5369S Seminar: Knowledge Graphs

What is a Knownledge Graph?

A report by

Rascoussier, Florian Guillaume Pierre

PRÜFER
Prof. Dr. Alsayed Algergawy

MODERATOR
Andreas Einwiller

June 14, 2023

Abstract

Knowledge graphs (KGs) represent a powerful approach to organizing and structuring real-world information by modeling entities, their properties, and the relationships between them. As an enabling technology, KGs have gained significant traction over the last decade in various domains such as NLP, information retrieval, recommendation systems, and semantic search. This paper provides a comprehensive introduction to knowledge graphs, outlining their use cases, the current state of research, and industry adoption.

By facilitating advanced querying, reasoning, and knowledge discovery, KGs have become instrumental in numerous applications. The integration of KGs with machine learning techniques, such as graph neural networks and entity embeddings, has further bolstered their capabilities in prediction and pattern recognition. Research efforts are concentrated on KG construction, embedding methods, reasoning techniques, and evaluation metrics, while addressing issues like scalability, incompleteness, and dynamic evolution.

In the industry, major technology companies, including Google, Microsoft, and Facebook, have embraced KGs to enhance search engines, virtual assistants, and social media platforms. A rising number of startups and specialized firms are also employing KGs for diverse applications, ranging from drug discovery to fraud detection and smart manufacturing. Despite the considerable progress, challenges persist in areas such as data validation, real-time updates, privacy preservation, and usability. The current report discusses what are Knowledge Graphs, and introduces related concepts like KG construction, embedding methods, reasoning techniques, and evaluation metrics, while addressing issues like scalability, incompleteness, and dynamic evolution. It also outlines the current state of research, industry adoption, and future directions to advance the adoption and impact of knowledge graphs.

Acknowledgements

Contents

List of Figures

List of Tables

1 Introduction and History

1.1 Motivation

Knowledge Graphs are now widely used in many applications, such as search engines, virtual assistants, and social media platforms. Their usage span over many domains ranging from drug discovery to fraud detection passing by smart manufacturing. The current report is based mainly on prenote „Knowledge Graphs", postnote and similarly tries to introduce the subject of Knowledge Graphs to the seminar participants. It discusses what are Knowledge Graphs, the history behind, and introduces related concepts like KG construction, or reasoning techniques. The oral presentation has been given on the 7th of June 2023, under the supervision of Prof. Dr. Alsayed Algergawy, with Andreas Einwiller as monitor. This presentation was the introducing one and was followed by a discussion session. The current report is based on the presentation and the discussion session.

1.2 Historical Evolution of Knowledge Engineering (KE)

The history and evolution of the knowledge engineering discipline has seen significant transformation since its inception during the expert systems development phase in the 1980s. We can distinguish four periods ranging from 1955 to the present day, with each period introducing new requirements for knowledge production processes to overcome the limitations of systems developed in preceding periods.

- **Dawn of AI:** The initial focus was on reliable and effective processes.

- **Expert Systems Era:** Feigenbaum stressed the need for domain-specific focus for automated knowledge production, leading to the creation of expert systems. However, these systems proved to be brittle and hard to maintain, thus the need for scalable, globally distributed, and interoperable systems arose.

- **Semantic Web Era:** Tim Berners-Lee advocated for a "Web of Data" based on linked data principles, standard ontologies, and data sharing protocols. This period saw the development of a globally federated open linked data cloud and techniques for ontology engineering. However, wider adoption was slow and led to the call for more developer-friendly tools and methods to deal with data noise and incompleteness.

- **Language Model Era:** Language Learning Models (LLMs) are now ubiquitous due to recent advancements in neural network architectures and graphical processing hardware. Language models can either serve as knowledge bases that are queryable using natural language prompts or as a component in a knowledge production workflow.

2 What is a Knowledge Graph?

2.1 Knowledge and Graph Theory

Defining knowledge is not a straightforward task, so this report will focus on "explicit knowledge". Explicit knowledge is defined as "something that is known and can be written down" [4, p.4]. It is composed of statements, such as sentences, that draw relationships between concepts and data.

