A Study on Application of Knowledge Graph

A thesis
Submitted in partial fulfillment of the requirements for the Degree of
Bachelor of Science in Computer Science and Engineering

Submitted by

Parvez Ahammemd	**20200104129**
Tabassum Tara Lamia	**20200104128**
Fatin Ishraq	**20200104106**
Shawon Kumar Modak	**20200104139**

Supervised by

Prof. Dr. S.M.A Al-Mamun

Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Dhaka, Bangladesh

December 2024

CANDIDATES' DECLARATION

We, hereby, declare that the thesispresented in this report is the outcome of the investigation performed by us under the supervision of Prof. Dr. S.M.A Al-Mamun, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE400: Project and Thesis I and CSE450: Project and Thesis II, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Parvez Ahammemd
20200104129

Tabassum Tara Lamia
20200104128

Fatin Ishraq
20200104106

Shawon Kumar Modak
20200104139

CERTIFICATION

This thesis titled, **"A Study on Application of Knowledge Graph"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in December 2024.

Group Members:

Parvez Ahammemd	**20200104129**
Tabassum Tara Lamia	**20200104128**
Fatin Ishraq	**20200104106**
Shawon Kumar Modak	**20200104139**

Prof. Dr. S.M.A Al-Mamun
Professor & Supervisor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Prof. Dr. Md. Shamim Akhter
Professor & Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

ACKNOWLEDGEMENT

First and foremost, we extend our sincerest gratitude to the Divine Providence for bestowing upon us His boundless blessings, which have facilitated the successful culmination of this scholarly endeavor. Our profound appreciation is extended to Dr. S.M.A Al-Mamun, Profes sor in the Department of Computer Science Engineering at Ahsanullah University of Science and Technology, whose unwavering guidance and steadfast support havebeen indispensable in the realization of this dissertation.

We are grateful to our parents for their unconditional love, sacrifices, and prayers, which have been the foundation of our success. Their unfailing support and encouragement have empowered us to overcome hurdles and strive for excellence. This adventure would not be possible without their unwavering support and belief in our skills.

We extend our gratitude to Ahsanullah University of Science and Technology's Department of Computer Science and Engineering for creating a supportive and stimulating atmosphere for our research.

Dhaka
December 2024

Parvez Ahammemd

Tabassum Tara Lamia

Fatin Ishraq

Shawon Kumar Modak

ABSTRACT

Knowledge Graphs (KGs) have gained significant attention in academia and industry for their ability to structure information semantically and provide a formal understanding of the world. This study focuses on their application across multiple domains, including cybersecurity and the medical sector, where they enable various tasks such as question-answering, recommendation systems, and information retrieval. Despite their widespread adoption, there has been limited systematic research on their implementation in domain-specific contexts.

This research provides a comprehensive analysis of the application of KGs, encompassing topics such as knowledge graph representation learning, knowledge acquisition and completion, and domain-specific knowledge-aware applications. By investigating the construction and utility of AISecKG and DRKG datasets, the study demonstrates how KGs can simplify complex concepts and uncover latent relationships. Advanced graph embedding models—such as TransE, RotatE, and ComplEx—were evaluated using key metrics like Mean Rank and HITS@k. Visualization techniques, including t-SNE, were employed to gain deeper insights into the structure and relationships within the datasets.

The findings emphasize the transformative potential of KGs in enhancing decision-making, pattern recognition, and domain-specific knowledge discovery. Additionally, the research identifies challenges such as scalability, data heterogeneity, and ontology evolution, which require further exploration. This study not only highlights the current state of KGs but also provides a curated collection of datasets and tools, setting the stage for future advancements and interdisciplinary innovation in the field.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Objective

The primary objective of this study is to conduct an in-depth investigation into the application of knowledge graphs (KGs) across diverse knowledge domains. Recognizing the broad applicability of KGs across various sectors, this research specifically zeroes in on the unique aspects and demands of employing KGs in particular knowledge areas. The focus is to uncover the distinct challenges, advantages, and methodological approaches necessary for developing and implementing KGs that are custom-made for specific domains. Through thorough reviews of existing literature, detailed case studies, and examples of real-world applications, this research intends to reveal patterns, trends, and domain-specific strategies for utilizing KGs effectively. The goal is to outline best practices, methodologies, and essential tools that can aid in leveraging KGs for domain-specific objectives such as knowledge organization, information extraction, and enhanced decision-making processes. Ultimately, this study aims to offer insightful and actionable recommendations for academics, industry practitioners, and domain experts interested in maximizing the benefits of KGs for domain-centric innovations, facilitating richer collaboration, and promoting advanced knowledge discovery in their respective areas of focus.

1.2 Motivation

The motivation for this research is rooted in the changing world of data use and management, where knowledge graphs (KGs) emerge as a key technology leading us into a new age of data understanding. In a world flooded with data, marked by its immense volume and complexity, old ways of managing data are no longer enough. Knowledge graphs offer a strong solution for dealing with this complexity. They provide a solid structure for linking

detailed data relationships and meanings, offering a clearer and more connected way to analyze data.

This research is driven by the critical need to manage the vast amounts of data that fill many industries, seeking to unlock the full potential of knowledge graphs in the face of growing data amounts, scattered data sources, and the need for quick, useful insights. It aims to lay down the basics of knowledge graphs for anyone grappling with these current data challenges, sharing knowledge on how they can be applied and developed, and their power to transform how we handle data.

At its core, this study is inspired by the potential of knowledge graphs to redefine knowledge management and intelligent systems. Amidst growing demands for applications capable of sophisticated data interpretation, reasoning, and learning, knowledge graphs emerge as a pivotal enabler. Through an examination of their implementation across different scenarios, this research seeks to distill essential practices, chart the course of ongoing developments, and spotlight opportunities for breakthroughs in knowledge organization, searchability, and decision-making tools.

Additionally, this investigation is driven by a commitment to enhancing data-driven decision-making across varied sectors. In today's competitive and complex environment, the strategic importance of informed and agile decision-making cannot be overstated. This study aims to elucidate how knowledge graphs can refine these decision-making processes, equipping organizations to navigate risks, seize opportunities, and achieve strategic objectives more effectively.

Ultimately, the motivation for this research is rooted in a vision of a data-empowered future, where knowledge graphs play a critical role in transcending traditional data analysis limitations. By casting a spotlight on the multifaceted applications and impacts of knowledge graphs, the research seeks to stimulate innovation, bolster informed decision-making, and contribute to the progress toward an advanced, data-centric world across a spectrum of domains and industries.

1.3 Thesis Outline

In this thesis, we aim to explore the landscape of knowledge graphs (KGs) and their applications across various domains, focusing on the theoretical foundations, methodologies, and the diverse ways KGs are utilized to address domain-specific challenges. Through a literature review and analysis of case studies, we will delve into how KGs enhance knowledge management, information retrieval, and decision-making across fields such as healthcare, cybersecurity, and environmental science, among others.

Our exploration will center on the architecture of KGs, including their construction, data encapsulation, and semantic relationships, without deploying any new KGs. Instead, we will conduct an analytical review of existing KGs to understand their strategic applications and the outcomes achieved in different sectors.

A key part of our thesis will be to compare and contrast KG applications across domains, identifying benefits, limitations, and best practices. This analysis aims to shed light on KGs' role in promoting data-driven decision-making and innovation, synthesizing insights into future research directions and practical implications for leveraging KGs in domain-specific knowledge discovery.

Ultimately, this thesis seeks to provide a succinct overview of KG technology, its current applications, and potential future developments, contributing valuable insights for scholars, practitioners, and those interested in the multidisciplinary applications of knowledge graphs.

1.4 Synopsis

In this chapter, we have discussed the motivation behind our work. Then we defined the Knowledge Graph. In the next chapter, we will focus on the background study.

Chapter 2

Background Study

2.1 Overview of Knowledge Graph

Human knowledge provides a formal understanding of the world. Incorporating human knowledge is one of the research directions of artificial intelligence(AI) [3]. A knowledge graph is a structured representation of facts, consisting of entities, relationships, and semantic descriptions. The structured information has brought important possible solutions for many tasks including question answering, recommendation, and information retrieval. In 2012, Google proposed a new technology, called Knowledge Graph, to use semantic knowledge in web searches [4]. Google's knowledge graph is used to identify and disambiguate entities in text, to enrich search results with semantically structured summaries, and to provide links to related entities in exploratory search, to improve the ability of the search engine and enhance the search experience of users. The research on knowledge graphs in four aspects: KRL(Knowledge Representation Language) , knowledge acquisition, temporal knowledge graphs, and knowledge-aware applications [3]. Knowledge Representation Learning is a critical research issue of the knowledge graph, which paves the way for many knowledge acquisition tasks and downstream applications. Knowledge Acquisition tasks are divided into three categories, i.e., KGC(Knowledge Graph Construction), relation extraction, and entity discovery. Temporal Knowledge Graphs incorporate temporal information for representation learning [3]. Knowledge-aware applications include natural language understanding (NLU), question-answering, recommendation systems, and miscellaneous real-world tasks. Knowledge graph hosts a large research community and has a wide range of methodologies and applications. Knowledge graphs have a great ability to provide semantically structured information and important advancements in applying such ability to specific domains have been made in recent years [4]. A knowledge graph may identify new relationships that were not present in the initial knowledge graph.

2.2 Study of Existing Papers

A Review on Application of Knowledge Graph in Cybersecurity [5] In the paper, Zhihao Yan et al. discuss the growing complexity of cybersecurity challenges and the escalating need for advanced solutions to analyze and counteract cyber threats effectively. They highlight the potential of knowledge graphs in cybersecurity, presenting them as a powerful tool for integrating, analyzing, and visualizing massive, fragmented, multi-source heterogeneous data. The authors review the development of knowledge graph technology and its applications in cybersecurity, emphasizing its role in enhancing threat intelligence, vulnerability assessment, and security incident response, thereby improving the overall cybersecurity posture.

Developing an Ontology of the Cyber Security Domain [6] In the paper, Leo Obrst et al. explores the development of a Cyber ontology aimed at integrating data across various sources in the cybersecurity domain. Initially focusing on malware, the effort leverages existing knowledge from the Malware Attribute Enumeration and Characterization (MAEC) language. The methodology adopted is a "middle-out" approach, combining top-down analysis for understanding end-user semantics with bottom-up analysis for data source semantics. The goal is to create a modular ontology framework that spans upper, mid-level, and domain-specific ontologies, facilitating precise searches and complex queries in cybersecurity. This study underscores the importance of foundational, utility, and domain-specific ontologies in creating a comprehensive cyber ontology, aiming to enhance data integration and security analysis capabilities.

Developing an Ontology for Cyber Security Knowledge Graphs [7] Michael Iannacone and his team present an ontology developed for cyber security knowledge graphs. Their work aims to create an organized schema that incorporates information from a broad spectrum of structured and unstructured data sources relevant to cyber security. They compare their ontology with previous efforts, discussing its strengths, limitations, and areas for future improvement. This initiative is part of the STUCCO project, which seeks to provide an integrated data resource for cyber security professionals, combining publicly available datasets with internal data such as net flows and IDS alerts. The ontology is designed to make cybersecurity information more accessible and useful for both human analysts and automated systems.

Knowledge Graphs for Cybersecurity: A Framework for Honeypot Data Analysis [8]

Yevonnael Andrew et al. provide a complex approach to improve cybersecurity research by building knowledge graphs. They do this by combining machine learning (ML) and natural language processing (NLP) approaches with graph databases like Neo4j. Their methodology, which focuses on honeypot data analysis, includes data collecting, preprocessing, en-

tity extraction, and knowledge graph building that enriches and organizes data from other sources like AbuseIPDB and ip-api. This thorough technique allows for the geographical display of attack origins in addition to facilitating a deeper understanding of cybersecurity concerns through the analysis of attack sequences and activity summaries from particular IP addresses.The framework's ability to greatly increase threat detection and provide incisive analysis of attacker activity is demonstrated in the paper's review, which also offers suggestions for future developments in cybersecurity data analytics.

Disease ontologies for knowledge graphs [9] Natalja Kurbatova and Rowan Swiers discuss the challenges and solutions related to integrating multiple disease ontologies into a single, coherent knowledge graph in the biomedical domain. They highlight the diversity and range of existing disease ontologies, each with its own structure and hierarchy, and propose a knowledge graph solution that uses disease ontology cross-references. This allows for flexible data integration and the ability to switch between ontology hierarchies easily. Their work, leveraging the Grakn core, presents an elegant solution to the problem of multiple disease ontologies, facilitating the construction of biomedical knowledge graphs by enabling straightforward ontological queries and data integration.

