Masterarbeit

Predicting SSH keys in Open SSH Memory dumps

A report by

Rascoussier, Florian Guillaume Pierre

Prüfer

Prof. Dr. Michael Granitzer

Christofer Fellicious

Prof. Dr. Pierre-Edouard Portier

August 8, 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 that has seen major developments in the last decade, KGs have gained significant traction in various domains such as NLP, information retrieval, recommendation systems, and semantic search.

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. In the industry, major technology companies, including Google, or Amazon have embraced KGs to enhance search engines, virtual assistants, and social media platforms. Major open-source projects are centered around KGs 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 construction, embedding methods as well as reasoning techniques. As a seminar report, it is based on several papers and also include the questions raised during the presentation.

Acknowledgements

I would like to express my sincere gratitude to all the participants of the seminar on Knowledge Graph for their valuable insights, discussions, and contributions.

A special acknowledgment goes to my monitor, Prof. Dr. Alsayed Algergawy, for his support, encouragement, and inspiration throughout the course of this work, especially for the selection of relevant papers to review. His knowledge and experience in the field have been invaluable in this seminar. I would also like to thank Andreas Einwiller for his guidance and feedback before and during the presentation and discussion session.

Contents

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 „Knowledge Graphs“ 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 first one of the seminar 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. Four different periods can be distinguished, 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 [0].

- **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 or Large Language 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.

All thoses differents phases have seen the development of new concepts, tools and methods to deal the ever growing complexity of the knowledge production processes and analysis. In the wake of Google's announcements in 2012 [0], the last decade has seen the development of Knowledge Graphs as a powerful approach to organizing and structuring real-world information by modeling entities, their properties, and the relationships between them.

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" [0, 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" [0, 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" [0, p.31].

2.2 Knowledge Graphs

"At the foundation of any knowledge graph is the principle of first modelling data as a graph" [0, p.4]. A Knowledge Graph is thus 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 that is both flexible and extendable.

A difficult definition

The term "knowledge graph" first appeared in 1973, but really gained popularity through a 2012 blog post about Google's Knowledge Graph [0]. According to litterature, below are listed some of the most common definitions of Knowledge Graphs:

- "A knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge." [0]
- "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." [0]
- "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." [0]

The very nature of KG makes any definition attempt difficult. Indeed, KG is a broad concept that can be applied to many different domains, use cases and can have diverse implementations. The definition of KGs is thus very context-dependent. However, the common denominator of all KGs is that they are graphs of data that represent some knowledge or information in a graph-structured representation.

Understanding KG

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" [0, 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 way that is not hierarchical. They can represent incomplete information, and does not require a precise schema [0, 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 source, research 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

Building Knowledge Graphs often require physical data integration systems that amalgamate information from a variety of sources into a logically centralized, graph-like representation. This creation process often involves integrating data from diverse sources, including direct human input, extraction from existing text, markup, legacy file formats, csv or json files, relational databases, or even other knowledge graphs.

The construction of KGs involves several tasks. The initial step often consists in data acquisition and preprocessing. This phase involves the selection of relevant sources, acquisition and transformation of relevant source data, as well as initial data cleaning. Since the data used can be of diverse types, metadata management is important so as to deal with different kinds of metadata like the provenance of entities, structural metadata, temporal information, quality reports, and process logs. The complexity of integrating diverse data from different sources also lies in the ontology management. A simple approach to this is to allow incremental evolution of the ontology, but this can lead to inconsistencies and incoherences.

Once the KG is in place, more valuable information can be extracted from it: this is the knowledge extraction phase. Knowledge Extraction (KE) refers to the derivation of structured information and knowledge from unstructured or semi-structured data. This can be done through a range of techniques such as entity resolution and fusion, quality assurance, and knowledge completion. 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 [0] 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. Graph patterns can follow either homomorphism-based semantics, which allow multiple variables to be mapped to the same term, or isomorphism-based semantics, which require variables on nodes and/or edges to be mapped to unique terms.

In more complex scenarios, Complex Graph Patterns and Navigational Graph Patterns can be used. Complex Graph Patterns combine several graph patterns using operators, such as union, filter, and where. While powerful, these can generate duplicates in the answer. On the other hand, Navigational Graph Patterns use regular expressions for matching paths, supporting more complex querying similar to using regular expressions (disjunction, concatenation, set of possible values, etc.). However, these can generate an infinite number of paths, so it can be more efficient to only return nodes, which are always finite in number.

