K., Wu, J., Wu, Z., et al.: Information extraction for scholarly digital libraries. In: Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries. ACM, JCDL '16, pp. 287-288. (2016). https://doi.org/10.1145/2910896.2925430
37. Zhang, R., Cairelli, M.J., Fiszman, M., et al.: Using semantic predications to uncover drug-drug interactions in clinical data. J. Biomed. Inform. **49**, 134–147 (2014). https://doi.org/10.1016/j.jbi.2014.01.004
38. Zhang, Y., Chen, Q., Yang, Z., et al.: Biowordvec, improving biomedical word embeddings with subword information and mesh. Sci. Data **6**(1), 1–9 (2019)

B.8. JCDL 2023: Enriching Simple Keyword Queries for Domain-Aware Narrative Retrieval

JCDL'23

Hermann Kroll, Christin Katharina Kreutz, Pascal Sackhoff, and Wolf-Tilo Balke. "Enriching Simple Keyword Queries for Domain-Aware Narrative Retrieval". ACM/IEEE Joint Conference on Digital Libraries (JCDL) Santa Fe, NM, USA, 2023, IEEE. DOI: https://doi.org/10.1109/JCDL57899.2023.00029 arXiv: https://doi.org/10.48550/arXiv.2304.07604

Enriching Simple Keyword Queries for Domain-Aware Narrative Retrieval

Hermann Kroll
kroll@ifis.cs.tu-bs.de
Institute for Information Systems, TU Braunschweig
Braunschweig, Lower Saxony, Germany

Christin Katharina Kreutz
christin.kreutz@th-koeln.de
TH Köln - University of Applied Sciences
Cologne, North Rhine-Westphalia, Germany

Pascal Sackhoff
p.sackhoff@tu-bs.de
Institute for Information Systems, TU Braunschweig
Braunschweig, Lower Saxony, Germany

Wolf-Tilo Balke
balke@ifis.cs.tu-bs.de
Institute for Information Systems, TU Braunschweig
Braunschweig, Lower Saxony, Germany

ABSTRACT

Providing effective access paths to content is a key task in digital libraries. Oftentimes, such access paths are realized through advanced query languages, which, on the one hand, users may find challenging to learn or use, and on the other, requires libraries to convert their content into a high quality structured representation. As a remedy, narrative information access proposes to query library content through structured patterns directly, to ensure validity and coherence of information. However, users still find it challenging to express their information needs in such patterns. Therefore, this work bridges the gap by introducing a method that deduces patterns from keyword searches. Moreover, our user studies with participants from the biomedical domain show their acceptance of our prototypical system.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Users and interactive retrieval**; *Digital libraries and archives.*

KEYWORDS

Narrative Retrieval, Keyword Search, Digital Libraries

1 INTRODUCTION

Digital libraries maintain extensive collections of scientific literature and make them accessible for a variety of uses. For search, generally simple yet intuitive keyword-based access paths are implemented, see, e.g., PubMed, Google Scholar, or dblp [29]. However, while such access paths are easy to use and relatively cheap to implement, users may benefit from more advanced access paths. Here, using simple keyword-based search is oftentimes insufficient due to the limited expressiveness and the considerable amount of domain knowledge required to focus and/or refine searches [20].

As a remedy, user experience and retrieval quality can be severely boosted by advanced access paths paired with thorough semantic enrichment of digital library objects [31]. In particular, this means to annotate publications with domain-specific concepts and connecting relations. In this line, scientific knowledge bases (KBs) provide innovative ways to access literature using advanced query types like navigational queries or graph patterns, e.g., KBs in [4, 11, 17, 33]. While navigational and exploratory queries are beneficial in practice, they require users to be proficient in complex query languages like SQL or SPARQL. As a remedy, keyword search on databases and knowledge bases has been proposed [9, 14, 30, 43]. However, it is still challenging for digital libraries to convert their content into such structured representations with an acceptable quality, e.g., designing reliable extraction workflows.

A different approach, the so-called narrative information access [25], allows users to formulate their information needs as short narratives (stories) of interest – involving relevant concepts and their interactions. The main advantage is that narrative information access puts again textual publications of digital libraries in its focus. An example of such a querying mechanism are the so-called narrative query graphs, basically directed edge-labeled graph patterns [23]. Different from other knowledge base approaches, these patterns are matched against single publications instead of a single knowledge base to preserve validity and coherence of information [21]. So, users can precisely retrieve suitable publications, and additionally, generate structured overviews of the literature. However, a query log analysis of their system revealed that formulating pattern-like queries is already challenging for users; See Sect. 2. Our overall goal is to allow users to formulate their information needs as intuitively and easily as possible, i.e., as keyword queries, while providing a system capable of satisfying their possibly complex demands. Consequently, this work deals with the following research question **RQ**: *Can a user's search intent be deduced from a keyword query?*

In this work, we strive to better connect users' actual information expression strategies with narrative information access, i.e., our proposed method translates keyword to narrative queries. Unfortunately, deducing narrative queries from keywords might be ambiguous. We therefore propose a feedback loop: Users state keywords, we generate narrative queries, selection strategies select the best query options concerning different criteria, and the best queries are visualized for the users to choose from. However, integrating users into the process requires a suitable query representation, so that they can assess the generated patterns quickly and easily. We therefore perform user studies to answer the following questions: *Q1. How should generated patterns be presented to the users, i.e., which query representation is suitable for our users? Q2. How useful is the end-to-end system?*

To answer the first question, we interviewed domain experts and asked them to complete a qualitative questionnaire. Still, one may ask whether our suggested workflow (keyword entering + graph generation + user selection) is suitable to support users. Thus we

asked experts to utilize our prototypical system before we interviewed them to understand the usefulness and suitability of the end-to-end system. But, *how effectively does our method translate keyword-based queries to narrative queries for users (Q3)?* So, we analyzed our method's quality on biomedical retrieval benchmarks to better generalize our findings, involving abstracts and full-texts, highly specific and more general queries, and natural language questions. Our contribution is thus an effective digital library system that is accepted by users in our domain.

2 RELATED WORK

PubPharm's Narrative Discovery System. In PubPharm, the German specialized information service for Pharmacy, we have implemented narrative information access since 2021[1] [22, 25]. Our system allows users to formulate their information need as a graph query which is then matched against graph representations of biomedical articles [23]. In a pre-processing step, biomedical articles are converted into graph representations by identifying biomedical concepts and their relationships. Users can then search via a list of statements (basically concept-interactions) and the system replies with matching documents that contain the searched statements.

However, a query log analysis from 2021 and 2022 revealed that only 440 of 7268 queries contained more than a single statement (concept interaction). This means that users refrain from formulating complex queries. Discussions with our users later revealed difficulties in formulating more complex query patterns. The triple-based query construction seemed not intuitive enough. In this work, we built upon our narrative retrieval system [24] by simplifying users' interaction with the system while keeping its expressiveness.

Graph-based Retrieval. Dietz et al. proposed the usage of knowledge graphs for text-centric information retrieval [8]. They discussed how entities, graph structures, and relations might be incorporated to boost retrieval quality. Another work discussed how open relation extraction might accelerate passage retrieval for given entities in queries [18]. Although both works are related to ours, they rather suggest features and first evaluations instead of implementing a complete system. Krause [19] developed a graph-based retrieval system making texts more accessible which differs from our user-focus.

Pattern Mining. In general, pattern mining aims to find useful patterns in data, e.g., association rule mining discovers rules from existing data. Mining rules in knowledge bases then allows to infer new facts (complete KBs), or finding errors [12]. Fang et al. [10] produce interesting explanations for connections between entity pairs in KBs by constrained graph patterns and path enumeration algorithms. Saleh and Pecina [38] propose query expansion for cross-lingual medical information retrieval while focusing on the vocabulary mismatch problem. In contrast, our work is focused on deducing narrative queries from keywords, visualizing them for users, and letting users search with them.