Construction of Knowledge Graph of Neurodegenerative Diseases Based on Ontology [10] In the paper by Zhuocun Wu et al., the authors embark on the construction of a knowledge graph of neurodegenerative diseases based on ontology, leveraging clinical guidelines, medical textbooks, and medical websites as knowledge sources. They focus on structuring the vast, complex domain of neurodegenerative diseases—like Alzheimer's and Parkinson's—into an accessible and analyzable format. The constructed ontology comprises 10 concept types, 17 object attributes, and 7 data attributes, encapsulating the domain's critical elements. This foundational work sets the stage for the knowledge graph's development, which incorporates 2012 disease-related entities, aiming to enhance early diagnosis and intervention strategies by structuring and making sense of the sprawling data in this field.

Building the Knowledge Graph from Medical Conversational Text Data and its Applications [11]

Rugwedi Kulkarni and Yashodhara Haribhakta use Named Entity Extraction and Knowledge Graph Embedding, two novel Natural Language Processing (NLP) approaches, to creatively arrange medical conversational data into a structured knowledge graph. Their method captures important medical data for improved diagnostic and treatment procedures by transforming unstructured conversations between physicians and patients into a graph-based format. Their study improves medical data structure and lays the groundwork for its application in predictive healthcare analytics by using a mathematical approach to validate tuples based on entity occurrences. The review emphasizes the importance of the created

medical knowledge graph in medical data analysis and use by concentrating on its accuracy and possible applications.

A Graph-based Approach for Integrating Biological Heterogeneous Data Based on Connecting Ontology [12]

In 2021, Shilong Zhang et al. developed a graph-based approach that creates a linking ontology to facilitate the integration of various biological information. This method makes it easier to bring disparate biological ontologies together, improves data annotation, and makes it possible to build an extensive biological knowledge graph. This paper showcases the efficiency and practicality of the technique through the use of case studies in rice gene-phenotype and lactobacillus data integration, which are implemented on a platform to illustrate its usefulness. Their innovative pipeline simplifies the integration process, making it simple to annotate various data sources and produce an excellent biological knowledge tree automatically. By offering a structured, unified picture of diverse biological data, this considerably enhances the science of bioinformatics and paves the way for future developments in biological data analysis and application.

Domain-specific Knowledge Graphs: A Survey [2]

This paper, "Domain-specific Knowledge Graphs: A Survey" by Bilal Abu-Salih, provides a comprehensive analysis and definition of domain-specific Knowledge Graphs (KGs), examining state-of-the-art KG construction approaches across seven knowledge domains. Acknowledging the absence of a unified definition for domain-specific KGs and the imperfections in current KG construction methods, it identifies various limitations, challenges, and potential research directions. By reviewing over 140 papers, the study not only categorizes KGs into healthcare, education, ICT, sciences and engineering, finance, society and politics, and travel but also highlights each domain's unique KG usage, construction algorithms, resources, embedding techniques, evaluation measures, and limitations. The survey underlines the necessity for better data quality, privacy, credibility standards, enhanced semantic expansion, improved KG construction algorithms, and the incorporation of time-awareness in KGs. It emphasizes the significance of KG evaluation, the computational challenges of large-scale KGs, the need for domain-specific KG reasoning, and the importance of making domain-specific KGs publicly available. This survey aims to motivate further research in domain-specific KGs, urging the integration of KGs with blockchain technologies, addressing data quality and interoperability, and encouraging the development of methodologies for automated KG construction.

A Survey on Application of Knowledge Graph [4]

In "A Survey on Application of Knowledge Graph," Xiaohan Zou conducts an in-depth examination of the flourishing interest in knowledge graphs (KGs) from both the industrial and academic perspectives, especially highlighting their significance in semantically structuring

information. This paper emphasizes the unprecedented potential KGs hold in fostering more intelligent machine architectures through their semantic information structuring capability. Despite the widespread adoption of KGs across various "Big Data" applications since their popularization by Google in 2012, Zou notes a gap in systematic reviews focusing on KG applications across different domains. This survey aims to fill that gap by offering an extensive overview of KG applications, particularly in question answering, recommendation systems, information retrieval, and other specific domains like medical, financial, cybersecurity, news, and education. Zou acknowledges the progress made in utilizing KGs but also points out areas that require further exploration, such as data quality, privacy, semantic expansion, and the need for domain-specific reasoning within KG frameworks. This paper sets a foundational survey that encourages further research and application of KGs across various fields.

2.3 Comparative Analysis of existing papers

2.3.1 Cybersecurity Applications of Knowledge Graphs

Knowledge graphs (KGs) have emerged as a powerful tool in the field of cybersecurity, enabling improved data integration, visualization, and threat analysis. They allow for the seamless organization of diverse datasets, helping to identify patterns and relationships critical for cyber defense. Table 2.1 provides an overview of key contributions and applications of knowledge graphs in cybersecurity research.

Table 2.1: Cybersecurity Applications of Knowledge Graphs

Reference	Authors	Focus Areas	Key Contributions
[5]	Zhihao Yan et al	Application of KGs in analyzing and counteracting cyber threats.	Emphasized KGs as powerful tools for integrating, analyzing, and visualizing data to enhance threat intelligence and security incident response.
[6]	Leo Obrst et al.	Development of a comprehensive ontology for cybersecurity, focusing on malware.	Highlighted the role of structured ontology in effective data integration and security analysis.
[7]	Michael Iannacone et al.	Ontology development for cybersecurity KGs to organize and make information more accessible.	Discussed the importance of integrating various data sources for a comprehensive overview of cybersecurity.
[8]	Yevonnael Andrew et al.	Combining ML, NLP, and graph databases for honeypot data analysis in cybersecurity research.	Demonstrated a methodology that enriches and organizes data for improved threat detection and attacker activity analysis through KGs.

2.3.2 Medical Applications of Knowledge Graphs

Knowledge graphs (KGs) have proven to be valuable in the medical field by enabling structured representation, integration, and querying of complex medical and biological data. They facilitate enhanced diagnostics, treatment planning, and research by transforming unstructured medical data into an organized, accessible format. The able 2.2 outlines various research efforts and contributions showcasing the applications of KGs in the medical domain.

Table 2.2: Medical Applications of Knowledge Graphs

Reference	Authors	Focus Areas	Key Contributions
[9]	Natalja Kurbatova and Rowan Swiers	Integrating multiple disease ontologies into a coherent KG for the biomedical domain.	Proposed a solution for easy data integration and query across diverse and complex disease data through disease ontology cross-references
[10]	Zhuocun Wu et al.	Construction of a KG for neurodegenerative diseases based on ontology.	Focused on structuring complex medical data to aid in early diagnosis and intervention, demonstrating the potential of KGs in organizing complex medical information.
[11]	Rugwedi Kulkarni and Yashodhara Haribhakta	Transforming medical conversational data into a KG using NLP techniques.	Highlighted the significance of structuring unstructured data for enhanced diagnostic and treatment processes, improving the medical data structure for predictive healthcare analytics
[12]	Shilong Zhang et al.	Creating a linking ontology to integrate various biological information.	Showcased a method for simplifying the integration process of disparate biological data sources into a unified KG, enhancing bioinformatics research through a structured, unified view of data.

2.4 Knowledge Graph-Based Applications

Knowledge graphs have generated significant attention in both industry and academia. They offer semantically structured data that computers can interpret, a feature widely seen as holding great potential for advancing the development of more intelligent machines, according to many experts [4]. Research on knowledge graphs can be classified into two categories, research on construction techniques of knowledge graph and application of knowledge graph [4]. A taxonomy of the application fields of the knowledge graph is i.e. Question Answering, Recommender System, Information Retrieval, Domain Specific, and Other applications. In our paper, we introduced domain-specific applications of knowledge graphs. The construction of a Domain-specific Knowledge Graph has great application value in intelligent search, intelligent question and answer, intelligent recommendation, and other information services. Various domain-based KGs are Healthcare, Education, ICT, Science and Engineering, Finance, Society and Politics, and Travel [4].KGs offer the healthcare sector

technical means to derive meaningful insights from voluminous and heterogeneous healthcare data. Knowledge Graphs have been widely used to improve Information and Communication Technology such as Cybersecurity, software development, and Telecommunication. We leveraged the advantages of utilizing a Knowledge Graph in the cybersecurity domain. Cyber-attack activities are complex and ever-changing, posing severe challenges to cybersecurity personnel. Introducing knowledge graphs into the field of cybersecurity helps depict the intricate cybersecurity landscape [13]. Domain-specific KGs are used to tackle several real-life problems.A new large-scale graph-based reasoning method for malware detection is presented, called Polonium, presents the challenge of malware detection as one of graph mining and inference. Polonium uses the Belief Propagation technique to give each file a confidence score.To detect malicious software an adaptive algorithm 'Aesop' is built.Aesop determines a file's maliciousness by analyzing variations in its location hashes. [5]

2.5 Ontology Design

Ontology is the mathematical representation of knowledge in a particular sector, encompassing definitions of classes, relations, functions, and other objects. Enabling the reuse of domain knowledge was one of the driving forces behind the recent surge in ontology research. An ontology is a specification of a conceptualization. It defines a common vocabulary for researchers who need to share information in a domain. Ontologies are used in artificial intelligence, the semantic web, software engineering, biomedical informatics, and information architecture as a form of knowledge representation about the world or some part of it. The Knowledge Representation Ontology uses a framework of distinctions to automatically generate a hierarchy of categories, rather than a fixed one [7]. One of the main roles of ontologies is supporting knowledge-sharing activities. Classes are the focus of most ontologies. Classes describe concepts in the domain. For example, a class of wines represents all wines. Developing an ontology is similar to specifying a collection of information and organizing its format, making it accessible for utilization by other software programs [14]. An ontology together with a set of individual instances of classes constitutes a knowledge base. There is no one "correct" way or methodology for developing ontologies [14]. There are always viable alternatives. Ontology design is a creative process and no two ontologies designed by different people would be the same [14]. Ontologies can be grouped into three broad categories: upper, mid-level, and domain ontologies, according to their levels of abstraction [6].

- Upper ontologies are high-level, domain-independent ontologies that provide common knowledge bases from which more domain-specific ontologies may be derived. Standard upper ontologies are also referred to as foundational or universal ontologies.

- Mid-level ontologies are less abstract and make assertions that span multiple domain ontologies. These ontologies may provide more concrete representations of abstract concepts found in the upper ontology. There needs to be a clear demarcation point between the upper and mid-level. Mid-level ontologies also encompass the set of ontologies that represent commonly used concepts, such as Time and Location. These commonly used ontologies are sometimes referred to as utility ontologies.

- Domain ontologies specify concepts particular to a domain of interest and represent those concepts and their relationships from a domain-specific perspective. Domain ontologies may be composed by importing mid-level ontologies. They may also extend concepts defined in mid-level or upper ontologies.

The scope of our ontology will extend to cybersecurity and medical diseases. It's pertinent to remember that an ontology is a model of reality, and its concepts should reflect that reality. Developing the class hierarchy and defining properties of concepts (slots) are the two most important steps in the ontology design process [9]. If a class has only one direct subclass there may be a modeling problem or the ontology is not complete. The class hierarchy represents an "is-a" relation: a class A is a subclass of B if every instance of A is also an instance of B [14]. The ontology should not contain all the possible information about the domain. Similarly, the ontology should not contain all the possible properties of and distinctions among classes in the hierarchy [14].

2.5.1 Ontology and Knowledge Graph

Knowledge representation research customarily adopts a graph-based view of data, and bases most of its semantics on similar concepts [15]. Graphs can be stored in many formats but in the use of query languages. The knowledge graphs in modern applications are characterized by several properties that together distinguish them from more traditional knowledge management paradigms [15]:

- Normalisation: Information is decomposed into small units of information, interpreted as edges of some form of graph.

- Connectivity: Knowledge is represented by the relationships between these units.

- Context: Data is enriched with contextual information to record aspects such as temporal validity, provenance, trustworthiness, or other side conditions and details.