It is worth noting that graph query languages may support a range of other features such as aggregation, complex filters, datatype operators, sub-queries and so on [0]. These additional features further enhance the flexibility and power of KG querying, enabling users to perform sophisticated data retrieval and analysis operations.

3.3 Validation, Schema and Ontology

Knowledge graphs, due to their inherent ability to represent incomplete and possibly incoherent data, often need some kind of validation processes in order to ensure certain properties within the graph depending on its intended use.

Key concepts related to graph validation include Shapes Graphs, Conformance, and Context [0]. Shapes Graphs are a selected subset of nodes, with specified constraints, typically expressed using UML diagrams. These can be either open or closed shapes, allowing or disallowing the node to have additional properties not specified by the shape. Conformance is another crucial aspect of validation. A node is said to conform 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. Various shape languages extensions to RDF are available for this purpose, such as ShEx and SHACL. Context is also an important consideration during validation. Every piece of information exists with respect to a particular context. This, along with origin/provenance and time frame, defines the "scope of truth". A context can be implicit or explicit and can be represented in various ways such as Direct Representation, Reification, Higher-arity Representation, annotations, and other more complex solutions.

Ontologies provide a formal convention-like 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 as discussed earlier. Many ontologies already exist depending on the context and use cases, such as the Web Ontology Language (OWL) by W3C, which is RDF compatible, and the Open Biomedical Ontologies Format (OBOF).

Ontologies in knowledge graphs involve several key concepts such as Interpretations, Assumptions, Semantic Conditions, Individuals, Properties, and Classes. Interpretations refer to the mapping of nodes and edges to entities and relations in the real world. Assumptions dictate how knowledge graphs can be interpreted. Semantic Conditions are case-specific assumptions that facilitate reasoning and entailment in the graph. Individuals refer to real-life entities. Properties are terms that can be used as edge-labels. And Classes are groups of nodes under a similar type. Ontologies can be complex, with many more features like datatype facets, which involve defining new datatypes by applying restrictions to existing datatypes. Despite their complexity, ontologies play a crucial role in the construction and interpretation of knowledge graphs.

3.4 Deduction, Inference and Entailment

In the context of Knowledge Graphs (KGs), several key processes facilitate the extraction of new knowledge from existing data. These include deduction, inference, and entailment [0]. The distinction between thoses three concepts can be hard to grasp, but in a few words, deduction and inference involve deriving new facts or knowledge from the existing data. 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.

Deduction refers to the process of deriving new data from what is already given, along with some implicit or explicit rules. This allows us to know more than what is explicitly given to us by the data. Deductions can serve a range of applications, such as improving query answering, classification, finding inconsistencies, and so on.

Inference, on the other hand, refers to the process of deriving or deducing new facts or knowledge from the existing data in the graph. This is typically achieved through logical reasoning based on the relationships and rules defined within the graph, or even context and additional external information. Inference rules can be added as if-then statements with body and head being graph patterns, and predefined sets of rules even exist for popular ontologies. Inference in KGs is thus a powerful tool for enriching the graph with additional information, improving the quality of search and query results, and enabling more sophisticated data analysis and decision-making processes.

Entailment is a deductive process where a relationship between statements or sets of statements exists such that the truth of one statement or set implies the truth of another, with some degree of confidence. These processes are often guided by Model-theoretic Semantics, which involve adding property axioms that define truth conditions, meaning that only certain interpretations become possible. The interpretations that satisfy a graph are called 'models' of the graph. One can say that a graph entails another one if any model of the former graph is also a model of the latter graph. In other words, two graphs are entailed if they mean the same.

3.5 Inductive Reasoning and Learning

Inductive reasoning involves making generalizations based on observed patterns. This could involve using machine learning techniques to infer new knowledge and generalize patterns from input observations. It can then be used to generate novel but potentially imprecise predictions. However, inductive reasoning also presents challenges such as handling noise, incompleteness, and uncertainty in the data [0]. Inductive reasoning techniques can be broadly categorized into graph analytics, knowledge graph embeddings, graph neural networks, and symbolic learning.