Keyword Queries on Knowledge Bases. Searching KBs with keywords has already been explored [14, 30]. Existing approaches [5, 6, 15] usually work as follows: *i)* map the keywords to structured data elements, *ii)* connect the keywords by searching for substructures and *iii)* rank the retrieved substructures via a scoring function. Gkirtzou et al. [14] proposed keyword-based searches on RDF-type data sources. Elbassuoni and Blanco [9] use a backtracking algorithm to construct RDF subgraphs from keyword queries to retrieve information from RDF graphs. All triples in a KB are treated as documents where the components (subject, predicate, object) represent its content. Maximum subgraphs are constructed by retrieving documents for each query keyword (producing #keywords lists) and merging triples from different lists as subgraphs. They re-rank these subgraphs with statistical language models. Zenz et al. [43] propose QUICK, an RDF schema-based approach. While disregarding the actual keywords, in a first step they construct query templates from one-edged templates and recursively extend them by new edges. The second step associates the keywords from the query to the properties, classes and concepts from the templates.

While keyword search on knowledge bases is related to our work, the main difference is however, that the previous works in this domain assume a single KB with a known schema. In contrast, our data model includes millions of small documents graphs which allows a new definition of support, i.e., we can estimate how many graphs support a generated query.

Natural Language Queries. Another area of research is based on directly stating queries in natural language and automatically translating them into query languages like SQL (known as NL2SQL). Affolter et al. [1] present a survey comparing different textual query to database approaches and categorize them into keyword-, pattern-, parsing- and grammar-based, depending on their underlying methodology. Gkini et al. [13] study text-to-SQL approaches from a performance point of view with their benchmarking system. Liang et al. [30] propose an end-to-end BERT-based model to transfer natural language queries into subject-relation-object triples. They also jointly learn the auxiliary tasks of output variable selection, query type classification and ordinal constraint detection. Revanth et al. [35] transform natural language expressions in English to SQL queries by using NLP methods: lexical analysis, syntactic analysis, semantic analysis and transform the outputs. ChatGPT is one of the most recent methods.

The main limitation of these approaches is that they require training data to learn the actual translation. In practice, this can be an issue for digital libraries as typically not enough training data (natural language queries and their ideal translation) is available. That is why we decided to design an unsupervised translation method that does not require training data. To evaluate our algorithm on natural language questions, we selected a suitable biomedical benchmark.

Query Visualization in Digital Libraries. Keyword-based retrieval systems have been well established for scientific information needs, e.g., PubMed, Google Scholar, Scopus, or dblp [29]. In brief, these systems typically capture the user's keywords in some text input field and display results as a list to the user. When systems utilize semantic search techniques through machine learning, these systems typically boost the retrieval quality [31, 39]. However, they usually do not visualize what is happening with the query to the users, e.g., Semantic Scholar [3]. Knowledge bases like Wikidata [41] or the Open Research Knowledge Graph [17] provide the users either with entity-centric interfaces to click and navigate through the knowledge, or with SPARQL endpoints requiring users to learn SPARQL for posing queries. In contrast, we enrich keyword queries and present derived queries in a feedback loop.

[1] http://narrative.pubpharm.de

3 QUERY MODEL AND RETRIEVAL

In our work [23], we defined narrative query graphs as directed edge-labeled graphs with concepts as nodes and relationships as edges between them. However, we observed two major drawbacks of our system: First, users may not know how a certain concept should be connected to a query pattern. Suppose a user search for case-based studies in connection with Metformin diabetes treatments. In this case, a `treats` relationship can be placed between the concepts `Metformin` and `diabetes`. But to which concept should the concept `case-based studies` at best be connected? Second, some users faced the out-of-vocabulary problem, i.e., they wanted to search for concepts that were not known in the system. In this work we adjust our previous query model to tackle both drawbacks: In addition to graph patterns, we allow queries to also search for concepts that are not connected and for terms to tackle the out-of-vocabulary problem.

Formally, we denote C as the set of known concepts. Each **concept** c is identified by an identifier, e.g., $c_{Metformin} = (CHEMBL1431)$. Concepts are usually collected and arranged in domain-specific taxonomies or ontologies, e.g., the Medical Subject Headings[2]. Some concepts may be arranged in a **subconcept** relation, e.g., *Diabetes Mellitus Type 1* is sub concept of the super concept *Diabetes Mellitus*. We denote relationships between two concepts by $\mathcal{R}$ – the set of predicates (also known as relationship labels), e.g., `associated` or `treats`. Those predicates might be very general like `associated` or could be more specific like `treats`. In Wikidata for example, predicates are understood as resources/items that can be arranged in an hierarchy. Note that some domains might not arrange their predicates in this way. With that, we can define a **statement** as a triple, e.g., $(c_{Metformin}, \texttt{treats}, c_{Diabetes\ Mellitus})$. Next, we define the set of statements as $Statements \subseteq C \times \mathcal{R} \times C$. Note that our definition of statements is similar to the representation of knowledge in the Resource Description Framework (RDF) [32].

The retrieval system returns documents as results. Therefore, we denote $\mathcal{D}$ as the set of documents. Our documents $d \in \mathcal{D}$ consist of texts and each text consists of terms (single words/tokens). $\mathcal{T}$ is the set of all terms and $t \in \mathcal{T}$ is some term. For the actual retrieval, we need to know the terms of a document, which concepts have been detected in it, and which statements were extracted from it.

(1) $doc\_terms(d) = \{t \in \mathcal{T} \mid t \text{ is term in } d\}$
(2) $doc\_concepts(d) = \{c \in C \mid c \text{ detected in } d\}$
(3) $doc\_stmts(d) = \{s \in Statements \mid s \text{ extracted from } d\}$

Finally, we define a **narrative query** $nq = (Q_S, Q_C, Q_T)$ with $Q_S \subseteq Statements$, $Q_C \subseteq C$ and $Q_T \subseteq \mathcal{T}$. In other words, a query may ask for statements, concepts, and terms. We call d a **match** with regard to $nq = (Q_S, Q_C, Q_T)$, iff the following conditions hold: 1. $Q_S \subseteq doc\_stmts(d)$, 2. $Q_C \subseteq doc\_concepts(d)$, 3. $Q_T \subseteq doc\_terms(d)$. The set of all document matches regarding a query nq can then be defined as $answers(nq) = \{d \in \mathcal{D} \mid d \text{ is match to } nq\}$.

Now we define the **query translation task** as:

Given a keyword query $q = (w_1, \ldots, w_n)$ with terms w_i, find all narrative queries such that each term of q is mapped to some query term, query concept or query statement. In other words, all words of keyword query q must be reflected in some way.

[2] http://meshb.nlm.nih.gov

4 KEYWORDS TO NARRATIVE QUERIES

In this section, we present our algorithm to deduce narrative queries from keywords. Given a keyword query, our goal is therefore to first generate **all** possible narrative queries, i.e., all combinations that can be derived. In a second step, the **best** queries concerning different criteria are selected to be shown to the users. However, generating all possible narrative queries from keywords could yield a plethora of queries as keywords might refer to concepts, predicates, or, even worse, be synonymous with a set of concepts. And then, we still would have to place predicates between those concepts to derive statements. We first introduce suitable lookup indexes to minimize the generation space. For simplicity, we call a query's terms, concepts, and statements **query components**.

We utilize a **document collection index** that retrieves **support** of query components, i.e., how many documents in our collection include the corresponding component. Possible query components with low support (e.g., no documents) could be disregarded because queries with those components will yield empty (or fewer) results in the end. We design this index as an inverted index. Given the functions *doc_terms*, *doc_concepts*, and *doc_stmts*, we iterate over all documents of the collection, use the functions to retrieve the required entries and finally build this index.