Knowledge graphs are often used in data integration – graphs are well suited for capturing heterogeneous(diverse or dissimilar elements or parts) data sources and their relationships

–, and it is natural to retain basic information regarding, e.g. The source and trustworthiness of a particular piece of data [15]. On the other hand, An ontology together with a set of individual instances of classes constitutes a knowledge base. Ontology serves as the blueprint for constructing a knowledge graph. A knowledge graph is a combination of ontology and data. An ontology is a framework for a knowledge graph. As a framework, an ontology consists of a person, data, classes, properties, relations, and axioms. An ontology is a formal framework that is used to define concepts, organize knowledge, and explain connections within a particular topic. Usually, it has a hierarchy of ideas along with the characteristics of each. The ontology acts as a schema layer for building a knowledge network by defining terms and characteristics pertinent to a certain area. In contrast, a knowledge graph is a particular instance of knowledge within a domain, shown as a graph with relationships as edges and things as nodes. It combines data from several sources into a logical framework. The knowledge graph, when kept in a database management system such as Neo4j, makes it easier to store, retrieve, and analyze linked data efficiently [10].

In conclusion, the knowledge graph integrates data from several sources into a cohesive graph structure, representing the actual instances of knowledge, whereas the ontology supplies the basic framework for knowledge organization within a domain

2.5.2 Ontology for a Knowledge Graph-Based Application

Sharing a common understanding of the structure of information among people or software agents is one of the more common goals in developing ontologies [14].An ontology is a conception that is clearly defined. By establishing a collection of representational ideas, we can explain the ontology of a program. Each ontology defines a set of classes, relations, functions, and object constants for some domain of discourse [14]. Most prominent ontology languages are unable to express even the simplest types of relationships on knowledge graphs [6]. To the development of ontology, existing ontologies can be used. The key to ontology development lies in an understanding of the cyber domain. STUCCO is an ontology that can represent the cyber security domain, allowing information to be combined from as many sources as possible within the domain [7]. Several foundational ontologies could be considered for use in the cyber ontology. BFO(Basic Formal Ontology), is an upper-level ontology that serves as a basis for the development of more specialized ontologies [6]. Finding basic categories and relationships that are shared by several areas is its main objective.

Three degrees of granularity are distinguished: Domain Ontologies (built upon BFO-Basic to handle domain-specific ideas), BFO-Core (core top-level classes and relations), and BFO-Basic (extends BFO-Core with more particular categories). BFO emphasizes the portrayal of things and processes as they occur in reality, using a realism-based methodology. It seeks to provide philosophical and scientific research with a shared foundation. Applications of

BFO have been made in many fields, such as biomedical informatics, where ontologies for the representation of biological entities, clinical data, and medical knowledge have been developed. [4]

Open Biomedical Ontologies (OBO) [16] aims to build and manage a library of controlled vocabularies for a variety of biological and medical disciplines. Examples of resources like the Gene Ontology (GO), which contains vocabularies for cellular components, molecular activities, and biological processes, demonstrate how the OBO library is meant to be used collaboratively across disciplines. These ontologies are organized as entities in graph theory, with concepts connected by relations like "is_a" and "part_of." Improving these vocabularies' ontological and logical rigor is a major objective of the OBO initiative as it will promote increased interoperability and the capacity to assist automated reasoning about biological and medical phenomena.The significance of organized, interoperable data for improving healthcare and biological research is increasingly acknowledged, as seen by this endeavor.

2.6 Knowledge Graph Construction

A knowledge graph is a way of storing data that resulted from an information extraction task. Many basic implementations of knowledge graphs make use of a concept we call triple, that is a set of three items(a subject, a predicate and an object) that we can use to store information about something.

Researchers have created various knowledge graphs for various domains. A few notable Knowledge graphs are Yago, DBPedia, WikiData, FreeBase, Microsoft Satori, Google KG, etc [15]. Even Though these knowledge graphs have millions of entities and facts they all suffer from problems of incompleteness [17]. To tackle this issue, knowledge graph completion tasks can be used [17]. Cyber security knowledge graph generated from actual Windows executable files [17]. Knowledge graphs provide a structured and semantic approach to representing data, which can be particularly useful in the complex and dynamic field of cybersecurity and medical science. Knowledge reasoning involves deducing new knowledge or identifying incorrect information based on the existing knowledge within a knowledge graph.

- 1. Level of Knowledge Extraction:
  - Entity-level: In this case, the text's entities are recognized and categorized. Generally speaking, entities are nouns that describe things, people, locations, and so forth.
  - Named Entity Recognition (NER): This procedure entails locating and categorizing items inside a text according to predetermined standards, including names

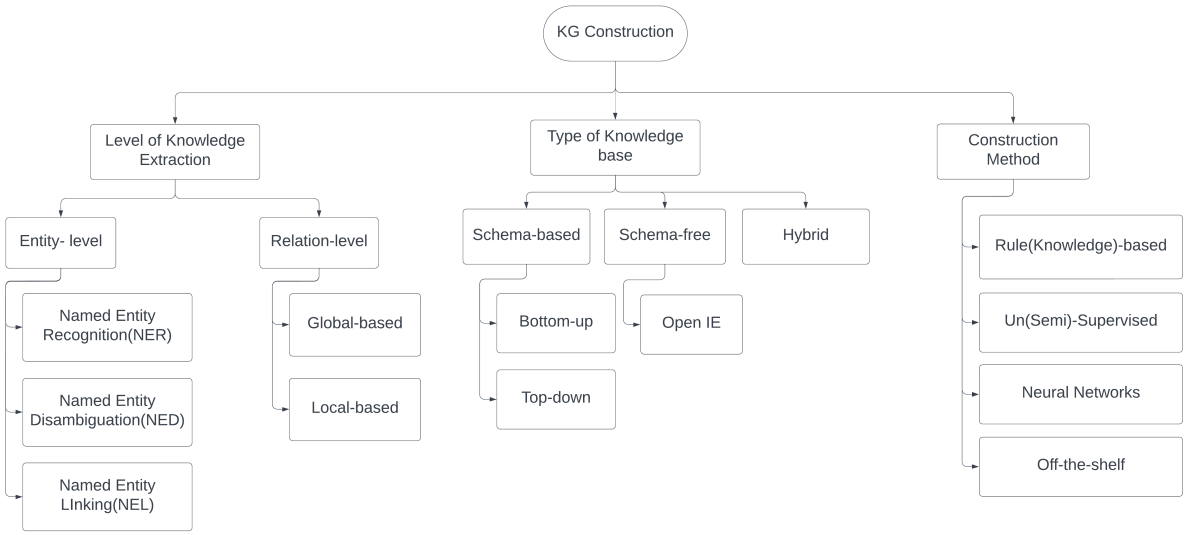

Figure 2.1: Domain-specific Knowledge Graphs: A survey by Bilal Abu-Salih [2]

of people, places, businesses, amounts, percentages, and times expressed.

- Named Entity Disambiguation (NED):It resolves textual references to their distinct representation in a knowledge base (e.g., distinguishing between 'Jordan' the nation and 'Michael Jordan' the football player).
- Named Entity Linking (NEL):The process of connecting entity references in text with the relevant entities in a knowledge base.

- 2. Relation-level:
  - Global-based: This method considers the global context while examining the complete corpus of text to detect relations.
  - Local-based: In order to determine the relationships between things, this relies on the immediate surroundings around them.

- 3. Type of Knowledge base:
  - Schema-based: With this method, the types of entities and connections are predetermined and a knowledge graph is built using a specified schema.(Bottom-up/ Top-down)
  - Schema-free: It is sometimes referred to as Open Information Extraction, or Open IE, as it operates independently of a preset schema. Rather, it automatically extracts entities and associations from unstructured text.
  - Hybrid: To take use of the advantages of both, this methodology blends schema-based and schema-free techniques.

- 4.Construction Method:
  - Rule(Knowledge)-based: This technique extracts entities and relations from text by using pre-established linguistic or expert-generated criteria.
  - Un(Semi-)Supervised: These are machine learning techniques that need little or no assistance from humans. Semi-supervised approaches employ a little amount of labeled data to direct the learning process, whereas unsupervised approaches do not utilize any annotated data at all.
  - Neural Networks: These are models that mimic how the human brain learns to accomplish tasks by looking at examples. They are particularly helpful in identifying intricate patterns in data.
  - Off-the-shelf: This means that instead of creating new algorithms or models, one may create a knowledge graph by utilizing pre-existing tools and models created by others.

2.7 Key Terms in Knowledge Graph Technology

This section provides definitions for essential terms related to the technology and application of Knowledge Graphs (KGs), helping to clarify concepts that are fundamental to understanding how KGs function and are utilized across various domains.

Entity An entity in a KG represents a real-world object or concept, such as a person, place, organization, or any specific item that can have attributes and can be connected to other entities.

Relationship A relationship (or relation) in a KG denotes how two entities are connected. It illustrates the type of link between entities, for example, "works for," "located in," or "is a part of."

Triple A triple is the fundamental data structure in a KG, consisting of a subject (entity), predicate (relationship), and object (entity or attribute value). It forms a statement expressing a fact or knowledge, such as (Paris, is the capital of, France).

Ontology In the context of KGs, an ontology defines a set of concepts, categories, and relationships within a domain. It provides a formal, shared vocabulary for entities and their relationships, enabling consistent knowledge representation.

RDF (Resource Description Framework) RDF is a standard model for data interchange on the Web, allowing structured and semi-structured data to be mixed, exposed, and shared across different applications. It uses triples to represent data.

SPARQL (SPARQL Protocol and RDF Query Language) SPARQL is a query language and protocol for querying and manipulating RDF data. It enables complex queries over KGs to retrieve and manipulate the stored information.

Embedding In KGs, embedding refers to the representation of entities and relationships in a continuous vector space. This facilitates the application of machine learning algorithms by translating the discrete structures of KGs into forms that algorithms can process.

Link Prediction Link prediction in the context of KGs is the task of predicting whether a relationship (link) should exist between two entities based on the existing data in the KG.

Ontology Alignment Ontology alignment involves finding correspondences between semantically related entities of different ontologies, facilitating interoperability among heterogeneous data sources or KGs.

Semantic Web The Semantic Web is an extension of the World Wide Web through standards set by the World Wide Web Consortium (W3C). It enables data to be shared and reused across application, enterprise, and community boundaries, making it possible to connect data across the web through semantic relationships.

2.8 Graph Embedding Methodology

Graph embedding is a technique used to transform a graph's nodes, edges, or subgraphs into low-dimensional vector representations. These embeddings capture the structural properties and semantic information of the graph, which can then be used for various downstream tasks such as node classification, link prediction, and clustering.

It refers to the process of mapping a graph's nodes and/or edges into an infinite vector area, such that the graph's topological structure is preserved in the embedding. The goal is to represent the graph in a way that captures both the global structure (such as communities or clusters) and the local structure (such as direct neighbors and local connectivity) of the graph.

2.8.1 Usage of Graph Embedding

Graph embedding is particularly useful because it enables the application of machine learning techniques to graph-based data. Graph data is inherently non-Euclidean, meaning traditional machine learning algorithms that rely on vectorized data (like decision trees or linear classifiers) cannot be directly applied. By converting graphs into vector representations, graph embedding bridges this gap and allows the use of sophisticated algorithms for

problems like:

- Node classification: Predicting the labels or categories of nodes based on their features and graph structure.
- Link prediction: Predicting the presence or absence of edges between pairs of nodes, useful in recommendation systems or social network analysis.
- Graph clustering: Grouping similar nodes or subgraphs into clusters based on their structural properties.
- Anomaly detection: Identifying unusual or unexpected patterns in graph structures, often used in fraud detection or network security.
- Graph visualization: Representing large-scale graphs in low-dimensional spaces for easier visualization and analysis.

2.8.2 Types of Graph Embedding

There are several different approaches to graph embedding, depending on whether the focus is on node embeddings, edge embeddings, or the entire graph. Some of the most common types include:

- **Node Embedding**: This technique focuses on representing individual nodes of the graph as vectors. The idea is that nodes with similar structural properties (such as nodes connected by edges or nodes belonging to the same community) will have similar embeddings. Popular methods for node embedding include DeepWalk, Node2Vec, and GraphSAGE.
- **Edge Embedding**: Instead of embedding individual nodes, edge embedding methods aim to learn representations of the relationships (edges) between nodes. Methods like TransE, DistMult, and RotatE fall into this category.
- **Graph-Level Embedding**: This method embeds the whole graph in a low-dimensional vector space.. This is particularly useful in tasks such as graph classification. Graph neural networks (GNNs) are often used for learning graph-level embeddings.