Graph theory is a field that lies at the intersection of computer science and mathematics and is concerned with the study of graphs. A graph is a type of data structure consisting of nodes (also known as vertices) and edges (or arcs) that connect pairs of nodes. Graph theory is used to model and analyze various types of relationships and structures in a wide range of fields, including computer networks, social networks, biological networks, and many others.

A Knowledge Graph can be defined as "a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent potentially different relations between these entities" [4, p.3]. Knowledge graphs serve as a common substrate of knowledge within an organization or community, enabling the representation, accumulation, curation, and dissemination of knowledge over time [4, p.31].

2.2 Knowledge Graphs

A Knowledge Graph is a data graph intended to accumulate and convey real-world knowledge. The nodes in the graph represent entities, and the edges represent relations between these entities. It serves as a common substrate for knowledge representation and dissemination. At the foundation of any knowledge graph is the principle of first modelling data as a graph [4, p.4].

A difficult definition

The term "knowledge graph" dates back to 1973, but it gained popularity through a 2012 blog post about Google's Knowledge Graph. Since then, several definitions of knowledge graphs have been proposed in research papers and by companies using or supporting KGs. For instance, Ehrlinger et al. define a knowledge graph as a structure that "acquires and integrates information into an ontology and applies a reasoner to derive new knowledge" [1]. However, this definition is very specific and excludes various industrial KGs which helped to popularize the concept. Therefore, we define KGs more inclusively as a graph of data consisting of semantically described entities and relations of different types that are integrated from different sources. Entities have a unique identifier. KG entities and relations are semantically described using an ontology or, more clearly, an ontological representation [3].

Understanding KG

At the foundation of any knowledge graph is the principle of first modelling data as a graph [4, p.4]. Graphs offer a flexible way to conceptualize, represent, and integrate diverse and incomplete data. Knowledge graphs use a graph-based data model to capture knowledge in application scenarios that involve integrating, managing and extracting value from diverse sources of data at large scale [4, p.2]. They have a number of benefits when compared with a relational model or NoSQL alternatives, such as the ability for data to evolve in a more flexible manner, and the capacity to organize data in a flexible way that is not hierarchical, can represent incomplete information, and does not require a precise schema [4, p.2].

Types of KG

Knowledge graphs can adopt any graph data model, and data can typically be converted from one model to another. Some of the different types of graphs include:

- **Directed Edge-labelled Graphs (DEL):** The classic graph, set of nodes and edges that connect the nodes with in certain way. RDF is a popular DEL data model.

- **Heterogeneous Graphs:** Each node and edge is assigned one type, allowing for partitioning nodes according to their type, which is useful for machine learning.

- **Property Graphs:** Allows a set of property–value pairs and a label to be associated with nodes and edges. This model is used in Neo4j and offers great flexibility but is harder to handle and query.

- **Graph Dataset:** A set of named graphs, with a default graph with no ID. Useful when working with different sources.

- **Hypergraphs:** Allow edges that connect sets rather than pairs of nodes.

2.3 Applications and Use Cases

Knowledge Graphs have found widespread application in both open and enterprise contexts. Open Knowledge Graphs are publicly accessible and often integrate data from various sources, while enterprise Knowledge Graphs are typically proprietary and used internally within organizations.

Open Knowledge Graphs

Several Open Knowledge Graphs have been developed, including:

- **BabelNet:** Integrates several resources including Wikipedia and WordNet to provide multilingual lexical knowledge.

- **DBpedia:** Extracts structured content from Wikipedia to make it accessible on the Web.

- **Freebase:** A crowdsourced database of well-known people, places, and things.

- **Wikidata:** Serves as the central storage for the structured data of Wikimedia projects.

- **YAGO:** Automatically extracts and integrates knowledge from Wikipedia and other sources.

Enterprise Knowledge Graphs

Enterprise Knowledge Graphs are used in various industries including web search, commerce, social networks, and finance. Some examples include:

- **Google Knowledge Graph:** Enhances Google Search's results with semantic-search information gathered from various sources.