The set of predicates $\mathcal{R}$ is typically known, either it is known during the information extraction from texts, or it can be derived by iterating over all statements. In brief, we design a **predicate index** that maps labels to predicates. Suppose a user may enter a keyword query such as: `Metformin therapy Diabetes Mellitus`. In that case, it would be beneficial to detect that the keyword `therapy` refers to a `treats` predicate. In practical knowledge bases, predicates are usually described by human-readable labels and sometimes by a list of synonyms, e.g., Wikidata includes a list of *also known as* labels (Property P2175). If such information is available, we can additionally incorporate it in our predicate index.

Our primary goal is the generation of narrative queries with concepts and their statements, i.e., we need to map keywords to concepts. To do so, each concept of C should, at best, be described by a human-readable label and a list of synonyms (e.g., Wikidata's *also known as* labels). To align keywords with concepts, we utilize those labels (label + synonyms) to compute an **concept index** that maps labels to concepts. In practice, homonyms might exist, i.e., a label could refer to a set of concepts and not only to a single one. In that case, the subsequent step generates different queries – at least one for each homonymous concepts.

4.1 Generating Narrative Queries

In general, one could assume a certain order of keywords, i.e., a subsequent list of keywords is an information unit and corresponds to a concept. However, users might extend or refine keyword queries which may break such an order. Our algorithm provides the option to consider all keyword permutations when mapping. Another point to think about is that users may query with arbitrary, information-sparse keywords such as *of* or *in*. So, we provide the option to ignore stopwords in users' queries. Our algorithm operates as follows:

1. Mapping Phase. The first step takes a list of keywords (the user's keyword query) as its input. We optionally remove stopwords

and then tokenize those keywords into tokens. This step yields all possible mappings from keywords to concepts, predicates, and terms. We iterate over all tokens in the keyword query and check if a token maps to a concept. Note that a concept/predicate label might also consist of multiple tokens. Thus we also need to check combinations of keywords in this mapping phase. By default we assume that the tokens in a query follow a certain order, and thus, only check combinations of subsequent tokens, e.g., in *Metformin treats Diabetes Mellitus* we would check the following combinations *Mellitus*, *Diabetes Mellitus*, *treats Diabetes Mellitus*, *Metformin treats Diabetes Mellitus* but not check the combination *Metformin Diabetes Mellitus*. The non-default option does consider permutations, i.e., also *Metformin Diabetes Mellitus*.

Next our document collection index comes into play. For this check, we introduce τ as a threshold parameter (default 0). We remove all mappings to concepts having a support below τ, i.e., each concept must at least be included in more than τ documents. Having our concept mappings, we compute possible statements by iterating over all concept combinations (c_i, c_j). An inner loop iterates over all predicates $\mathcal{R}$ with the variable p. We then test whether each statement (c_i, p, c_j) has support $> \tau$ in our collection index. If yes, we keep the possible statement, i.e., we store it in a list. If not, we ignore the statement because it is not supported enough and will thus yield too few or no documents. The last step is to map tokens to possible predicates if the predicate label or one of its synonyms matches the token (or multiple tokens). If so, we add the token to the predicate mapping. Similarly, predicate labels may consist of multiple terms, so we also check token combinations here as done for the concept mappings. Finally, this step yields 1) mappings from query tokens to concepts and to predicates, and 2) a list of possible statements.

2. Generation Phase. The second step takes the query tokens, both mappings (concept and predicate), and a list of possible statements as input. The central strategy for this step is to map all tokens unambiguously to components of a narrative query. In other words, for each generated query, we must decide what we do with a token, i.e., whether we map it to a concept, a term, or a predicate, but not to multiple ones at the same time. Furthermore, for each decision, we might have multiple options. The algorithm works as follows:

1. Map tokens to concepts and predicates. If a token can be mapped to multiple concepts, generate a combination for each one. Also include the option not to map a token. This allows term-based only queries and every combination in between.

2. Select Mappings. For each combination from the previous step, generate a query. In this query, include the targets of the mappings (concepts and predicates). Also include those tokens that have not been mapped yet, as terms. Only keep those terms that have support $> \tau$. With this, we ensure that all tokens are mapped somehow, except token to term mappings that would yield too few document result.

3. Integrate Statements. For each query, we could decide which statements we include. Again, we have to compute all combinations here. That is why we compute a sub-list of statements from all possible statements (previous step) that applies to this query (the query must include the statement's concepts as concepts). Note, that we only allow putting a single predicate between two concepts. Then, compute all combinations (include a statement, do not include a statement). Again, generate queries for all different possibilities.

4. Filter the generated queries with the following rule: If we map a keyword to a predicate, we must include a corresponding statement with that predicate in this query. If no corresponding statement is included, the query is removed.

Furthermore, we only allow putting a single predicate between two concepts because each query should represent a specific information need. If several predicates are possible, our algorithm generates them as different queries. Note that checking all combinations generates a query where each keyword is mapped to a term. We finally return a list of narrative queries. Note that the selection of τ will affect the overall exploration space since a low value forces our algorithm to generate more queries than a large value. However, for this paper, our goal was to generate all possible queries, i.e., we set $\tau = 0$. Including concepts, statements, or terms with a support $\leq \tau$ will not be helpful because all narrative queries asking for them will yield no or less results. Via our feedback loop in the user interface, we show which tokens have been excluded. As an alternative, a system could include those tokens, return empty results, and force the users to refine their queries.

4.2 Query Selection Strategies

In the following, we introduce strategies to select the *best* queries concerning different criteria. In brief, all strategies have to balance specificity and generality. We, therefore, design three strategies that we further analyze in this paper: a general one aiming for recall, a mixed one aiming for F1, and a specific one aiming for precision.

The following two strategies should prefer queries with statements. That is why both strategies filter out all queries that do not contain a statement. Queries with statements can get very specific and may likely not yield any document results. Due to our focus on users, each selected query should at least return some results. So, both strategies rank the queries with statements and only keep queries that yield at least a single result. A predicate hierarchy may arrange predicates (see Sect. 3), e.g., `treats` is more specific than `associated`, or `inhibits` is more specific than `induces`. In our biomedical use case, *associated* is the most general predicate, and every other predicate is a specialization. In Wikidata, for instance, the *Wikibase property* (Item Q29934218) could be seen as a very general predicate.

That is why we designed the following two strategies: The **mixed strategy** allows all predicates in queries. And the **specific strategy** forces queries to include specific predicates, i.e., prefers queries with more specialized predicates, e.g., prefers `treats` over `associated`. If predicates are not arranged in a hierarchy, the strategies select the same queries. As our motivation for these two strategies we assumed that selecting very specific predicates will likely boost the precision, but reduce the recall. A more general predicate might be a good mix between precision (because we force a statement) and recall (we do not force a to specific one). We still have to weigh the number of statements and the number of returned results. Due to our focus on users, we decided to rank the remaining queries by the number of results so that users can expect a fair number. Our last strategy focuses on recall. **The most-supported strategy** executes all generated queries and ranks the queries by the number

Figure 1: Prototypical user interface with markings. 1. highlights the search slit, 2. gives the query variants which users can choose from, 3. indicates the results of the query variant.

of returned results in descending order, i.e., prefer queries that yield more documents than other queries. Then the best ranked query is yielded. This strategy usually prefers term/concept-only queries because the number of hits is likely higher.

5 SYSTEM IMPLEMENTATION

We have already described implementation details for our narrative retrieval system in [23, 25]. In the following, we implemented our algorithm and extended our previous retrieval system. The following numbers are based on a snapshot of the system from December 2022. The system worked on the whole biomedical National Library of Medicine (PubMed) Medline.