2.8.3 Challenges in Graph Embedding

While graph embedding has proven useful for many applications, there are several challenges involved:

- **Scalability**: Graphs, especially in real-world scenarios, can be very large and contain millions of nodes and edges. Efficiently generating embeddings for such large graphs can be computationally expensive.
- **Graph Heterogeneity**: Real-world graphs can be heterogeneous, containing different types of nodes and edges. Designing embeddings that can capture the full complexity of such graphs remains a challenging task.
- **Dynamic Graphs**: Many graphs change over time, and capturing temporal dynamics is an important aspect of graph embedding. Most embedding methods are designed for static graphs, which limits their applicability to real-world dynamic networks.
- **Preserving Structural Properties**: Ensuring that the graph's structural properties (e.g., connectivity, hierarchy) are well-preserved in the embeddings is a difficult task.

By transforming graphs into continuous vector representations, graph embedding provides a powerful tool for extracting useful information and solving complex problems across diverse domains.

2.9 Graph Embedding Models

To extract meaningful vector representations from the constructed graph, the following graph embedding models are explored:

2.9.1 TransE

TransE [18] is a foundational translational embedding model designed for knowledge graphs. It represents entities and relations as vectors in a continuous vector space. The core idea behind TransE is to interpret relations as translations acting on entity embeddings. For a given triplet (h, r, t), where h is the head entity, t is the tail entity, and r is the relation, TransE enforces the following property:

$$\mathbf{h} + \mathbf{r} \approx \mathbf{t} \tag{2.1}$$

This implies that the embedding of the head entity $\mathbf{h}$, when translated by the relation vector $\mathbf{r}$, should be close to the embedding of the tail entity $\mathbf{t}$. The closeness is measured using a distance function, typically the L_1 or L_2 norm:

$$d(\mathbf{h} + \mathbf{r}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_p, \quad p \in \{1, 2\}. \tag{2.2}$$

Optimization Objective The model's training objective aims to minimize the distance for valid triplets (h, r, t) while maximizing it for corrupted triplets (h', r, t'). This is achieved through a margin-based ranking loss:

$$\mathcal{L} = \sum_{(h,r,t)\in\mathcal{T}} \sum_{(h',r,t')\in\mathcal{T}'} \left[\gamma + d(\mathbf{h}+\mathbf{r},\mathbf{t}) - d(\mathbf{h}'+\mathbf{r},\mathbf{t}')\right]_+, \tag{2.3}$$

where $\mathcal{T}$ is the set of valid triplets, $\mathcal{T}'$ is the set of corrupted triplets, γ is the margin, and $[\cdot]_+$ denotes the hinge loss function.

Strengths of TransE TransE is computationally efficient due to its simple linear operations and low parameter count, making it suitable for large-scale knowledge graphs. Its interpretability stems from its intuitive translation-based mechanism, which aligns well with the semantics of knowledge graph relations. TransE performs exceptionally well for one-to-one relationships, where the relation uniquely maps head entities to tail entities.

Limitations and Extensions Despite its strengths, TransE has limitations in modeling complex relationships, such as one-to-many, many-to-one, or many-to-many relations. For instance, in a one-to-many relationship, the model struggles to position multiple tail entities equidistantly from a single head entity in the embedding space.

Several extensions of TransE address these shortcomings:

- **TransH** [18] Introduces relation-specific hyperplanes to better capture diverse relation patterns.
- **TransR** [18]: Projects entities into relation-specific spaces to handle heterogeneous entity and relation embeddings.
- **TransD** [18]: Further simplifies relation-specific projections while maintaining flexibility in embedding interactions.

These extensions build on the foundational principles of TransE, offering enhanced modeling capabilities for complex knowledge graph structures.

Applications of TransE TransE has been widely applied in various knowledge graph tasks, including:

- **Link Prediction**: Predicting missing links in knowledge graphs by estimating the plausibility of triplets.

- **Entity Classification**: Leveraging embeddings to classify entities based on their relationships and positions in the graph.
- **Relation Extraction**: Inferring relationships between entities using the learned embeddings and relation vectors.

As a pioneering model, TransE has inspired numerous advancements in knowledge graph embedding research, establishing a robust foundation for subsequent developments in the field.

2.9.2 RotatE

RotatE [19] extends the translational approach by representing relations as rotations in the complex vector space. Unlike TransE, which models relations as translations, RotatE captures relational semantics through rotations, enabling it to model more diverse properties such as symmetry, anti-symmetry, and inversion. For a given triplet (h, r, t), the model enforces:

$$\mathbf{t} = \mathbf{h} \circ \mathbf{r} \tag{2.4}$$

where $\circ$ says multiplication by elements, and $\mathbf{r}$ denotes the rotation in the complex plane, represented as a unit modulus complex number. This representation allows RotatE to encode the relational transformation between entities effectively.

Mathematical Properties RotatE is capable of modeling:

- **Symmetric Relations**: Achieved when $\mathbf{r} = 1$, ensuring $\mathbf{h} \circ \mathbf{r} = \mathbf{h}$.
- **Anti-Symmetric Relations**: Captured through distinct rotations for $\mathbf{h}$ and $\mathbf{t}$.
- **Inversion**: Modeled by the reciprocal of the relation embedding, i.e., $\mathbf{r}^{-1}$.

These properties make RotatE highly expressive and suitable for complex relational structures in knowledge graphs.

Optimization Objective Similar to TransE, RotatE uses a margin-based ranking loss to optimize embeddings. However, the distance function is defined in the complex space as:

$$d(\mathbf{h} \circ \mathbf{r}, \mathbf{t}) = \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|, \tag{2.5}$$

where the modulus of the complex difference is used to compute distances.

Strengths of RotatE RotatE excels in scenarios where relational semantics play a critical role. Its ability to encode diverse relational patterns, including one-to-many and many-to-one, makes it superior to translational models like TransE for tasks requiring nuanced relational reasoning.

Limitations and Extensions Despite its expressiveness, RotatE can face challenges with scalability due to the computational complexity of operations in the complex space. Extensions to RotatE have focused on improving efficiency and generalizability, such as incorporating attention mechanisms and hybrid approaches with real-valued embeddings.

Applications of RotatE RotatE has been effectively applied to tasks such as:

- **Knowledge Graph Completion**: Predicting missing entities and relations by leveraging complex-valued embeddings.
- **Multi-Hop Reasoning**: Exploiting rotational properties to infer multi-step relationships in knowledge graphs.
- **Relational Pattern Analysis**: Analyzing and interpreting relational structures within large-scale graphs.

RotatE represents a significant advancement in embedding models, offering a robust framework for capturing and reasoning about complex relational patterns in knowledge graphs.

2.9.3 ComplEx

ComplEx [20] embeds entities and relations in complex vector spaces, enhancing its ability to capture asymmetric and complex relationships common in real-world knowledge graphs. The scoring function of ComplEx is defined as:

$$\phi(h, r, t) = \mathrm{Re}(\langle \mathbf{h}, \mathbf{r}, \bar{\mathbf{t}} \rangle) \tag{2.6}$$

where Re denotes the real part, $\bar{\mathbf{t}}$ is the complex conjugate of $\mathbf{t}$, and $\langle \cdot \rangle$ is the generalized dot product. By incorporating complex conjugates, ComplEx effectively models directional relationships, distinguishing between (h, r, t) and (t, r, h).

Mathematical Properties ComplEx's use of complex numbers provides unique advantages:

- **Asymmetric Relations**: Captured naturally through the inclusion of complex conjugates.
- **Expressiveness**: Models various relational patterns, including symmetric, anti-symmetric, and hierarchical relations.

These features make ComplEx well-suited for knowledge graphs with diverse and intricate relational structures.

Optimization Objective ComplEx optimizes embeddings using a ranking-based loss function, similar to TransE and RotatE. However, the distance function leverages the real part of the Hermitian inner product, ensuring effective modeling of directional relationships. This approach provides both flexibility and computational efficiency.

Applications of ComplEx ComplEx has been widely applied in tasks such as:

- **Knowledge Graph Completion**: Predicting missing triplets by leveraging its ability to encode asymmetric relationships.
- **Entity Classification**: Classifying entities based on the embeddings learned from the complex vector space.
- **Relational Reasoning**: Exploiting the model's capacity to handle complex relationships for downstream reasoning tasks.

By introducing complex-valued embeddings, ComplEx provides a robust framework for capturing the nuanced relationships present in real-world knowledge graphs.

2.9.4 DistMult

DistMult [21] is a bilinear embedding model that represents entities and relations in a real-valued vector space. The scoring function is defined as:

$$\phi(h, r, t) = \mathbf{h}^T \text{diag}(\mathbf{r})\mathbf{t} \tag{2.7}$$

where diag($\mathbf{r}$) is a diagonal matrix constructed from the relation embedding $\mathbf{r}$. This formulation allows the model to compute the compatibility between entities and relations effectively.

Strengths of DistMult

- **Computational Efficiency**: Due to its simple bilinear structure, DistMult is highly efficient and scales well to large knowledge graphs.
- **Simplicity**: Its straightforward formulation makes it easy to implement and integrate into various tasks.

Limitations of DistMult Despite its strengths, DistMult is limited in modeling asymmetric relationships due to the symmetry of the bilinear form. This restricts its effectiveness for tasks where the directionality of relations is crucial.

Extensions to DistMult Several extensions have been proposed to overcome these limitations:

- **ComplEx**: Introduces complex-valued embeddings to address the symmetry constraint.
- **SimplE**: Extends DistMult by introducing two embeddings for each entity to handle asymmetric relations effectively.

Applications of DistMult DistMult is commonly applied in:

- **Link Prediction**: Identifying missing links in a knowledge graph by evaluating the plausibility of triplets.
- **Entity Ranking**: Ranking entities based on their relationships with other entities in the graph.

While DistMult is a foundational model, the clarity and computational efficacy of DisMult can build the strong baseline for many knowledge graph problems.

2.9.5 RESCAL

RESCAL [**?**] is a knowledge graph embedding model based on tensor factorization. It models relations as matrices, capturing the interactions between entities through bilinear transformations. Unlike translational models like TransE, which represent relations as translations, RESCAL uses tensor decomposition to learn embeddings for both entities and relations,

which allows it to model more complex relationships. For a given triplet (h, r, t), RESCAL defines the scoring function as:

$$f(h, r, t) = \mathbf{h}^\top \mathbf{R}_r \mathbf{t} + b_r \tag{2.8}$$

where $\mathbf{h}$ and $\mathbf{t}$ are the embeddings for the head and tail entities, $\mathbf{R}_r$ is the relation matrix, and b_r is a bias term for the relation r. This bilinear form enables the model to represent complex interactions between entities and relations.

Mathematical Properties RESCAL captures various relational semantics through its tensor factorization approach:

- **Symmetric Relations**: Symmetry is naturally modeled by the relation matrix $\mathbf{R}_r$ being symmetric, i.e., $\mathbf{R}_r = \mathbf{R}_r^\top$.
- **Anti-Symmetric Relations**: Anti-symmetry is captured when the relation matrix is skew-symmetric, i.e., $\mathbf{R}_r = -\mathbf{R}_r^\top$.
- **Transitivity and Other Complex Relationships**: The bilinear model of RESCAL allows it to capture transitive relationships, where the relationship between entities can follow a path or chain of transformations.

These properties make RESCAL a flexible model capable of capturing a variety of relational patterns.

Optimization Objective RESCAL uses a margin-based ranking loss to optimize the embeddings, similar to other knowledge graph embedding models. The loss function aims to maximize the score of correct triples while minimizing the score of negative triples. The distance function used in RESCAL is based on the bilinear scoring function:

$$d(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \|\mathbf{h}^\top \mathbf{R}_r \mathbf{t} - \mathbf{t}_{\text{neg}}^\top \mathbf{R}_r \mathbf{h}_{\text{neg}}\|_1 \tag{2.9}$$

where the negative samples $(\mathbf{h}_{\text{neg}}, \mathbf{r}, \mathbf{t}_{\text{neg}})$ are corrupted versions of the original triples.