- **Amazon Product Knowledge Graph (PKG):** A large-scale, semi-structured knowledge graph that organizes information about products sold on Amazon and relationships between them.

These Knowledge Graphs are used in a wide range of applications including search, recommendations, information extraction, personal agents, advertising, business analytics, risk assessment, automation, and more.

3 Advanced Topics in Knowledge Graphs

3.1 Construction, Creation, Extraction

Knowledge Graphs (KGs) are physical data integration systems that amalgamate information from a variety of sources into a logically centralized, graph-like representation. The creation of KGs often involves integrating data from diverse sources, including direct human input, extraction from existing text, markup, legacy file formats, relational databases, and other knowledge graphs.

The construction of KGs involves several tasks. Data Acquisition and Preprocessing involves the selection of relevant sources, acquisition and transformation of relevant source data, and initial data cleaning. Metadata Management involves the acquisition and management of different kinds of metadata, such as provenance of entities, structural metadata, temporal information, quality reports, and process logs. Ontology Management involves the creation and incremental evolution of a KG ontology. Knowledge Extraction (KE) involves the derivation of structured information and knowledge from unstructured or semi-structured data. Entity Resolution (ER) and Fusion involves the identification of matching entities and their fusion within the KG. Quality Assurance (QA) involves identifying and repairing data quality problems in the KG. Knowledge Completion involves extending a given KG, for example, by learning missing type information, predicting new relations, and enhancing domain-specific data.

3.2 Search and Querying

Querying KGs often involves the use of graph query languages such as SPARQL for RDF graphs and Cypher, Gremlin, and G-CORE [4] for property graphs. These languages allow for the creation of graph patterns, which are graphs similar to the data being queried that can contain variables so that they can be evaluated to retrieve information from constants in the KG. Complex Graph Patterns combine several graph patterns with operators, while Navigational Graph Patterns use regular expressions for matching paths with support for more complex querying.

3.3 Schema, Ontology and Validation

Ontologies provide a formal representation of what terms mean within the scope in which they are used. They allow for the creation of assumptions, semantic conditions, individuals, properties, and classes. Validation of KGs involves ensuring certain properties in the graph depending on its uses. This can be achieved through the use of shapes graphs and conformance. Shapes Graphs are selected subsets of nodes, with specified constraints, usually expressed using UML diagrams. Conformance refers to a node conforming to a shape

if it satisfies all of the constraints of the shape. A valid graph is such that every node conforms to a given shape.

3.4 Deduction, Inference and Entailment

Deduction and inference involve deriving new facts or knowledge from the existing data in the KG. This is typically achieved through logical reasoning based on the relationships and rules defined within the graph, or even context and additional external information. Entailment refers to the process where the truth of one statement necessarily implies the truth of another. By extension, one graph entails another if and only if any model of the former graph is also a model of the latter graph.

3.5 Inductive Reasoning and Learning

Inductive reasoning involves making generalizations based on observed patterns. This could involve using machine learning techniques to predict missing links or infer new knowledge. Techniques for inductive reasoning in KGs include graph analytics, knowledge graph embeddings, graph neural networks, symbolic learning, rule mining, and axiom mining. Graph analytics involve using well-known algorithms to detect communities or clusters, find central nodes and edges, and so on, in a graph. Knowledge graph embeddings learn a low-dimensional numerical model of elements of a knowledge graph. Graph neural networks are a kind of neural network where nodes are connected to their neighbours in the data graph. Symbolic learning involves learning symbolic models—i.e., logical formulae in the form of rules or axioms—from a graph in a self-supervised manner. Rule mining involves learning meaningful rules as logical patterns. Axiom mining involves determining what are impossible cases from the graph, known as disjointness constraints.

4 Critical Evaluation of the papers

4.1 Main paper

Evaluation of paper prenote „Knowledge Graphs", postnote.

Contributions and Improvements

The paper "Knowledge Graphs" provides a comprehensive exploration of Knowledge Graphs (KGs), offering an overview of their structures, applications, and related concepts. It is particularly valuable for readers with varying levels of expertise in KGs, as it does not assume specific knowledge in this area. The paper conducts a meta-analysis of 13 external papers and books, providing a broad perspective on the subject matter.