We retrieved 35M documents (titles + abstracts), 711M concept annotations, and 842M extracted statements. The concepts stem from existing ontologies: the Medical Subject Headings, the ChEMBL database, and Wikidata [41]. The concept ontology had a root node (Thing) and then branched out into 13 basic concepts, e.g., drugs, diseases, targets, genes, species, etc. In sum, 635k concepts were known in the system. The retrieval system organized ten different predicates into a hierarchy, e.g., `associated` is the most general predicate, and every predicate is a specialization of it. Information about the predicates (synonyms and hierarchy) can be found at[3]. Given the concepts and predicates plus synonyms, we derived our concept and predicate index. We then used the document data to compute our document collection index.

Consider some searches for diseases. In that case, all documents should support a `disease` concept if the disease concept or one of its subconcepts can be found. We materialized, therefore, the concept ontology like suggested in [26], i.e., if a certain concept was found in a document, all super-concepts could also be found in that document. The same rule applies to statements: Suppose the statement (s, p, o) was extracted from some document. In that case, we also store the same relation p between all super-concept combinations of s and o. In addition to that, we also store all statements with more general predicates, i.e., if a document contains a `treats` statement, it also implies a corresponding `associated` statement.

Next, we computed a case-insensitive inverted term index for those documents. To remove stopwords, we used the English NLTK stopword list [7]. Therefore we iterated over the documents' contents, split the text by a space character, removed stopwords, and used the remaining tokens for indexing. As an additional option we repeated that procedure but replaced all punctuation in texts with a space as biomedical concepts often contain special characters like - and +. We used the Python Punctuation set: !"#$%&'()*+,-./:;<=>?@[\]^_`{|} .. Finally, we computed the document collection index with 39M term and 635k concept, and 318M statement inverted index entries. For the query tokenizer, we removed brackets and split the keywords by a space.

Next, we implemented a prototypical user interface[4] which is depicted in Figure 1. The interface works as follows: 1. Users enter a keyword query. 2. Three generated queries were visualized for the users. 3. A user's click on one of them started a search and returned matching documents. We used our previously introduced strategies to select the three queries. If one strategy might not yield a query, e.g., we simply did not visualize it. Concerning the query visualization, we first conducted a user study which is described in the following section. The study concluded that the graph representation was most suitable for our users. We then implemented a graph representation for the second study, i.e., concepts were visualized as nodes, statements as edges between them and terms as a simple comma-separated list.

[3] http://www.narrative.pubpharm.de/help/

[4] http://narrative.pubpharm.de/keyword_search/

(a) Graph query of *'Through which target do Simvastatin and Amiodarone interact so that Simvastatin may induce a Muscular Disease?'*

Metformin administered **?X(DosageForm)**
?X(DosageForm) administered **Patients**
Patients associated **Diabetes**

(b) Structured query of *'As which dosage forms can Metformin be administered to diabetic patients?'*

Some vaccine is associated with COVID 19.
The same vaccine is administered to patients.
These patients suffer from CVST.

(c) Natural language query of *'Which COVID 19 vaccines may make patients suffer from cerebral venous sinus thrombosis (CVST)?'*

Figure 2: Query representations and information needs.

We used the same visualization for the document result lists as in the narrative retrieval service. Please note that one feature of this system is the support of variables in queries [23]. A variable asks for any possible concept that fits into the query, e.g., for all *diseases that can be treated by Metformin*. Internally, PubPharm rewrites queries that include some very general concepts like `disease`, `drug`, and `target` to aggregate the literature by suitable substations, i.e., showing one list of documents about the drug `Simvastatin` and one about `Metformin`. For our first user study, we investigate whether we should include such information in our query representations, i.e., we included variables written as $?X$.

6 USER STUDIES

In the following, we describe our user studies and study results. For convenience, the user-centric evaluation was based on: *Q1. How should generated patterns be presented to the users, i.e., which query representation is suitable for our users? Q2. How useful is the end-to-end system?*

6.1 Study I: Questionnaire and Discussion

The overall goal of this study is to gain insights on query representations' suitability for (potential) users. This qualitative user study consists of two parts, a questionnaire and a following group discussion. This study was conducted in context of an online workshop from PubPharm in which participants from the broader area of the pharmaceutical domain were introduced to PubPharm's narrative information access. Participants took part in our study voluntarily.

6.1.1 Setup - Questionnaire. All study participants were presented an English online questionnaire. It first introduced three different query representations with different exemplary information needs. We used these representations to have a mix of structured (graph), semi-structured (triple-like text statements) and natural language query representations (see Figure 2). We showed this exact order: graph, structured and natural language. The information needs were well-known examples from the biomedical domain and of the exact same structure. For each representation participants were asked if they immediately understood it on a 5-point Likert scale.

Table 1: Study participants' ratings concerning the immediate understandability (IU) of the three compared query representations (++ strongly agree, -- strongly disagree).

IU	++	+	+/-	-	--
graph	3	3	2	1	0
structured	1	4	2	2	0
natural language	3	4	2	0	0

The next part of the questionnaire showed the three different information needs and possible representations (so the nine combinations) in one single figure. Study participants were then asked to answer (or skip) free text questions intending to capture their satisfaction with and the suitability of the representations. The *questionnaire open questions* were derived from the main components of user satisfaction described in the user experience questionnaire[5] [27]:

- QQ_1 What did you like/dislike about the q. representations?
- QQ_2 Which query representation would be/not be easy-to-learn for you and why?
- QQ_3 Working with which query representation would/would not introduce unnecessary effort for you and why?
- QQ_4 With which query representations would you be interested/disinterested in working and why?

The first question QQ_1 strove to capture users' perception of *attractiveness*, their overall impression and leaned onto the question from the UEQ, if users like or dislike a product (here the representation). QQ_2 tackled *perspicuity* and leaned on the UEQ's question if it is easy to get familiar with a product and to learn how to use it. With QQ_3 we strove to observe *efficiency*. This aspect of an UEQ usually assesses, if users can solve their tasks without unnecessary effort. Lastly, QQ_4 aimed to look at *stimulation*, so if users are excited and motivated to use a product. We deliberately refrained from posing questions related to dependability and novelty of query representations in the questionnaire for time reasons. In our opinion dependability can only be assessed with actually using or constructing queries in the different query formulation. Novelty is one of the less important factors in our case, as none of our query representations are truly novel. After these aspect-based open questions, participants were given the opportunity to answer what their favorite query representation was and to explain their choice. Finally, the questionnaire asked them if they wanted to be contacted again for participating in another study on the same subject. For answering the whole questionnaire part, we gave our participants 15 minutes.

6.1.2 Results - Questionnaire: Likert Scales. Nine participants answered the Likert Scale part of the evaluation by indicating their first impression on the understandability of the three query representations (see Table 1). Study participants rated the graph representation and natural language representation similarly, the structured representation's immediate understandability was rated lower.

[5] https://www.ueq-online.org

6.1.3 Results - Questionnaire: Open Questions. Usability aspects of the three query representation variants were evaluated through open questions by seven of the nine initial participants:

QQ_1: Attractiveness. Participants indicated liked/disliked components regarding the representations: They commented the graph representation was fast to understand in general. It was taking time to be understood but was then stated as the best representation. They stated that the graph was valuable because it visualizes relations, which play an important role in pharmacy. The structured representation was mentioned to take time to understand. Two participants stated their unfamiliarity with its subject-predicate-object structure. The natural language representation was mentioned as simple by three participants. One mentioned it took time to understand while another one praised its quick understandability. One rated it as the best one. Another participant disliked its partially non-naturally sounding sentences. One participant disliked the absence of colorful highlighting in the graph representation. Another one rated all representations as unintuitive.

QQ_2: Perspicuity. Participants indicated the easiness of learning the representations: Three participants rated the graph representation as the fastest to learn. Two people mentioned it was easy to read and one mentioned it as the most complex representation. The structured query representation was called confusing and unclear by one participant. Another person commented it required some time to learn this representation. One participant chose the natural language representation as the easiest to learn. Another one labeled it as confusing and unclear. In general one participant mentioned all representations being easy if one took the time to learn them while another participant rated them all as unintuitive.