Graph analytics involve the use of well-known algorithms to detect communities or clusters, find central nodes and edges, and so on, in a graph. This for instance includes centrality analysis, community detection, or connectivity analysis. The recent breakthrough in machine learning has led to the development of knowledge graph embeddings as new techniques for inductive reasoning. Knowledge graph embeddings aim to learn a low-dimensional numerical model of elements of a KG. This involves transforming the graph into a vector, a process known as embedding. Various techniques exist for this purpose, including adjacency sparse matrix, plausibility embedding or tensor decomposition models.

Graph neural networks (GNNs) are another type of neural network where nodes are connected to their neighbors in the data graph. They have been used extensively to perform classification tasks in an ever-growing range of situations. Two main types of GNNs are Recursive graph neural networks (RecGNNs) and Convolutional Graph Neural Networks (ConvGNNs). Other ML techniques can also be used for inductive reasoning, such as symbolic learning, which aims to learn logical formulae in the form of rules or axioms (symbolic models) from a graph in a self-supervised manner. This allows for the learning of logic rules and reasoning, such that the decision-making can rely on a well-defined explanation provided by the model rather than on numerical values. Techniques in this category include rule mining and axiom mining.

4 Critical Evaluation of the papers

4.1 Main paper

The paper „Knowledge Graphs“ has been the main ressource for this report. It 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.

Some negatives points about the paper is its complexity, especially considering that its targeted audience is not necessarily familiar with KGs. I could also find some minor mistakes in the paper. For instance, page 29, we can read: "In more detail, we call the edges entailed by a rule and the set of positive edges (not including the entailed edge itself) the positive entailments of that rule. The number of entailments that are positive is called the support for the rule, while the ratio of a rule's entailments that are positive is called the confidence for the rule [127]." [0, p29]. There is a missing expression here, such that the corrected text would be "...we call the edges entailed by a rule *the entailments of the rule* and the set of positive edges (not including the entailed edge itself) the positive entailments of that rule.". Another aspect is the lack of discussion on challenges associated with knowledge graphs, such as issues related to scalability, data quality, diversity or dynamicity (temporal and streaming data).

However the paper is well structured and provides a clear overview of the subject, as well as an extended online version, 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. It provides an in-depth discussion of complex topics, demonstrating a deep understanding of these concepts and how they apply to KGs.

The paper thus contributes to the field as a valuable resource for anyone interested in understanding KGs. It provides comprehensive coverage and clear explanations even on complex concepts, making it a significant contribution to the field and a good entry-point. By providing an extensive bibliography and an extended version for further reading, this paper is particularly useful for readers who wish to delve deeper into specific topics while still grasping the bigger picture.

4.2 Paper used for information on KG construction

The paper „Construction of Knowledge Graphs: State and Challenges" provides a comprehensive review of the current state of Knowledge Graph 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 construction pipelines.

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 as well as metadata management. Special attention has been given to figures and illustrations which really help to understand the concepts discussed in the paper. However, it lacks practical examples or case studies to illustrate the discussed concepts, which is a weakness. 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 possibly not yet undergone peer review, which could impact the reliability of its findings.

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 alike once it has been peer-reviewed and published.

4.3 Paper used for information about KG related history

The paper „Knowledge Graphs and their Role in the Knowledge Engineering of the 21st Century“ provides an overview of the evolution of knowledge engineering since the 1980s, identifying four distinct periods and summarizing their consequences. It also includes a comprehensive figure to illustrate the different phases of KE. However, it should be noted that only section 3.2 of the paper was used for this review, limiting the scope of the critique.

This section 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 wanting to have an introduction about the history of KE.

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 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" [0]. 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 "Ben works at company X" and "company X is located in city Y", we might induce a rule like "Ben lives in city Y". This process is often used in rule mining or entity prediction. 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.

In conclusion, both deductive and inductive reasoning are crucial for the construction, maintenance and use of KGs. Ontologies and rules provide the foundation for deductive reasoning, while machine learning techniques facilitate both deductive and inductive reasoning by leveraging on the amount of data and its structure in a given KG.

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 [0]. 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 a 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 as well as increasing 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 [0]. 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 [0]. 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 [0]. 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, 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 [0] to transfer learning and explainability.