An extended online version of the paper is available, which includes concrete examples on GitHub. This is a significant contribution, as it allows readers to engage with practical applications of the concepts discussed in the paper. The paper also provides an in-depth discussion of complex topics, demonstrating a deep understanding of these concepts and how they apply to KGs.

The paper also contributes to the field by providing an extensive bibliography for further reading, which

is particularly useful for readers who wish to delve deeper into specific topics. However, the paper could be improved by providing more practical examples or case studies showing how these concepts are applied in real-world settings.

Evaluation

The paper is a valuable resource for anyone interested in understanding KGs. It provides comprehensive coverage and clear explanations of complex concepts, making it a significant contribution to the field. However, the complexity of the topics discussed may challenge readers who are new to the field.

The paper is well-structured and presents its content clearly, making it a readable resource. However, there are small mistakes in the paper, and it could benefit from more practical examples or case studies. Despite these minor issues, the paper provides a deep and insightful exploration of KGs and related concepts.

Overall, the paper is a thorough and detailed examination of KGs. It is useful for researchers and practitioners alike, and it makes a significant contribution to the field. However, it could be improved by providing more practical examples, discussing challenges associated with KGs, and providing clearer explanations of some concepts.

4.2 Paper used for information on KG construction

Evaluation of paper prenote „Construction of Knowledge Graphs: State and Challenges", postnote.

Contributions and Improvements

The paper provides a comprehensive review of the current state of Knowledge Graph (KG) construction. It discusses the main requirements, tasks, and challenges associated with KG construction and compares existing KG construction pipelines and toolsets. The paper introduces and categorizes general requirements for incremental KG construction and provides an overview of the main tasks in incremental KG construction pipelines and proposed solution approaches.

The paper contributes to the field by investigating and comparing existing construction efforts for selected KGs and recent tools for KG construction. It emphasizes the need for more open-source toolsets for KG development and highlights the importance of good data and metadata management. The paper also stresses the importance of high data quality and the evaluation of complete KG construction pipelines.

However, the paper is still a preprint and is available on Arxiv. As such, it has not yet undergone peer review, which could impact the reliability of its findings.

Evaluation

The paper provides a comprehensive overview and comparison of the state of KG construction. It identifies key challenges and requirements for KG construction, which is a significant strength of the paper. However, it lacks practical examples or case studies to illustrate the discussed concepts, which is a weakness.

The paper is well-structured with clear sections and subsections, and it uses clear and concise language, making it accessible to readers familiar with the topic. However, it could be difficult for readers without a background in KGs or related fields, as some technical terms and concepts are not explained in detail.

Overall, the paper provides a valuable contribution to the field of KG construction. It successfully identifies the current state of KG construction and highlights areas for future research and improvement. It is a valuable resource for researchers and practitioners in the field of KG construction. However, it could be improved by including practical examples or case studies and explaining technical terms and concepts in more detail.

4.3 Paper used for information about KG related history

Evaluation of paper prenote „Knowledge Graphs and their Role in the Knowledge Engineering of the 21st Century", postnote.

Contributions and Improvements

The paper provides an overview of the evolution of knowledge engineering since the 1980s, identifying four distinct periods and summarizing their consequences. It includes figures to illustrate the evolving requirements for knowledge production processes. However, it should be noted that only section 3.2 of the paper was used for this review, limiting the scope of the critique.

Evaluation

The paper is well-structured and presents a clear progression of ideas. It provides valuable insights into the history and evolution of knowledge engineering, making it relevant for anyone in the field. The paper contributes to the understanding of the changing requirements and challenges in knowledge engineering over time and raises important questions about the future of knowledge engineering, particularly in relation to knowledge graphs.

5 Questions raised during the presentation

5.1 Andreas Einwiller: How is inductive and deductive reasoning related to ML and Ontologies/Rules?

Deductive and inductive reasoning are fundamental methods used in knowledge graphs (KGs) and are closely related to machine learning and ontologies/rules.