QQ_3: Efficiency. Participants indicated the level of unnecessary effort using the representations would introduce: One person rated the graph representation as easy. Another one mentioned it would be easy to learn. The structured representation was seen as needing time to be learned. A participant deemed the natural language representation familiar. Another one found it imprecise and requiring more time to formulate. Lastly, one person stated using any of the representation would not introduce unnecessary effort.

QQ_4: Stimulation. Participants indicated which representations they were interested/disinterested in using: Two mentioned the graph representation as positive while one participant rated the graph representation as the most complex one. Another person disliked the structured representation as it would need to be learned first. One participant commented that the natural language representation was the easiest to use. Another person refrained from stating preferences in representations as these representations were merely a tool to answer interesting questions.

Favorite. Five participants picked the graph representation as their favorite one and one picked the natural language representation. The graph representation's easiness and the possibility of visualizing complex interconnections were liked.

6.1.4 Setup - Group Discussions. The second part of this study were group discussions. Study participants were evenly divided into two groups (one in English, one in German) with an interviewer and a transcript writer each. The semi-structured group discussions took place directly after completing our questionnaire and took 10 minutes. We asked them three *guide questions* for the subsequent discussions:

GQ_1 What representation could you imagine to use in practice?
GQ_2 What would you change? What should be different?
GQ_3 What was your favorite query representation and why?

6.1.5 Results - Group Discussions. Six study of the original nine participants took part in our group discussion. We evenly split them into two groups. In the following, we combine the opinions of both discussions. In each group, we started with three guiding questions.

GQ_1: Practical usage. One participant argued that if one is already using other information systems, they are used to a specific way of obtaining data. As all systems are different, simplifying usage and not introducing new query languages (so natural language representation) should be the focus. All other participants were more inclined towards the graphical representation. One participant stated that natural language seemed the easiest option in the beginning but was surpassed by the graph representation, as it clearly defines what is searched for. A further participant liked the graph representation, as it was easy to understand the query, but disliked the question marks in the representations (for the variables). They mentioned that the graph clearly shows the relations. One interviewee stated the graphic representation would be the best one and all other representations would be very cumbersome. This view was shared by another study participant who additionally mentioned the graph representation would be easily practiced. The natural language representation and structured representation with multiple rows was considered hard to read by multiple study participant. However, one mentioned that the boldly marked concepts would be helpful to an extent.

GQ_2: Desired changes. One participant considered the graph representation as not being self-explanatory and required the textual description of the information need to understand the representation. Another interviewee mentioned that users would need good examples to adapt them to their personal information needs. They further stated that with this help even complex graphs with more concepts could be constructed. Someone suggested that arrows in the graph representation should be different from each other to convey information on the type of relation (e.g., distinguishing between a *treats* and an *inhibits* relation). A study participant mentioned that (colorful) highlighting searched concepts could be helpful. Another one explicitly disliked having color in the graph representation as it would overload the query representation.

GQ_3: Favorite. One of six participants preferred the natural language representation as it did not require a user to learn a new query language. The remaining five participants preferred the graph representation because it would clearly highlight the connections between concepts and it would be easy to grasp.

6.2 Study II: Thinking-Aloud and Interview

In the following, we analyze the usefulness of the end-to-end system for potential users. Our goal is to capture users' perspectives when deciding on a query pattern, their query formulation strategy, overall impression and problems with our prototypical system. We therefore conducted a second user study fully online. It consisted of a thinking-aloud [28] exploration phase of our system (see Figure 1)

and a semi-structured interview. The sessions were conducted individually for each participant in the presence of two investigators. In total the study took about 30 minutes per participant. Ten participants took part (nine in German, one in English).

We implemented the graph representation to visualize the queries as it was found to be the most suitable representation in our first user study. As a small difference for better understandability, we adjusted the variable representation by removing the leading question mark and variable name, e.g., we replaced ?X(Drug) by Drug.

6.2.1 Participants. Our acquired participants were experts in the broader pharmaceutical domain and interested in the usage of domain-specific bibliographic information systems. Those participants were researchers in the pharmaceutical domain in different positions (PhD students, postdocs, and professors). They took part in this study voluntarily and were explicitly made aware that they could refrain from taking part any time.

6.2.2 Setup - Thinking-Aloud Exploration. Before starting with the exploration, an investigator introduced the system (see Figure 1) by showing a screenshot with instructions on how to use the system: 1. Enter a query, 2. Click on search, 3. Select a generated query, and 4. Explore the result list.

Participants fulfilled their own information needs from the domain with our prototypical system. We gave them the guiding question: *Think about the topic you are currently working on in the pharmaceutical domain. Which questions would you typically pose to PubMed or the keyword-based interface of PubPharm?* Participants were asked to describe their thoughts when interacting with the prototype system [28]. This step took 20 minutes at max.

6.2.3 Setup - Semi-Structured Interview. After the thinking-aloud exploration, semi-structured interviews with each participant tried to capture users' perspectives regarding the usefulness of the system. We used the following guide questions:

- What are your general thoughts regarding the system?
- Where did you encounter problems? What was unclear?
- What did you like/immediately understand?
- Which changes would make you consider using t. system?
- Anything else you want to add or ask?

6.2.4 Results. This section discusses the observations, encountered problems and suggestions from the thinking-aloud exploration and the semi-structured interviews conjointly.

In general, there were a lot of *'This is what I meant'*-moments when graph patterns were generated for keyword queries. Displayed graph patterns were described to be immediately understood by participants. The documents were found to be relevant when clicking on one of the graph patterns. Especially the combination of variables (e.g. diseases or targets) with concrete agents generated query patterns that were directly grasped by users.

Users were very confident with choosing graph patterns fitting their query. Once they pick out a graph pattern, they do not select another one to compare the results. Out of all participants and queries only once a second graph pattern fitting the same keyword query was also chosen. Beside the positive feedback, we found four core elements which lead to problems for most of the participants in varying degrees:

Entering queries. For the query formulation, users suggested a prefix-based suggestion of concepts (autocompletion), support of multi part terms indicated by quotes and a spell correction of terms, e.g. entering *pharmacokinetic* yielded no results but entering *pharmacokinetics* yielded results.

Choosing a graph pattern. A graph pattern of a query should always be constructed. Study participants were confused by or displeased with query variants which only had terms or a combination of terms which were not visualized as graphs. These combinations seemed to not be intuitively understood by users. Additionally, more or better query variants should be displayed. Especially with a drug and a disease, only *treats* and *associated* were suggested as the connecting relations, while *induces* would be another viable option to suggest.

Filtering results. Filter options should be provided for users to navigate or restrict their results without having to change the query. Users seem to prefer restricting their current results opposed to writing narrower queries. Participants of our study wanted to filter out documents by the year, the article type (e.g., surveys), and keywords contained in the title.

Exploring results. The Provenance function (explains matches to users in the service) should be extended to include concepts and terms for users. Alternatively, the query part in the document content view needs to be better highlighted.

Further remarks. Multiple times study participants utilized the number of displayed results for a query to estimate if the query and the chosen graph pattern fit their information need. We therefore derive the approximate number of results being a valuable information which should be displayed in advance.

Other suggestions were: shorter loading times, graph patterns not overlapping, PubMed-like Boolean operator support in queries, visual structure search, a shopping-cart-style system to save interesting results and a direct integration of relevant results' citations/references which fit the query.