Strengths of RESCAL RESCAL's tensor factorization approach allows it to effectively model complex, high-order interactions between entities and relations. The bilinear form provides a rich representation of relational data, making it highly expressive and capable of handling a wide variety of relational patterns, including symmetric, anti-symmetric, and transitive relations. It is particularly well-suited for tasks where relationships between entities require nuanced modeling beyond simple translations.

Limitations and Extensions One of the primary limitations of RESCAL is its scalability. The use of relation-specific matrices introduces a high computational cost, especially in large knowledge graphs with many relations. Additionally, RESCAL struggles with modeling very sparse graphs effectively. To address these challenges, extensions to RESCAL have introduced more efficient training methods, such as using low-rank approximations of the relation matrices or integrating additional techniques like negative sampling.

Applications of RESCAL RESCAL has been widely applied to various knowledge graph-related tasks:

- **Knowledge Graph Completion**: RESCAL is particularly effective in predicting missing entities and relations in a knowledge graph, making it useful for completing incomplete knowledge bases.
- **Link Prediction**: By learning embeddings for entities and relations, RESCAL can predict missing links (i.e., missing triples) in knowledge graphs.
- **Entity and Relation Classification**: RESCAL's embeddings can be used for classifying entities or relations based on their characteristics in the graph.
- **Multi-Hop Reasoning**: The model can be used for inferring multi-step relationships, which is useful in tasks like question answering or reasoning over knowledge graphs.

RESCAL is a powerful model for knowledge graph embedding, offering a rich framework for understanding and predicting complex relational patterns. Despite its challenges with scalability, it remains a popular choice for tasks involving intricate relational data.

2.9.6 Comparison of different models

Table 2.3: Comparison of Knowledge Graph Embedding Models

Criteria	**RotatE**	**Complex**	**TransE**	**DistMult**	**RESCAL**
Model Type	Rotation-based	Bilinear-based	Linear-based	Bilinear-based	Tensor-based
Scalability	High	Moderate	High	High	Moderate
Relation Types	Rotational (complex)	Complex vector space	Translational	Bilinear	Tensor decomposition
Embeddings	Complex-valued	Complex-valued	Real-valued	Real-valued	Real-valued
Interpretability	Good (Rotation in space)	Moderate	Moderate	Moderate	Moderate
Evaluation Metrics	MR, HITS@k	MR, HITS@k	MR, HITS@k	MR, HITS@k	MR, HITS@k
Training Difficulty	Moderate	High	Low	Low	High
Handling Symmetry	Yes	No	No	Yes	Yes

2.10 Evaluation Metrics for Knowledge Graph Embedding Models

To evaluate the performance of knowledge graph embedding models, various metrics are used that assess both the excellence of the taught embeddings and their ability to perform tasks such as link prediction, classification, and entity retrieval. Below, we describe the most commonly used evaluation metrics in detail:

2.10.1 Mean Rank (MR)

Mean Rank (MR) is a metric that measures the average rank of the correct entity in the list of possible entities when predicting the tail or head of a knowledge graph triplet. A lower mean rank indicates better performance, as the correct entity is ranked higher in the prediction list. The rank is calculated as follows:

$$MR = \frac{1}{N} \sum_{i=1}^{N} \text{rank}_i$$

where N is the number of test samples, and rank_i is the rank of the correct entity in the list of candidate entities for the i-th test triplet.

2.10.2 Precision, Recall, and F1-Score

These metrics are widely used in classification tasks and evaluate how well the model can predict positive labels.

- **Precision**: Precision measures the accuracy of positive predictions. It is the ratio of correctly predicted positive instances to the total predicted positives:

  $$\text{Precision} = \frac{TP}{TP + FP}$$

  where TP is the number of true positives and FP is the number of false positives.

- **Recall**: Recall measures the ability of the model to capture all relevant positive instances. It is the ratio of correctly predicted positive instances to the total actual positives:

  $$\text{Recall} = \frac{TP}{TP + FN}$$

  where FN is the number of false negatives.

- **F1-Score**: The F1-score is the harmonic mean of Precision and Recall, offering a balance between the two metrics:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-score provides a single measure of performance when there is a need to balance both Precision and Recall.

2.10.3 ROC Curve and AUC

The Receiver Operating Characteristic (ROC) curve plots the true positive rate (TPR, also known as Recall) against the false positive rate (FPR) for different thresholds. It is used to evaluate the model's ability to distinguish between classes.

- **True Positive Rate (TPR)**: Also known as Sensitivity or Recall, it is the ratio of correctly predicted positives to the total number of actual positives:

$$TPR = \frac{TP}{TP + FN}$$

- **False Positive Rate (FPR)**: It is the ratio of incorrectly predicted positives to the total number of actual negatives:

$$FPR = \frac{FP}{FP + TN}$$

- **AUC (Area Under the Curve)**: The area under the ROC curve (AUC) quantifies the overall ability of the model to discriminate between the positive and negative classes. AUC ranges from 0 to 1, with a higher value indicating better performance. An AUC of 0.5 indicates random performance, while an AUC of 1.0 indicates perfect classification.

2.10.4 Comparison of different common metrics

In evaluating graph embedding models and link prediction tasks, various performance metrics are used to assess their accuracy and effectiveness. These metrics provide insights into different aspects of model performance, such as ranking quality, classification accuracy, and prediction reliability. Table 2.4 provides a concise comparison of common metrics, their primary use cases, and the direction of optimal values (whether higher or lower is better).

Table 2.4: Summary of Common Metrics

Metric	**Use Case**	**Higher/Lower is Better**
Mean Rank (MR)	Link prediction, ranking evaluation	Lower is better
AUC	Binary classification (edge prediction)	Higher is better
Precision, Recall, F1 Score	Link prediction, classification	Higher is better

Chapter 3

Data Analysis and Sample Construction

3.1 Data Analysis

3.1.1 Wikepedia Dataset

[22]With 4,319 rows and one sentence column, the "Building Knowledge Graph" dataset is a great tool for creating knowledge graphs. Every statement, such when Matthew Bannister refers to someone as "the first star producer," reveals important details about other people and their connections. This layout highlights the connections between things in many areas by making it simple to extract and arrange data into a knowledge graph. By utilizing the rich, encyclopedic material of the dataset, such a network would greatly enhance information retrieval and natural language processing applications.

- Samples:
  confused and frustrated, connie decides to leave on her own.
  later, a womanâ€™s scream is heard in the distance.
  christian is then paralyzed by an elder.
  it’s a parable of a woman’s religious awakeningâ€”
  c. mackenzie, and craig vincent joined the cast.
  we just tried to make the film.

Table 3.1: Entities and Attributes

Entity	Type	Attributes
Connie	Person	Confused, Frustrated
Woman (scream)	Person	Involved in screaming (possibly Connie)
Christian	Person	Paralyzed
Elder	Person	Performs paralysis on Christian
C. Mackenzie	Actor	Joined the film cast
Craig Vincent	Actor	Joined the film cast
The film	Object	Described as a parable of religious awakening

Table 3.2: Relationships

Source	Relationship	Target
Connie	Decides to leave	-
Woman	Screams	-
Elder	Paralyzes	Christian
C. Mackenzie	Joins	The film cast
Craig Vincent	Joins	The film cast
-	Attempts to make	The film

3.1.2 AISecKG : Knowledge Graph Dataset for Cybersecurity Education [1]

Key Entities

Based on the analysis of the AISecKG dataset, entities extracted from NER and using domain knowledge, has been created following categories for cybersecurity related entities -

- Concept
- Application
- Role

The different types of entities under each category can be defined as follows:

Entity Metadata /or attributes

- entityID
- entityName

Table 3.3: Entity Categories

entityCategory	entityType	entityName (example)
Concept	feature	private key, cookies, protocol, port
	function	snort rules, hash, XOR
	attack	css, sql injection, spyware
	vulnerability	bad config, weak password
	technique	honeypot, risk assessment
	data	Files, logs, messages, packet
Application	tool	burp, wireshark, snort, sniffer
	system	linux, server, host
	app	browser, webapp
Role	attacker	black hat, attack host
	securityTeam	security engineer, white hat, ethical hacker, network admin
	user	employee, user

- entityType
- entityCategory

Relations

Based on the data analysis we identified the following relations -

Table 3.4: Relations

relation	Example
has_a	Snort has_a Packet Decoder
is_a	Trojan is_a malware
can_analyze	Packet decoder can_analyze header anomaly
can_expose	Home network can_expose Land Attack
can_exploit	Attack host can_exploit TCP SYN Packet
implements	Snort rules implements ICMP rules
uses	Team defense uses home network

Property Graphs

We plan to use property graphs because they allow us to add more attributes to the edges. We can add an attribute 'action' to the edge. That will help in explaining any action or specific detail associated with that triple.

For Example:

Table 3.5: Property Graphs

Entity	Relation	Entity	Action
IDS	is_a	Intrusion Detection System	means
Snort	can_analyze	network attacks	detect
Snort	can_analyze	previous attack pattern	match
Preprocessor	can_analyze	HTTP anomaly	verify

Triple Schema

The dataset or triples follows below schema:

Table 3.6: Triple Schema

Schema Edges
(system, can_expose, attack)
(system, can_expose, vulnerability)
(app, has_a, feature)
(tool, part_of, tool)
(tool, has_a, function)
(tool, can_analyze, function)
(tool, can_analyze, apps)
(tool, can_analyze, vulnerability)
(tool, implements, technique)
(tool, has_a, feature)
(function, can_expose, attack)
(function, has_a, feature)
(function, uses, tool)
(attack, implements, feature)
(feature, can_expose, attack)
(attacker, can_exploit, vulnerability)
(attacker, can_exploit, feature)
(securityTeam, can_analyze, vulnerability)
(securityTeam, can_analyze, feature)
(securityTeam, can_exploit, app)
(securityTeam, uses, technique)
(securityTeam, implements, function)
(securityTeam, uses, system)
(securityTeam, uses, tool)
(user, uses, system)
(user, can_expose, vulnerability)

3.1.3 Drug Repurposing Knowledge Graph (DRKG)

[23] The Drug Repurposing Knowledge Graph (DRKG) is an extensive biological knowledge graph that connects genes, compounds, diseases, biological processes, side effects, and symptoms. It incorporates data from six well-established databases: DrugBank, Hetionet, GNBR, String, IntAct, and DGIdb, as well as information from recent publications, especially those related to Covid-19. The DRKG consists of 97,238 entities across 13 entity types and 5,874,261 triplets, each corresponding to one of 107 edge types. These edge types represent various interactions between 17 entity type pairs, with multiple interaction types possible between the same entity pair. Additionally, the DRKG includes several notebooks that demonstrate how to explore and analyze the data using statistical techniques or machine learning approaches, such as knowledge graph embedding.

DRKG Dataset

The dataset is divided into four main parts:

- `drkg.tsv`: A tab-separated values file that contains the original DRKG data formatted as (h, r, t) triplets.
- `embed`: A directory containing pretrained Knowledge Graph Embeddings, which were generated using the entire `drkg.tsv` dataset as the training set, along with pretrained Graph Neural Network (GNN)-based embeddings for molecules derived from their SMILES representations.
- `entity2src.tsv`: A file that maps each entity in DRKG to its original data source.
- `relation_glossary.tsv`: A file that provides a glossary of the relations within DRKG, along with associated information and sources when available.

Statistics of DRKG

The type-wise distribution of the entities in DRKG and their original data-source(s) is shown in following table.

Entity type	Drugbank	GNBR	Hetionet	STRING	IntAct	DGIdb	Bibliography	Total Entities
Anatomy	-	-	400	-	-	-	-	400
Atc	4,048	-	-	-	-	-	-	4,048
Biological Process	-	-	11,381	-	-	-	-	11,381
Cellular Component	-	-	1,391	-	-	-	-	1,391
Compound	9,708	11,961	1,538	-	153	6,348	6,250	24,313
Disease	1,182	4,746	257	-	-	-	33	5,103
Gene	4,973	27,111	19,145	18,316	16,321	2,551	3,181	39,220
Molecular Function	-	-	2,884	-	-	-	-	2,884
Pathway	-	-	1,822	-	-	-	-	1,822
Pharmacologic Class	-	-	345	-	-	-	-	345
Side Effect	-	-	5,701	-	-	-	-	5,701
Symptom	-	-	415	-	-	-	-	415
Tax	-	215	-	-	-	-	-	215
Total	19,911	44,033	45,279	18,316	16,474	8,899	9,464	97,238

The following table shows the number of triplets between different entity-type pairs in DRKG for DRKG and various datasources.