Deductive reasoning is a top-down logical process that moves from general to specific. In the context of KGs, it involves inferring new facts based on existing facts and rules in the graph. If all premises are true, then the conclusion must also be true. Let's use the classical example about Socrates: if we know that "all men are mortal" (general premise, fact) and that "Socrates is a man" (specific premise, rule), we can thus *deduce* that "Socrates is mortal" (specific conclusion, new deduced fact). This process is also known as "knowledge graph completion" or "link prediction".

Ontologies, which provide a formal representation of terms and their relationships, play a crucial role in deductive reasoning. They form the basis for the rules used in this process. Inference rules and Description Logics (DLs) are used to make claims (known as axioms) about these elements, facilitating the process of deduction. Machine learning is also widely used to automate the process of deductive reasoning in KGs. For example, machine learning algorithms can be trained to predict missing links or infer new facts based on patterns in the existing data.

Inductive reasoning, on the other hand, is a bottom-up logical process that moves from specific to general. In the context of KGs, it involves learning new rules based on patterns observed in the data. It follows a probabilistic process: specific observations are generalized into rules, which are likely but not guaranteed to be true. A simple example of inductive reasoning could be that "all observed swans are white, therefore all swans are white". As history has shown, this is not necessarily true, but it is likely to be true.

For example, if we observe many instances of a relationship like "Person works at Company" and "Company is located in City", we might induce a rule like "Person lives in City". This process is often used in "rule mining" or "entity prediction" in KGs.

Machine learning plays a significant role in inductive reasoning. Machine learning algorithms can be trained to identify patterns in the data and generate new rules based on these patterns. This allows KGs to continuously evolve and improve, enhancing their utility and effectiveness.

In conclusion, both deductive and inductive reasoning are crucial for the construction and maintenance of KGs. Ontologies and rules provide the foundation for deductive reasoning, while machine learning techniques facilitate both deductive and inductive reasoning. The interplay between these elements enables KGs to continuously evolve and improve, enhancing their utility and effectiveness.

5.2 Amandeep Gill: How do knowledge graphs handle scalability issues, and what are the associated trade-offs?

KG scalability is still a major challenge, especially in dynamic KGs where data is frequently updated [3]. Scalability issues can be tackled through a variety of strategies, each with its own trade-offs.

Large-scale KGs often leverage distributed storage systems and parallel processing frameworks to handle the vast amount of data. This allows for the storage and processing of data across multiple machines, thereby increasing capacity and speed. However, this approach can introduce complexity in terms of data management and can also increase the risk of data inconsistency. So as to limit the size of KGs, partitioning the graph into smaller more manageable subgraphs is another common strategy. This can be done based on various criteria such as the type of entities or relationships. While partitioning can significantly improve query performance, it can also lead to challenges in managing inter-partition relationships and can complicate queries that span multiple partitions.

In order to speed up data retrieval, on can use indexing. By creating indexes on frequently accessed or queried data, the system can quickly locate the required data without scanning the entire graph. However, indexes can consume significant storage space and maintaining them can add overhead, especially in dynamic KGs where data is frequently updated. Caching is another strategy used to improve query performance. Frequently accessed data is stored in a cache for quick retrieval, similar to how computer memory is layed-out. While caching can significantly speed up data access, it requires careful management to ensure data

consistency. Additionally, it is most effective with a high degree of data locality, when the same data is accessed repeatedly.

For some types of queries, especially those involving complex graph algorithms, exact solutions can be computationally expensive or even impossible (taking infinite process time). In such cases, approximation techniques can be used to provide near-optimal solutions more efficiently, with some accuracy trade-off. In some cases, inferred knowledge (knowledge that can be deduced from existing facts and rules) can be precomputed and stored, or materialized, in the graph. This can speed up query processing but at the cost of increased storage requirements and potential issues with keeping the materialized knowledge up-to-date.

Thus, handling scalability in KGs involves finding balance between performance, storage requirements, complexity, and accuracy. The specific trade-offs depend on the characteristics of the KG and the specific requirements of the use case. However, it is important to note that scalability is still a major challenge in KGs.