6.3 Discussion

While participants of the first user study rated both the query and natural language representation as quite understandable at first glance (see Table 1), users' comments in the open questions and the group discussions showed their overall preference of the graph representation due to its clarity, ease of learning and capability of quickly conveying information. The second study supports this choice, participants mentioned the graph representations of queries being understood immediately. Our users stated that pharmacists are usually experienced with graph representations, e.g., with visual chemical reactions or when drawing molecular structures. Our study revealed that the most suitable generated query patterns' representation for users was the graph representation. We found all ten users of our second user study intuitively being able to use our prototypical end-to-end system. Moreover, it was suitable for satisfying their individual information needs. Everyone confidently picked the graph pattern which best fit their query. However, we identified room for improvements regarding the different aspects of the system (see Section 6.2.4) which should be rectified in a further iteration of the system.

7 EVALUATION OF EFFECTIVENESS

Our user studies have demonstrated that the graph representation was suitable and the overall system was accepted. However, *how effectively does our method translate keyword-based queries to narrative queries for users (Q3)?*

Due to our restriction to the biomedical domain, we had to restrict the evaluation to biomedical information retrieval benchmarks. We picked the following ones:

(1) **TREC Precision Medicine 2020** [36] (PM2020, 31 topics) covers precision-focused retrieval of biomedical PubMed abstracts. Each query asks for three concepts: a treatment (drug), a disease, and a variant (gene/target).
(2) **TREC COVID 2020** [40] (COVID, 50 topics) bases on retrieval of COVID-relevant literature, the COVID Open Research Challenge [42]. We use the data's 5th (most recent, from 16th July '20) release. Queries ask for different topics on COVID-19, e.g., treatments, outbreaks.
(3) **TREC Genomics 2007** [16] (Genom., 36 topics) includes natural language questions around biomedical target interactions for passage retrieval in full-texts.
(4) **TripJudge** [2] (TripJ., 1136 topics) holds queries and interaction data from the Trip Database for abstract retrieval. TripJudge is an extension of the TripClick [34] by improving the quality through human annotations.

We choose these benchmarks to cover a wide range of biomedical queries, from very specific ones (PM2020) to general queries in TripJudge, up to natural language questions in Genomics. Titles and abstracts of the relevant documents were available for all benchmarks. Topics were single input queries. PM2020 and TripJudge were based on abstract retrieval. COVID could be evaluated in two ways: only abstracts and abstracts + full-texts. Genomics, in contrast, was a full-text passage retrieval benchmark, and thus, we only evaluated the full-text setting. PubMed Medline documents required for PM2020 were already included in PubPharm's narrative system. For the missing TripJudge, COVID (pre-prints), and Genomics documents (especially for the full-texts), we applied the same concept linking and information extraction, which has already been conducted for the PubMed collection.

Setup. Retrieval benchmarks typically provide judged documents, queries, and a ranking of which documents are relevant for a specific query. Usually, retrieval is evaluated by ranking documents and computing scores for different rank values k, like precision@k and recall@k. However, our query model does Boolean retrieval, i.e., a document can be relevant for a query or not, there is no ranking among relevant documents. Therefore, we had to compute the number of retrieved relevant documents and how many relevant documents were missing, i.e., we report precision, recall, and F1. For our subsequent evaluation, we follow the definition of bpref [37] and only considered documents that have been judged in those benchmarks to determine whether a hit was relevant.

We designed our evaluation as follows: First, we translated the queries of each benchmark into all possible narrative queries with our algorithm. We then executed those queries and took the ones that achieved the highest precision, recall, and F1 score for each query in every benchmark. This determined an upper bound on achievable precision, recall, and F1 with our retrieval model. As a baseline, we used a Boolean term-based retrieval model, i.e., we used the terms of the queries directly for searching documents. Our second step then analyzed our selection strategies, i.e., we counted how many cases we selected one of the best precision queries, best-recall queries, and best-F1 queries. Note that multiple narrative queries may return the same score for some keyword queries. We can quantify how effectively our selection strategies pick the best queries concerning our evaluation metrics.

Table 2: Highest-achievable retrieval quality with our query model compared to term-based retrieval.

	Metric	TermB	BestP.	BestR.	BestF1
Abstract-Only Retrieval					
PM2020	Prec.	0.48	**0.84**	0.51	0.54
	Rec.	0.24	0.06	**0.41**	0.40
	F1	0.27	0.10	0.40	**0.41**
COVID	Prec.	0.33	**0.40**	0.33	0.34
	Rec.	0.26	0.20	**0.31**	0.29
	F1	0.22	0.17	**0.24**	**0.24**
TripJ.	Prec.	0.44	**0.51**	0.44	0.47
	Rec.	0.85	0.75	**0.87**	0.85
	F1	0.53	0.50	0.53	**0.55**
Full-text Retrieval					
COVID	Prec.	0.16	**0.26**	0.16	0.18
	Rec.	0.45	0.32	**0.49**	0.44
	F1	0.18	0.14	0.19	**0.21**
Genom.	Prec.	0.23	**0.42**	0.23	0.30
	Rec.	0.23	0.11	**0.26**	0.20
	F1	0.14	0.12	0.14	**0.18**

Effectiveness of Strategies. Table 2 lists the best results when generating all narrative queries and using our query model for retrieval. The evaluation showed that our query model improved the term-based search for every benchmark and metric. For example, on TripJudge, the term-based retrieval achieved a precision of 0.44 and recall of 0.85, whereas our model boosted the precision to 0.51 with a recall of 0.75, or the recall to 0.87 by retaining the precision of 0.44. The difference was even larger for PM2020, where the precision was boosted from 0.48 to 0.84 (by decreasing the recall from 0.24 to 0.06). Here, focusing on F1 boosted it from 0.27 to 0.41. Moreover, natural language questions of Genomics were translated into narrative queries that outperformed the baseline.

Further, we analyzed how many of those best queries contained statements and, thus, fully utilized our query model. We counted the number of topics for which the best query contained at least a single statement: Concerning precision, 26 out of 31 (PM2020), 12 out of 50 (COVID on full-texts), 195 out of 1136 (TripJudge), and 15 out of 36 topics (Genomics) did. Concerning recall, 2 out of 31 (PM2020), 0 out of 50 (COVID on full-texts), 54 out of 1136 (TripJudge), and 6 out of 36 topics (Genomics) did. Concerning F1, 3 out of 31 (PM2020), 5 out of 50 (COVID on full-texts), 85 out of 1136 (TripJudge), and 12 out of 36 (Genomics) did. We expected these numbers because precision-oriented queries may rather contain

Table 3: Number of topics where a query producing the highest metric was selected by one of our strategies. 'Any' denotes where any of the best metrics queries was selected.

Benchmark	|Q|	BestP.	BestR.	BestF1	Any
Exact Query Found					
PM2020	31	4	21	10	22
COVID	50	20	40	34	40
TripJ.	1136	548	806	579	849
COVID+F	50	22	45	29	45
Genom.	36	12	23	16	25
One Allowed Edit in Terms/Concepts					
PM2020	31	12	24	20	25
COVID	50	36	46	43	46
TripJ.	1136	804	926	839	947
COVID+F	50	38	50	45	50
Genom.	36	16	30	23	30
One Allowed Edit in Predicates					
PM2020	31	16	21	10	25
COVID	50	21	40	34	41
TripJ.	1136	569	807	594	854
COVID+F	50	25	46	32	46
Genom.	36	15	24	18	27

statements than recall-oriented ones. In addition, many benchmark queries, especially in TripJudge or COVID, were relatively short, e.g., *coronavirus origin*, *coronavirus quarantine*, or *green tea*. Such keywords were not converted into statement-based queries because we only placed statements if we found at least two concepts. Here, our concept vocabulary did not include concepts for *origin* and *quarantine*. We counted how many of the best queries included at least two different concepts. Concerning the queries with the best precision, 27 out of 31 (PM2020), 15 out of 50 (COVID on full-texts), 205 out of 1136 (TripJudge), and 17 out of 36 topics (Genomics) did. To verify that this was not just based on an out-of-concept-vocabulary problem, we counted how many queries contained three or more keywords: 31 out of 31 (PM2020), 36 out of 50 (COVID), 559 out of 1136 (TripJudge), and 36 out of 36 topics (Genomics) did. This justified our assumption that many benchmark topics were relatively short, i.e., we did not find concepts and, thus, did not ask for highly complex interactions. If queries tended to get longer and more precise, like in PM2020 or Genomics, statements in queries became more relevant.