Entity-type pair	Drugbank	GNBR	Hetionet	STRING	IntAct	DGIdb	Bibliography	Total interactions
(Gene, Gene)	-	66,722	474,526	1,496,708	254,346	-	58,629	2,350,931
(Compound, Gene)	24,801	80,803	51,429	-	1,805	26,290	25,666	210,794
(Disease, Gene)	-	95,399	27,977	-	-	-	461	123,837
(Atc, Compound)	15,750	-	-	-	-	-	-	15,750
(Compound, Compound)	1,379,271	-	6,486	-	-	-	-	1,385,757
(Compound, Disease)	4,968	77,782	1,145	-	-	-	-	83,895
(Gene, Tax)	-	14,663	-	-	-	-	-	14,663
(Biological Process, Gene)	-	-	559,504	-	-	-	-	559,504
(Disease, Symptom)	-	-	3,357	-	-	-	-	3,357
(Anatomy, Disease)	-	-	3,602	-	-	-	-	3,602
(Disease, Disease)	-	-	543	-	-	-	-	543
(Anatomy, Gene)	-	-	726,495	-	-	-	-	726,495
(Gene, Molecular Function)	-	-	97,222	-	-	-	-	97,222
(Compound, Pharmacologic Class)	-	-	1,029	-	-	-	-	1,029
(Cellular Component, Gene)	-	-	73,566	-	-	-	-	73,566
(Gene, Pathway)	-	-	84,372	-	-	-	-	84,372
(Compound, Side Effect)	-	-	138,944	-	-	-	-	138,944
Total	1,424,790	335,369	2,250,197	1,496,708	256,151	26,290	84,756	5,874,261

3.2 Domain-specific Knowledge Graph Overview

[2]This section delves into the application of Knowledge Graphs (KGs) across diverse fields, underscoring their pivotal role in addressing sector-specific challenges. It categorizes the exploration into healthcare, education, ICT, science and engineering, finance, society and politics, and travel. Each domain's review encapsulates the utility of KGs, the methodologies for their construction, the sources of data employed, the integration of embedding techniques, the evaluation strategies adopted, and the limitations identified.

3.2.1 Healthcare Focus

The spotlight on healthcare has intensified, especially with the ongoing challenges posed by the COVID-19 pandemic. KGs have emerged as crucial tools for gleaning actionable insights from the vast and varied data within the healthcare sector. The literature reveals a segmentation of KG applications within healthcare into generic healthcare data mining, disease-specific KGs, and healthcare management systems, highlighting the use of KGs for data analytics, personal health profiling, and enhancing medical query systems.

3.2.2 Generic Healthcare

Data Mining: Utilizes KGs for encoding medical processes into directed graphs, aiding expert analysis.

Biomedical Ontologies: Integrates KGs with ontologies for addressing data incompleteness, enhancing semantic data analytics.

Personal Health KGs: Focuses on personalized health insights through KGs, indicating a gap in standard representation. Disease-oriented Applications

Disease and Symptoms KG: Constructs KGs from electronic medical records to map diseases and symptoms, leveraging public health databases.

Depression and Mental Health: Develops sub-graphs to model mental health disorders, pulling data from comprehensive medical sources. Healthcare Management

Diet and Nutrition: Proposes KGs to inform dietary choices, backed by healthcare websites and nutritional databases, aiming to address public health issues and chronic diseases through informed dietary management.

3.2.3 Education Insights

The incorporation of KGs within education aims to enrich learning systems and manage the abundance of educational content. The analysis covers KGs developed for enhancing teaching resources, managing educational frameworks, and bolstering educational technologies, reflecting on their contribution to curating educational content, scheduling, and learner assessment.

3.2.4 Teaching Resources

Educational KGs: Focuses on creating KGs for K-12 education, integrating curriculum standards and enhancing mathematical learning through crowdsourced content. Education Management

Course Allocation and Management: Highlights KG applications in optimizing course scheduling and management, underscoring the need for more robust evaluation and broader application.

3.2.5 Educational Technologies

Academic Networking: Details the use of KGs for academic publication management and embedding techniques in educational KGs to support learning and information retrieval.

3.2.6 ICT Applications

KGs serve a vital role in enhancing cybersecurity measures, streamlining software development processes, advancing telecommunication services, and integrating IoT devices. This segment reviews KGs' contributions to securing cyberspace, facilitating software testing, and promoting efficient network and IoT device management.

3.2.7 Cybersecurity and Software Development

Cybersecurity KGs: Documents efforts to construct KGs for identifying vulnerabilities and enhancing cybersecurity education.

Software KGs: Discusses the development of KGs to assist in software testing and development processes.

3.2.8 Telecommunications and IoT

Network Management KGs: Explores KGs' utility in telecommunication for incident management and network monitoring.

IoT Integration: Investigates KGs' role in bridging the communication gap among IoT devices, highlighting their potential in creating a cohesive IoT ecosystem.

Chemistry, Biology, and Geology

KGs find significant application in modeling complex scientific data across chemistry, biology, geology, and engineering disciplines. This section explores KGs designed to articulate chemical reactions, biological interactions, geological data, and engineering processes, demonstrating their capacity to organize and interpret scientific information.

Scientific KGs: Constructs KGs to represent chemical kinetics, biological processes, and geological data, enabling advanced query capabilities and data integration. Engineering Applications

Engineering KGs: Develops KGs for various engineering sectors, including manufacturing and power systems, to facilitate design, production, and operational efficiencies.

3.2.9 Finance Sector

In finance, KGs are leveraged for investment forecasting, fraud detection, and market analysis. This examination illustrates how KGs underpin financial models for predicting stock market trends, identifying fraudulent activities, and offering insights into financial events and transactions.

3.2.10 Investment and Fraud Detection

Market Forecasting KGs: Discusses the development of KGs for financial market analysis and stock prediction, employing KGs to navigate the complexities of financial data for investment strategies.

Fraud Detection KGs: Highlights the use of KGs in identifying fraudulent patterns and enhancing financial security measures.

3.2.11 Society and Politics

KGs' utility extends to understanding societal dynamics and political landscapes. The literature covers KGs' application in social media analysis, political systems modeling, and cultural heritage preservation.

3.3 Construction of the knowledge graph

[24]One method of storing data that comes from an information extraction process is to create a knowledge graph. A notion known as the "triple"—a collection of three elements—a subject, a predicate, and an object—that we can use to hold information about anything is used in many fundamental knowledge graph implementations.

We can define a graph as a set of nodes and edges.

Figure 3.1: The smallest knowledge graph

These two entities are connected by and edge that represents the relationship between the two entities. So the smallest knowledge graph that can be built is this which is also known as a **triple**.

We can examine the sentence below as an example

Micky mouse is distributed by walt disney.

After processing the the sentence we will be able to get a tiple like this

(Micky mouse , distributed by , walt disney)

so the example we have two unique entities (Micky mouse , Walt disney) and a relation distibuted by.

To create a knowledge graph, we only need two associated nodes in the network with the entities and vertices with the relations, and the result will be something like this.Manually

creating a knowledge graph is not scalable. Nobody will sift through thousands of pages to extract all of the entities and their relationships.

That is why machines are better suited to this activity, since they can go through hundreds or thousands of pages with ease. However, there is another challenge: machines do not grasp natural language. This is where natural language processing (NLP) comes into play.

To create a knowledge graph from text, our system must understand natural language. This can be accomplished by employing NLP techniques such as sentence segmentation, dependency parsing, parts of speech tagging, and entity recognition.

AS the key focus of this part is to create a simple Knowledge graph a Wikipidea data set was used to which contained **4318** rows of individual sentences. These sentecs appear are narrative in nature and extracted from stories , movie scrips or descriptive texts. The sentences descrive event of actions of various kind. The dataset was then taken through the following steps to get the final Knowledge graph.

3.3.1 Sentence Segmentation

The initial step in creating a knowledge graph is dividing the text document or article into sentences. Then, we will shortlisted phrases that have exactly one subject and one object.

From the sentence below a observation can be found leveraging the technique of the NLP in the table 3.2 .

later, a woman's scream is heard in the distance.

Table 3.7: Tokens and Their Parts of Speech

Token	Part of Speech	Full Form
later	advmod	Adverbial modifier
,	punct	Punctuation
a	det	Determiner
woman	poss	Possessive modifier
's	case	Case marking
scream	nsubjpass	Passive nominal subject
is	auxpass	Passive auxiliary
heard	ROOT	Root
in	prep	Preposition
the	det	Determiner
distance	pobj	Object of preposition
.	punct	Punctuation

3.3.2 Entity Extraction

The most significant components of a knowledge graph were the nodes and the edges that connected them.

These nodes represented the entities that appeared in Wikipedia sentences. Edges were the relationships that connected these entities. We extracted these pieces unsupervised, using the grammar of the statements.

The two entities that can be found from the following line are

Actual Sentence : christian is then paralyzed by an elder Entities: ['christian', 'then elder']

Just like this entity pair all the entity pair for the each of the sentence was prepared for making the KG. A portion of that can be represented below :

Table 3.8: Entity Pairs in Sentences

Entity 1	Entity 2
vivienne graham	former employee dr
monster which	several monarch dr
madison	mythological texts
joe morton	older .
godzilla	king themselves
when dougherty	when reaction
we	godzilla
only mandate	monarch
ten writers	treatment
script	year

3.3.3 Extraction from Relation/Predicate

Our research is that in a sentence, the basis serves as a primary verb. so using this predicate we are creating the relation to generate the triple from a sentence. There were several relations which were extracted to see the frequency of the relations in the dataset. The top 5 relation that were found are 3.9

Table 3.9: Relation and their Frequency

Relation	**Frequency**
is	365
was	299
Released on	88
Are	78
include	72

3.3.4 Build a Knowledge Graph

We will finally create a knowledge graph from the extracted entities (subject-object pairs) and the predicates (relation between entities).

Using the network library to create a network from this data frame. The nodes will represent the entities and the edges or connections between the nodes will represent the relations between the nodes.

It is going to be a directed graph. In other words, the relation between any connected node pair is not two-way, it is only from one node to another. For example, “John eats pasta”. Now initially with all the relations the knowledge graph will look something like this for this dataset Which can further be simplified based on the relation to how things are done. We can find "written by" relation from the from the KG easily .

From figure 3.3 this enables us to capture all the possible complex relationships between the entities and visualize them to make a decision.