6 Conclusion

This report has provided a comprehensive review of three key papers in the field of Knowledge Graphs. Each paper was critically evaluated based on its contributions, improvements, and overall quality. The main paper, „Knowledge Graphs“ [0], 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. It's extended version, GitHub repository and references provide additional resources for readers who wish to delve deeper into specific topics.

The second paper, „Construction of Knowledge Graphs: State and Challenges“ [0], 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 section 3.2 of the third paper, „Knowledge Graphs and their Role in the Knowledge Engineering of the 21st Century“ [0], provides valuable insights into the history and evolution of knowledge engineering.

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.

7 Introduction

Motivate your research and outline the research gap in this chapter. Why is your thesis relevant and what do you address, what has not been addressed before.

General Requirements to the thesis:

- 60 pages of content in this format. Content does not include table of content, lists, appendices etc.
- Proper scientific referencing
- Introduction and Background should be less than 50% of the thesis
- Images should be readable and in the proper size.

8 Research Questions

Write down and explain your research questions (2-5)

9 Structure of the Thesis

Explain the structure of the thesis.

10 Example citation & symbol reference

For symbols look at [**latex_symbols_2017**].

11 Example reference

Example reference: Look at chapter 7, for sections, look at section 10.

12 Example image

Example figure reference: Look at Figure 1 to see an image. It can be `jpg`, `png`, or best: `pdf` (if vector graphic).

Figure 1: Meaningful caption for this image

First column	Number column
Accuracy	0.532
F1 score	0.87

Table 1: Meaningful caption for this table

13 Example table

Table 1 shows a simple table[1]

[1] Check `https://en.wikibooks.org/wiki/LaTeX/Tables` on syntax

14 Background

Introduce the related state-of-the-art and background information in order to understand the method developed in the thesis.

15 Methods

Describe the method/software/tool/algorithm you have developed here

16 Results

Describe the experimental setup, the used datasets/parameters and the experimental results achieved

17 Discussion

Discuss the results. What is the outcome of your experimetns?

18 Conclusion

Summarize the thesis and provide a outlook on future work.

A Code

B Math

C Dataset

Acronyms

DEL Directed Edge-labelled Graphs. 3

ER Entity Resolution. 5

GCN Graph Convolutional Networks. 11

GNN Graph Neural Network. 11

KE Knowledge Engineering. 5, 9

KG Knownledge Graph. i, 1

ML Machine Learning. 7

NLP Natural Language Processing. i

QA Quality Assurance. 5

RDF Resource Description Framework. 3, 5, 6

SPARQL SPARQL Protocol and RDF Query Language. 5

References

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

[0] Google. „Introducing the Knowledge Graph: Things, not strings“. In: *Google Blog* (May 2012). Accessed: 2023-06-16. URL: `https://blog.google/products/search/introducing-knowledge-graph-things-not/`.

[0] 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`.

[0] 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`.

[0] 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`.

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

Additional bibliography

[0] Gianluca Fiorelli. *Best of 2013: No 13 – Search in the Knowledge Graph era*. Accessed: 2023-06-12. 2013. URL: `https://www.stateofdigital.com/search-in-the-knowledge-graph-era/`.

[0] Jackson Gilkey. *Graph Theory and Data Science*. Accessed: 2023-05-25. 2019. URL: `https://towardsdatascience.com/graph-theory-and-data-science-ec95fe2f31d8`.

[0] M.S. Jawad et al. „Adoption of knowledge-graph best development practices for scalable and optimized manufacturing processes“. In: *MethodsX* 10 (2023), p. 102124. ISSN: 2215-0161. DOI: `https://doi.org/10.1016/j.mex.2023.102124`. URL: `https://www.sciencedirect.com/science/article/pii/S2215016123001255`.

[0] Michelle Venables. *An Introduction to Graph Theory*. Accessed: 2023-06-12. 2019. URL: `https://towardsdatascience.com/an-introduction-to-graph-theory-24b41746fabe`.

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich diese Masterarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe und alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, als solche gekennzeichnet sind, sowie, dass ich die Masterarbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt habe.

Passau, August 8, 2023

Rascoussier, Florian Guillaume Pierre