Highest Metric Queries. The evaluation demonstrated that our query model is indeed beneficial for information retrieval. However, how many of those best queries do our selection strategies find in practice? In other words, what can users expect when entering different keyword queries? To answer this, we counted how often our three strategies yield one of the best possible queries concerning an evaluation metric (best precision, etc.). Table 3 lists the results. For the 31 topics of PM2020, our strategies found the best precision queries for four topics, the best recall queries in 21 topics, the best F1 queries in ten topics, or at least one of the three best metric queries in 22 topics. For the 1136 topics of TripJudge, we found a best precision query for 548 topics, a best recall query for 806, a best F1 query for 579 topics, or at least one of the three best metric queries for 849 topics. For the 36 Genomics natural language topics, for twelve topics our strategies found of the best precision, for 23 topics the the best recall, in 16 topics the best F1 one, or at least one of the best metric queries in 25 topics. In summary, users can expect to get the best query concerning precision in 13% (PM2020) to 48% (TripJudge) of cases. For best recall, they can expect queries in 67.7% (PM2020) and 90% (COVID on full-texts) of the cases.

Allowed Edits. However, how different are our selected queries if we do not find the best possible one? We counted two cases: 1. Queries that differ just in one term and concept, i.e., one query contains a keyword as a term, whereas the other query had it as a concept (one allowed edit in terms/concepts). 2. Queries with the same statements except different predicates (one allowed edit in predicates). Of course, in both cases, those queries may differ concerning our metrics. However, it helped us to estimate how close our selected queries were compared to the best ones. The counts are reported in Table 3. Especially for PM2020, which had the lowest number of correctly selected queries concerning precision, we found twelve queries that just differed by one term/concept and 16 queries that had a different predicate. This finding also applied to the other benchmarks: Allowing a small edit in terms/concepts or a different predicate led to considerably better results.

Discussion. First, our query model boosted the search for complex information needs, like stated in Genomics or PM2020. Next, our selection strategies did produce a high number of best queries concerning different evaluation metrics. And moreover, our methods were not adjusted for different benchmarks and were, thus, generalizable to a broad range of biomedical information needs.

8 CONCLUSIONS

In this work, we bridged the gap between the ease of keyword search and sophisticated narrative retrieval. Our evaluation has demonstrated that our proposed solutions were effective and generalized to a broad range of biomedical information needs. Moreover, user studies with domain experts verified the usefulness of our prototypical system. Especially from a digital library perspective, this work can be seen as a deep dive into how keyword-based search combined with sophisticated retrieval can be implemented, and, which possible challenges have to be faced on this way. Future work could design more advanced translation and selection strategies, improve the user interface based on our users' suggestions, and finally, investigate users' exploration strategies in a broader study to identify more requirements for the system.

ACKNOWLEDGMENTS

Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): PubPharm – the Specialized Information Service for Pharmacy (Gepris 267140244).

REFERENCES

[1] Katrin Affolter, Kurt Stockinger, and Abraham Bernstein. 2019. A comparative survey of recent natural language interfaces for databases. *VLDB J.* 28, 5 (2019), 793–819. https://doi.org/10.1007/s00778-019-00567-8

[2] Sophia Althammer, Sebastian Hofstätter, Suzan Verberne, and Allan Hanbury. 2022. TripJudge: A Relevance Judgement Test Collection for TripClick Health Retrieval. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. Association for Computing Machinery, 3801–3805. https://doi.org/10.1145/3511808.3557714

[3] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, and al. 2018. Construction of the Literature Graph in Semantic Scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. Association for Computational Linguistics, 84–91. https://doi.org/10.18653/v1/N18-3011

[4] Christine Betts, Joanna Power, and Waleed Ammar. 2019. GrapAL: Connecting the Dots in Scientific Literature. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*. Association for Computational Linguistics, 147–152. https://doi.org/10.18653/v1/p19-3025

[5] Gaurav Bhalotia, Arvind Hulgeri, Charuta Nakhe, Soumen Chakrabarti, and S. Sudarshan. 2002. Keyword Searching and Browsing in Databases using BANKS. In *Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, USA, February 26 - March 1, 2002*. IEEE Computer Society, 431–440. https://doi.org/10.1109/ICDE.2002.994756

[6] Nikos Bikakis, Giorgos Giannopoulos, John Liagouris, Dimitrios Skoutas, Theodore Dalamagas, and Timos K. Sellis. 2013. RDivF: Diversifying Keyword Search on RDF Graphs. In *Research and Advanced Technology for Digital Libraries - International Conference on Theory and Practice of Digital Libraries, TPDL 2013, Valletta, Malta, September 22-26, 2013. Proceedings (Lecture Notes in Computer Science, Vol. 8092)*. Springer, 413–416. https://doi.org/10.1007/978-3-642-40501-3_49

[7] Steven Bird. 2006. NLTK: The Natural Language Toolkit. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*. The Association for Computer Linguistics. https://doi.org/10.3115/1225403.1225421

[8] Laura Dietz, Alexander Kotov, and Edgar Meij. 2018. Utilizing Knowledge Graphs for Text-Centric Information Retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. ACM, 1387–1390. https://doi.org/10.1145/3209978.3210187

[9] Shady Elbassuoni and Roi Blanco. 2011. Keyword Search over RDF Graphs. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management* (Glasgow, Scotland, UK) *(CIKM '11)*. Association for Computing Machinery, 237–242. https://doi.org/10.1145/2063576.2063615

[10] Lujun Fang, Anish Das Sarma, Cong Yu, and Philip Bohannon. 2011. REX: Explaining Relationships between Entity Pairs. *Proc. VLDB Endow.* 5, 3 (2011), 241–252. https://doi.org/10.14778/2078331.2078339

[11] Michael Färber. 2019. The Microsoft Academic Knowledge Graph: A Linked Data Source with 8 Billion Triples of Scholarly Data. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 11779)*. Springer, 113–129. https://doi.org/10.1007/978-3-030-30796-7_8

[12] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. 2013. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*. International World Wide Web Conferences Steering Committee / ACM, 413–422. https://doi.org/10.1145/2488388.2488425

[13] Orest Gkini, Theofilos Belmpas, Georgia Koutrika, and Yannis Ioannidis. 2021. An In-Depth Benchmarking of Text-to-SQL Systems. In *Proceedings of the 2021 International Conference on Management of Data* (Virtual Event, China) *(SIGMOD '21)*. Association for Computing Machinery, 632–644. https://doi.org/10.1145/3448016.3452836

[14] Katerina Gkirtzou, Kostis Karozos, Vasilis Vassalos, and Theodore Dalamagas. 2015. Keywords-To-SPARQL Translation for RDF Data Search and Exploration. In *Research and Advanced Technology for Digital Libraries - 19th International Conference on Theory and Practice of Digital Libraries, TPDL 2015, Poznań, Poland, September 14-18, 2015. Proceedings (Lecture Notes in Computer Science, Vol. 9316)*. Springer, 111–123. https://doi.org/10.1007/978-3-319-24592-8_9

[15] Hao He, Haixun Wang, Jun Yang, and Philip S. Yu. 2007. BLINKS: ranked keyword searches on graphs. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Beijing, China, June 12-14, 2007*. ACM, 305–316. https://doi.org/10.1145/1247480.1247516

[16] William R. Hersh, Aaron M. Cohen, Lynn Ruslen, and Phoebe M. Roberts. 2007. TREC 2007 Genomics Track Overview. In *Proceedings of The Sixteenth Text REtrieval Conference, TREC 2007, Gaithersburg, Maryland, USA, November 5-9, 2007 (NIST Special Publication, Vol. 500-274)*. National Institute of Standards and Technology (NIST). http://trec.nist.gov/pubs/trec16/papers/GEO.OVERVIEW16.pdf