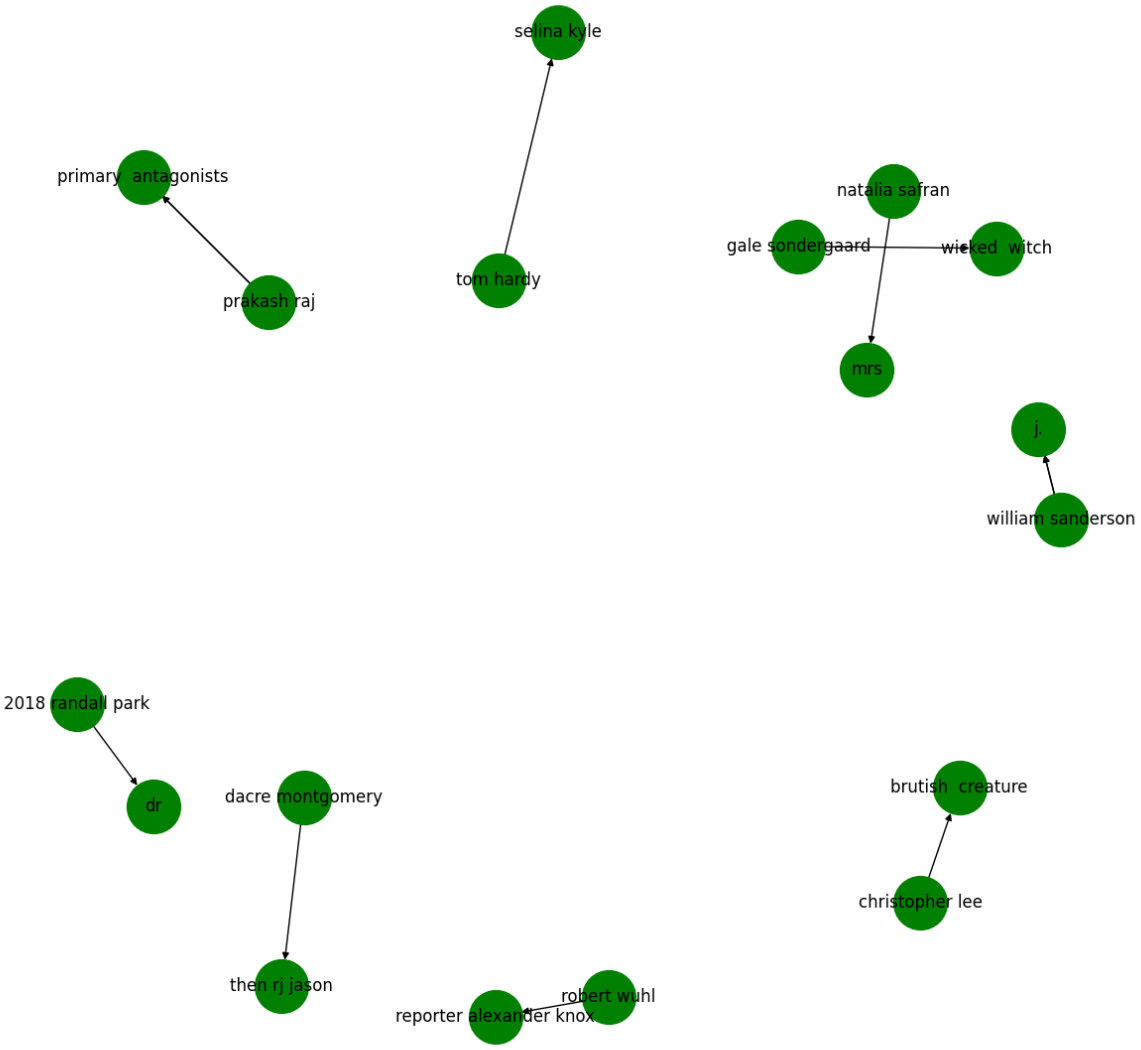

Figure 3.2: Relation "cast as" exist between the two entities

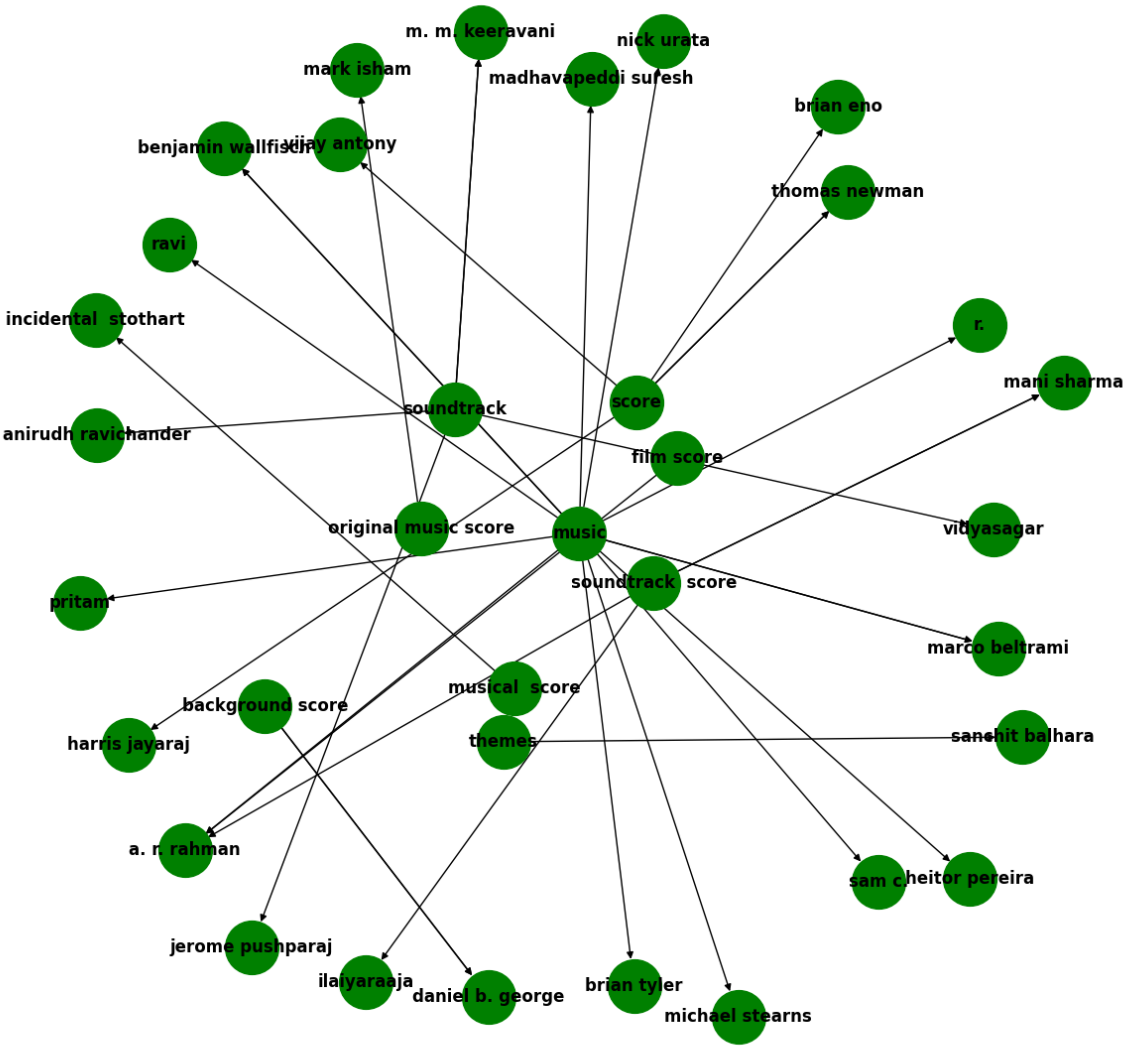

Figure 3.3: Relation "composed by" exist between the two entities

Chapter 4

Studying Application of Knowledge Graphs

4.1 AISecKG: Knowledge Graph Dataset for Cybersecurity Education

4.1.1 Introduction and Related Work

Cybersecurity education is inherently complex, requiring learners to master intricate attack-defense strategies, leverage various tools, and develop critical thinking skills to secure systems effectively. Artificial Intelligence (AI) integration into education has shown promise in enhancing cognitive engagement and fostering active learning. Knowledge graphs (KGs) serve as effective tools in such AI-enabled systems, offering visual representations of domain-specific knowledge, enabling reasoning, and facilitating interpretation.

Despite the growing demand for cybersecurity professionals, there exists a significant gap in publicly available datasets tailored for cybersecurity education. Current resources are often fragmented and unstructured, scattered across course materials, wiki pages, and writeups from capture-the-flag (CTF) challenges. Previous efforts, such as the Unified Cybersecurity Ontology (UCO) for vulnerability management and MALOnt for malware threat intelligence, primarily focus on advanced applications like intrusion detection and vulnerability analysis, leaving educational contexts for novice learners underexplored.

AISecKG addresses this gap by presenting a threefold contribution:

- A domain ontology for cybersecurity education.
- An annotated triple dataset to construct knowledge graphs.

- Downstream applications demonstrating the utility of this dataset in building educational tools.

These contributions aim to simplify complex cybersecurity concepts, promote active learning, and support novice learners in navigating the cybersecurity ecosystem.

4.1.2 Data Source and Dataset Structure

The dataset development process began with collecting instructional materials from advanced cybersecurity lab manuals used in graduate courses. These materials, spanning topics such as intrusion detection systems with NMap and Snort, honeypot deployment in Metasploit, and system monitoring using Syslog, consist of approximately 26,886 words across 100 pages. Preprocessing excluded code snippets, focusing on textual instructions and concepts.

The dataset is structured as rows of sentences, tags, and document identifiers (*DocNo*). Table 4.1 shows a sample structure.

Table 4.1: Sample rows from the AiSecKG dataset.

DocNo	Sentence	Tag	Description
D1	Attacks	B-ATTACK	Represents an attack entity.
D1	can	O	Non-entity word.
D1	damage	O	word that is not an entity.
D1	public	B-SYSTEM	The start of a system entity
D1	domain	I-SYSTEM	Continuation of a system entity.

4.1.3 Ontology

AISecKG ontology was developed using a bottom-up approach, leveraging domain knowledge and insights from the lab manuals. The ontology is categorized into three primary pillars: concepts, applications, and roles, encompassing 12 distinct entity types such as features, functions, attacks, vulnerabilities, and tools. These entities are interconnected through nine relation types, such as *has_a*, *uses*, *can_analyze*, and *can_harm*, to represent real-world interactions in the cybersecurity domain.

For instance, *users* interact with *applications* and *systems*, while *attackers* exploit *vulnerabilities* to compromise *systems* and *data*. Similarly, *security teams* employ *tools* and *techniques* to detect and mitigate *attacks*.

4.1.4 Dataset Annotation and NLP Applications

The dataset was annotated using the BIO tagging scheme, marking 964 unique entities across 593 sentences. Semi-automated techniques with NLP tools like SpaCy facilitated the annotation process, significantly reducing manual effort. Manual validation ensured the quality of the final dataset, which includes 730 validated triples.

NLP models were further trained for named-entity recognition (NER) using the annotated dataset. After refinement, transformer-based models such as BERT and RoBERTa achieved above 80% entity recognition accuracy. Metrics like F1-score, precision, and recall showed how well the models identified various cybersecurity entities.

4.1.5 Knowledge Graph Construction and Representation

AISecKG supports the development of knowledge graphs, providing visual representations of cybersecurity concepts and their relationships. The dataset is structured as a graph $G = (V, E)$, where V denotes the nodes and E denotes the edges. The graph encompasses:

- **Nodes (V)**: Representing words, tags, and document IDs.
- **Edges (E)**: Establishing relationships such as sequence, labeling, and contextual dependencies.

Subgraphs focused on specific entities, such as the NMap tool, provide students with an intuitive understanding of complex relationships, such as how specific tools interact with vulnerabilities or techniques. Graph databases like Neo4j and query languages such as SPARQL or GraphQL can be employed for dynamic exploration.

4.1.6 Applications and Use Cases

The embeddings derived from the AISecKG dataset enable several downstream tasks:

- **Entity Classification**: Classify nodes into predefined categories based on their embeddings.
- **Relationship Prediction**: Predict missing edges in the graph.
- **Clustering**: Cluster entities with similar semantic meanings.
- **Visualization**: To visualize high-dimensional embeddings, apply dimensionality reduction techniques like PCA or t-SNE.

4.2 DRKG: Knowledge Graph Dataset for Drug Repurposing Research

4.2.1 Introduction

Drug repurposing, the process of identifying new uses for existing drugs, offers several advantages over de novo drug design, such as shorter development timelines and lower risk. [25] [26] It is estimated that drug repurposing can shorten the typical 10- to 17-year drug development cycle to just 3 to 12 years. With the advancement of machine learning and embedding techniques, numerous studies have assessed their effectiveness in drug repurposing. Donner et al. [27] utilized deep neural networks to analyze gene expression embedding profiles for predicting pharmacological similarities in drug repurposing. Mei and Zhang [28] introduced a multi-label learning framework that leverages L2 regularized logistic regression to uncover new applications for old drugs and identify new drugs for known target genes. Additionally, Xuan et al. [29] proposed a non-negative matrix factorization-based approach to predict new indications for existing drugs, utilizing diverse information such as disease similarities, drug-disease associations, and drug similarities. While these studies have made significant progress, certain limitations remain, and the following research questions arise:

1.How can drug repurposing be effectively applied to predict novel interactions between genes and chemical compounds, particularly for Covid-19 treatments?

2.What methods were used to ensure the high quality of the DRKG structure and the embeddings learned for drug repurposing tasks?

3.Can the TransE model effectively identify relationships and embeddings in drug-disease interaction networks?