5.3 Chirag Natesh Vijay: Could you explain a bit more as to how exactly knowledge graph is suitable for machine learning?

KGs are particularly suitable for machine learning for a range of reasons. KGs provide a rich, structured representation of data, capturing not only entities but also the relationships between them [4]. This allows ML models to leverage this relational information, which can provide important context and improve performance. The inherent nature of KGs makes it easier to generate features for ML models. For example, the properties of an entity, the types of relationships it has with other entities, or the structure of its neighborhood in the graph can all be used as features.

KGs are also naturally suited to relational learning or graph-based ML methods. Techniques such as Graph Neural Networks (GNNs) or Graph Convolutional Networks (GCNs) can leverage the graph structure of KGs to propagate information across the graph and learn more accurate models [4]. KGs can facilitate transfer learning, where knowledge learned in one context is applied to another. For example, a model trained on one part of the KG can potentially be applied to other parts of the KG, leveraging the shared structure and semantics (onthologies, rules, etc).

KGs often contain a mix of labeled and unlabeled data, making them suitable for semi-supervised learning approaches. These methods can leverage the large amounts of unlabeled data to improve learning from the smaller amounts of labeled data. The structures and relations and rules can also help with explainability in ML. The reasoning paths in a KG can provide intuitive, human-understandable explanations for the predictions of an ML model.

In conclusion, KGs provide a rich, structured, and semantically meaningful source of data that can enhance various aspects of machine learning, from feature generation and model training [4] to transfer learning and explainability.

6 Conclusion

This report has provided a comprehensive review of three key papers in the field of Knowledge Graphs (KGs). Each paper was critically evaluated based on its contributions, improvements, and overall quality. The main

paper, „Knowledge Graphs" [4], provides an in-depth overview of KGs, with a focus on theoretical concepts. It is a valuable resource for anyone interested in understanding KGs, despite some areas that could benefit from further clarification or practical examples.

The second paper, „Construction of Knowledge Graphs: State and Challenges" [3], offers a state-of-the-art review of KG construction, highlighting the need for more open-source toolsets and good data management. However, it is still a preprint and lacks practical examples. The third paper, „Knowledge Graphs and their Role in the Knowledge Engineering of the 21st Century" [2], provides valuable insights into the history and evolution of knowledge engineering. However, only section 3.2 was used for this review, limiting the scope of the critique.

In conclusion, the field of KG have been rapidly evolving especially over the course of the last decade, with significant contributions being made in both theoretical concepts and practical applications. The reviewed papers provide valuable insights into this field and highlight areas for future research and improvement. This report has aimed to critically evaluate these papers and provide a comprehensive overview of the current state of KGs.

Acronyms

KG Knownledge Graph. i, 1

NLP Natural Language Processing. i

References

[1] Lisa Ehrlinger and Wolfram Wöß. „Towards a Definition of Knowledge Graphs". In: (2016), pp. 1–4.

[2] Paul Groth et al. „Knowledge Graphs and their Role in the Knowledge Engineering of the 21st Century". In: *Dagstuhl Reports* 12.9 (2022). Report from Dagstuhl Seminar 22372. Specific usage: pp. 60-72, Subsection "3.2 A Brief History of Knowledge Engineering: A Practitioner's Perspective", pp. 60–120. DOI: 10.4230/DagRep.12.9.60.

[3] Marvin Hofer et al. „Construction of Knowledge Graphs: State and Challenges". In: *arXiv preprint arXiv:2302.11509* (2023). URL: https://doi.org/10.48550/arXiv.2302.11509.

[4] Aidan Hogan et al. „Knowledge Graphs". In: *ACM Comput. Surv.* 54.4 (July 2021). ISSN: 0360-0300. DOI: 10.1145/3447772. URL: https://doi.org/10.1145/3447772.

Additional bibliography

[5] Frederick Edward Hulme. *Proverb Lore: Many Sayings, Wise Or Otherwise, on Many Subjects, Gleaned from Many Sources.* E. Stock, 1902, p. 188.