[17] Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. 2019. Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. In *Proceedings of the 10th International Conference on Knowledge Capture, K-CAP 2019, Marina Del Rey, CA, USA, November 19-21, 2019*. ACM, 243–246. https://doi.org/10.1145/3360901.3364435

[18] Amina Kadry and Laura Dietz. 2017. Open Relation Extraction for Support Passage Retrieval: Merit and Open Issues. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. ACM, 1149–1152. https://doi.org/10.1145/3077136.3080744

[19] Thomas Krause. 2019. *ANNIS: A graph-based query system for deeply annotated text corpora*. Ph. D. Dissertation. Humboldt University of Berlin, Germany. http://edoc.hu-berlin.de/18452/20436

[20] Christin Katharina Kreutz and Ralf Schenkel. 2022. Scientific paper recommendation systems: a literature review of recent publications. *Int. J. Digit. Libr.* 23, 4 (2022), 335–369. https://doi.org/10.1007/s00799-022-00339-w

[21] Hermann Kroll, Jan-Christoph Kalo, Denis Nagel, Stephan Mennicke, and Wolf-Tilo Balke. 2020. Context-Compatible Information Fusion for Scientific Knowledge Graphs. In *Digital Libraries for Open Knowledge - 24th International Conference on Theory and Practice of Digital Libraries, TPDL 2020, Lyon, France, August 25-27, 2020, Proceedings (Lecture Notes in Computer Science, Vol. 12246)*. Springer, 33–47. https://doi.org/10.1007/978-3-030-54956-5_3

[22] Hermann Kroll, Niklas Mainzer, and Wolf-Tilo Balke. 2022. On Dimensions of Plausibility for Narrative Information Access to Digital Libraries. In *Linking Theory and Practice of Digital Libraries - 26th International Conference on Theory and Practice of Digital Libraries, TPDL 2022, Padua, Italy, September 20-23, 2022, Proceedings (Lecture Notes in Computer Science, Vol. 13541)*. Springer, 433–441. https://doi.org/10.1007/978-3-031-16802-4_43

[23] Hermann Kroll, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. 2021. Narrative Query Graphs for Entity-Interaction-Aware Document Retrieval. In *Towards Open and Trustworthy Digital Societies - 23rd International Conference on Asia-Pacific Digital Libraries, ICADL 2021, Virtual Event, December 1-3, 2021, Proceedings (Lecture Notes in Computer Science, Vol. 13133)*. Springer, 80–95. https://doi.org/10.1007/978-3-030-91669-5_7

[24] Hermann Kroll, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. 2023. A discovery system for narrative query graphs: entity-interaction-aware document retrieval. *International Journal on Digital Libraries* (2023). https://doi.org/10.1007/s00799-023-00356-3

[25] Hermann Kroll, Florian Plötzky, Jan Pirklbauer, and Wolf-Tilo Balke. 2022. What a Publication Tells You—Benefits of Narrative Information Access in Digital Libraries. In *Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries* (Cologne, Germany) *(JCDL '22)*. Association for Computing Machinery, Article 9, 8 pages. https://doi.org/10.1145/3529372.3530928

[26] Markus Krötzsch and Sebastian Rudolph. 2016. Is your database system a semantic web reasoner? *KI-Künstliche Intelligenz* 30, 2 (2016), 169–176. https://doi.org/10.1007/s13218-015-0412-x

[27] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and Evaluation of a User Experience Questionnaire. In *HCI and Usability for Education and Work, 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2008, Graz, Austria, November 20-21, 2008. Proceedings (Lecture Notes in Computer Science, Vol. 5298)*. Springer, 63–76. https://doi.org/10.1007/978-3-540-89350-9_6

[28] Charles R. Lewis. 1982. Using the "thinking aloud" method in cognitive interface design. IBM TJ Watson Research Center Yorktown Heights, NY.

[29] Michael Ley. 2009. DBLP - Some Lessons Learned. *Proc. VLDB Endow.* 2, 2 (2009), 1493–1500. https://doi.org/10.14778/1687553.1687577

[30] Zhicheng Liang, Zixuan Peng, Xuefeng Yang, Fubang Zhao, Yunfeng Liu, and Deborah L. McGuinness. 2021. BERT-based Semantic Query Graph Extraction for Knowledge Graph Question Answering. In *Proceedings of the ISWC 2021 Posters, Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 20th International Semantic Web Conference (ISWC 2021), Virtual Conference, October 24-28, 2021 (CEUR Workshop Proceedings, Vol. 2980)*. CEUR-WS.org. http://ceur-ws.org/Vol-2980/paper379.pdf

[31] Zhiyong Lu. 2011. PubMed and beyond: a survey of web tools for searching biomedical literature. *Database* 2011 (01 2011). https://doi.org/10.1093/database/baq036 baq036.

[32] Frank Manola, Eric Miller, Brian McBride, et al. 2004. RDF primer. *W3C recommendation* 10, 1-107 (2004), 6.

[33] Jason Priem, Heather Piwowar, and Richard Orr. 2022. OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. https://doi.org/10.48550/ARXIV.2205.01833

[34] Navid Rekabsaz, Oleg Lesota, Markus Schedl, Jon Brassey, and Carsten Eickhoff. 2021. TripClick: The Log Files of a Large Health Web Search Engine. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 2507–2513. https://doi.org/10.1145/3404835.3463242

[35] T. J. Revanth, K. Venkat Sai, R. Ramya, Renusree Chava, V. Sushma, and B. S. Ramya. 2022. NL2SQL: Natural Language to SQL Query Translator. In *Emerging Research in Computing, Information, Communication and Applications*. Springer

Singapore, 267–278.

[36] Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, Steven Bedrick, and William R. Hersh. 2020. Overview of the TREC 2020 Precision Medicine Track. In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020 (NIST Special Publication, Vol. 1266)*. National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec29/papers/OVERVIEW.PM.pdf

[37] Tetsuya Sakai. 2007. Alternatives to Bpref. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Amsterdam, The Netherlands) *(SIGIR '07)*. Association for Computing Machinery, 71–78. https://doi.org/10.1145/1277741.1277756

[38] Shadi Saleh and Pavel Pecina. 2019. Term Selection for Query Expansion in Medical Cross-Lingual Information Retrieval. In *Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14-18, 2019, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 11437)*. Springer, 507–522. https://doi.org/10.1007/978-3-030-15712-8_33

[39] Lynda Tamine and Lorraine Goeuriot. 2022. Semantic Information Retrieval on Medical Texts: Research Challenges, Survey, and Open Issues. *ACM Comput. Surv.* 54, 7 (2022), 146:1–146:38. https://doi.org/10.1145/3462476

[40] Ellen M. Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R. Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2020. TREC-COVID: constructing a pandemic information retrieval test collection. *SIGIR Forum* 54, 1 (2020), 1:1–1:12. https://doi.org/10.1145/3451964.3451965

[41] Denny Vrandecic. 2012. Wikidata: a new platform for collaborative data collection. In *Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume)*. ACM, 1063–1064. https://doi.org/10.1145/2187980.2188242

[42] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, and al. 2020. CORD-19: The Covid-19 Open Research Dataset. *CoRR* abs/2004.10706 (2020). arXiv:2004.10706 https://arxiv.org/abs/2004.10706

[43] Gideon Zenz, Xuan Zhou, Enrico Minack, Wolf Siberski, and Wolfgang Nejdl. 2009. From Keywords to Semantic Queries-Incremental Query Construction on the Semantic Web. *Web Semant.* 7, 3 (sep 2009), 166–176. https://doi.org/10.1016/j.websem.2009.07.005