In order to effectively respond to these research questions,we constructed a comprehensive Drug Repurposing knowledge graph(DRKG).This graph integrates data from diverse sources such as DrugBank, String, GNBR, and recent publications on Covid-19. By leveraging graph machine learning models, they were able to predict novel interactions between genes and chemical compounds.Secondly,To ensure the DRKG's quality, we have performed meticulous analyses to validate both the graph structure and the embeddings. we filtered out noisy links and nodes to enhance the robustness of the DRKG. Furthermore, we conducted evaluations to confirm that the embedding models accurately represented biological entities and their relationships.Finally,we applied the TransE framework to datasets related to drug repurposing and discovery. This framework was used to effectively capture rela-

tionships within the drug-disease interaction networks. TransE demonstrated its capability in generating embeddings that could accurately identify and represent meaningful relationships.

4.2.2 Data Source and Data Structure

Six current databases—DrugBank, Hetionet, GNBR, String, IntAct, and DGIdb—as well as data gathered from recent publications, especially those pertaining to Covid-19, are included in the aforementioned DRKG.The Knowledge Graph (KG) contains 9708 compounds sourced from DrugBank, categorized into 8968 small-molecule drugs (92.4%) and 736 biotech drugs (7.6%). Among the small-molecule drugs, SMILES data has been successfully extracted for 8807 compounds (98.2%) from DrugBank and other sources, while 161 lack this information due to missing structural details or multi-ingredient compositions.The dataset is designed to help with exploring and using the knowledge graph for machine learning tasks.We initially loaded the dataset from a file drkg.tsv into a pandas DataFrame, and the total number of triplets is counted as 5,874,261.We converted the DRKG data into a homogeneous format to ensure standardization, reduce computational complexity, and facilitate evaluation and interpretation

4.2.2.1 Constructing the DRKG

In order to guarantee a thorough and superior depiction of biological entities and their relationships, the Drug Repurposing Knowledge Graph (DRKG) was constructed by methodically integrating data from several different sources. The steps listed below provide a summary of the building process:

Data Integration and Normalization: Triplets taken from six main databases—DrugBank, GNBR, Hetionet, STRING, IntAct, and DGIdb—as well as bibliographic information were used to construct the DRKG. The relationships and interactions between two items, like as genes, chemicals, or illnesses, are represented by each triplet. To guarantee consistency across databases, the extracted data were standardized to a single ID space:

- **Compounds**: Mapped to DrugBank or ChEMBL IDs.
- **Genes**: Mapped to Entrez IDs.
- **Diseases**: Mapped to MeSH IDs.

This mapping allowed entities from different sources to be seamlessly connected within the DRKG.

Filtering and Quality Assurance: The initial dataset underwent filtering to remove noisy or unreliable links. This process included:

- Discarding entities involved in fewer than two triplets.
- Removing relationships with insufficient evidence based on confidence scores or co-occurrence thresholds.
- Excluding relation types with fewer than 50 occurrences to ensure sufficient data for training embedding models.

Enrichment with COVID-19-Specific Data: To address the urgent need for COVID-19 treatment research, the DRKG was enriched with interactions from recent publications. This included:

- Data on SARS-CoV-2 proteins and their interactions with human host proteins and chemical compounds.
- Information linking COVID-19-related diseases to genes and compounds involved in related pathways.

Standardization and Format Conversion: In order to facilitate computation and comprehension in subsequent tasks, the extracted triplets were transformed into a uniform, standardized format. By simplifying heterogeneous interactions, this format made it easier to apply machine learning models.

The final DRKG contains:

- **97,055 entities** categorized into 13 types, including compounds, genes, diseases, pathways, and side effects.
- **5,869,294 edges** spanning 107 types of relationships, such as “treats,” “binds,” and “inhibits.”

Graph quality was validated using graph embedding models such as TransE. This included:

- Ensuring that embeddings clustered biologically similar entities together.
- Verifying that predicted interactions aligned with known biological mechanisms.

The resulting DRKG serves as a powerful tool for machine learning applications, enabling researchers to predict novel drug-disease and drug-gene interactions efficiently.

4.2.3 TransE Model for Knowledge Graph Link Prediction and Embedding

The TransE model is typically designed for homogeneous graphs, where all nodes (entities) are of the same type and relations are treated uniformly. In the case of a heterogeneous graph, which includes multiple types of nodes and edges, issues may arise because the DGL (Deep Graph Library) sampler may not be compatible with such graphs.

4.2.3.1 Link Prediction

Knowledge Graphs (KGs), which utilize graph-based knowledge representation, have significantly advanced various AI tasks. As a result, KGs are now extensively applied across multiple sectors, including science, industry, and business, for effective data management. However, a key challenge with KGs is that despite containing millions of triples, fully capturing real-world knowledge is nearly impossible, even within specific domains. Consequently, KGs often remain incomplete. To address this challenge, we introduce TransE, an embedding model designed specifically for link prediction in KGs. The main objective of link prediction in the TransE model is to predict missing edges in a knowledge graph. For each triplet (head, relation, tail), the model learns embeddings for the head, relation, and tail entities. The core concept of TransE is that for a valid triplet (h, r, t), the vector representation of the head entity plus the relation vector should be close to the tail entity vector in the embedding space. The link prediction task is carried out by:

- Calculating the scores for both positive and negative triplets using the forward() method
- Assessing the prediction quality with the loss() function
- Adjusting the model's parameters during training to minimize the loss and enhance link prediction accuracy.

4.2.3.2 Graph Embedding

Graph embedding involves learning vector representations for entities and relations that capture the structural properties of the knowledge graph. These embeddings use geometric modifications in the vector space to depict the relationships between entities. Finding high-quality embeddings for entities and relations in a domain-specific knowledge graph (DRKG) is the major goal. The following tasks can benefit from these embeddings' ability to capture significant patterns and semantic insights:

- Drug discovery

- Drug repurposing
- Predictions and recommendations

The model should be able to generalize well across a variety of tasks, including classification within the knowledge graph, link prediction, and relation prediction, thanks to the learnt embeddings. As the model picks up vector representations for the entities and relations in the knowledge graph throughout the forward pass, embedding takes place. A margin-based ranking loss function is used to improve these embeddings during training.

4.2.4 Embedding Analysis

Embeddings are vector representations of entities or relationships in a multi-dimensional space. These vectors are learned during training, often in knowledge graph embeddings, and encode semantic and structural information. The analysis of embeddings provides insight into how well the model has learned the underlying structure of the data. For instance:

- In the vector space, are related relationships clustered together?
- Does the vector space contain near clusters of related relationships?

The embeddings capture patterns like similarity, relatedness, and even functional relationships in the data.By analyzing embeddings, we can identify problems such as poor clustering, biases, or incorrect relationships, which can guide us in improving the model or data.

4.2.4.1 Methodologies to Analyze Embeddings

1.Projections to Lower-Dimensional Spaces

- While embeddings are often constituted with enormous dimensionality, dimensionality reduction techniques such as t-SNE or PCA are employed to compress their sprawling and intricately woven structure into simpler subspaces better suited for human comprehension.
- Mapping the embeddings onto fewer dimensions unfolds their intricate topology, revealing clusters and connections between relations that flourish indistinctly within their original astronomical dimensionality.

2.Cosine Distance for Similarity Analysis

Cosine similarity is used to measure the similarity between two embeddings. The cosine similarity between two vectors **a** and **b** is defined as:

$$\text{cosine\_similarity}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2}$$

This measure ranges from -1 (completely dissimilar) to 1 (completely similar), with 0 indicating orthogonal vectors.

- To gauge the angular similarities and dissimilarities between the directional semantics of word pairs, cosine distance evaluates their orientation in the multidimensional embedding space.
- Through computing this angle-based measure of correspondence, it becomes possible to quantify how closely two linguistic concepts are related - or unrelated - in the embedded semantic realm.
- A small cosine distance (or high cosine similarity) implies the relations are semantically or functionally similar.

3.Frobenius Distance for Similarity Analysis
The Frobenius distance is used to measure the difference between two matrices (or sets of embeddings). It is defined as the Frobenius norm of the difference between two matrices A and B:

$$\text{frobenius\_distance}(A, B) = \|A - B\|_F = \sqrt{\sum_{i,j} |A_{ij} - B_{ij}|^2}$$

- The Frobenius distance is a suitable measure of matrix norm used to compare matrices.
- This is especially effective when embeddings are shown as higher-dimension matrices or gentle distinctions must be discerned.

4.2.5 Application and Use Cases

t-SNE and embedding exploration help in easy identification of trends, anomalies within large data set and used in wider applications like knowledge graph analysis, clustering, and biomedical analysis. Because of their ability to discover latent patterns and relationships no motor behind them, they become powerful tools for applications, including drug discovery, graph-based machine learning, and also exploratory data analysis.

Chapter 5

Result & Discussion

5.1 Analysis with dataset - AiSecKG

5.1.1 Result

Table 5.1 provides a detailed performance comparison of five graph embedding models—RotatE, Complex, TransE, DistMult, and RESCAL—evaluated on the AiSecKG dataset. The metrics used for evaluation include Mean Rank, Precision, Recall, F1 Score, Accuracy, and AUC.

From the results, it is evident that the Complex model consistently outperforms the others across most metrics, achieving the highest Precision (0.99), Recall (0.93), F1 Score (0.96), and AUC (0.99). On the other hand, RotatE and DistMult also show competitive performance, with DistMult achieving a high Precision (0.99) and F1 Score (0.92). In contrast, the TransE model performs moderately well, while RESCAL shows the lowest performance across multiple metrics, particularly in Mean Rank (59.79) and AUC (0.48). These results highlight the varying strengths and weaknesses of the models when applied to the AiSecKG dataset.

Table 5.1: Evaluation of Graph Embedding Models' Performance on the AiSecKG Dataset

Metric	RotatE	Complex	TransE	DistMult	RESCAL
Mean Rank	2.29	1.41	12.54	1.60	59.79
Precision	0.16	0.99	0.48	0.99	0.50
Recall	1.00	0.93	0.50	0.86	0.50
F1 Score	0.28	0.96	0.49	0.92	0.50
Accuracy	0.16	0.96	0.96	0.93	0.91
AUC	0.04	0.99	0.56	0.99	0.48

5.1.2 Discussion

The performance of various graph embedding models was evaluated using the AiSecKg dataset. From the results presented in Table 5.1, the following observations can be made:

1. **RotatE**: This model demonstrates a strong performance in terms of Recall (1.00), indicating its ability to retrieve all relevant results. However, it struggles significantly with Precision (0.16), suggesting a high rate of false positives. As a result, the F1 Score is only 0.28, and the model has the lowest AUC (0.04) and Accuracy (0.16) among all models. These findings suggest that while RotatE is effective at identifying relevant items, its utility is limited due to poor filtering of irrelevant data.

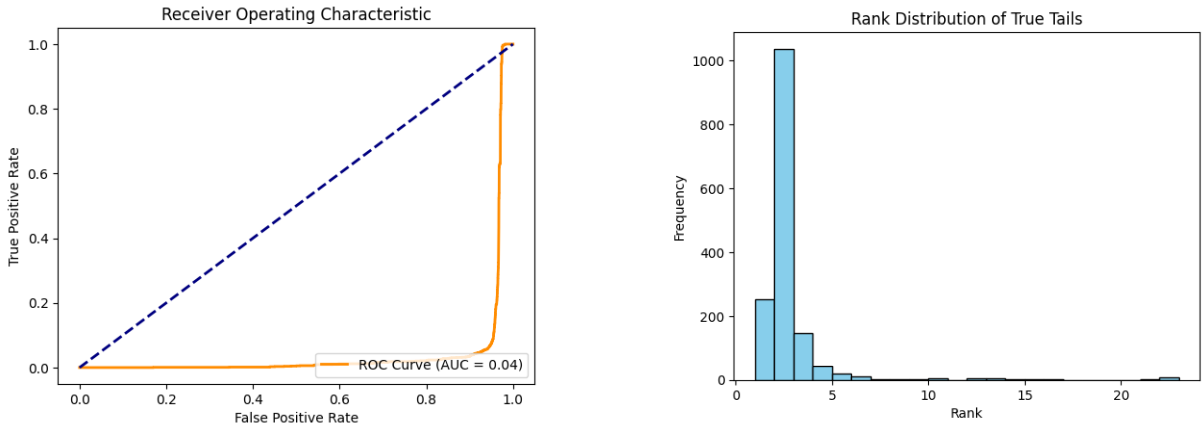

Figure 5.1: ROC and Rank Distribution of True tails for RotateE Model

2. **Complex**: The Complex model stands out for its well-rounded performance, achieving high Precision (0.99), Recall (0.93), and F1 Score (0.96). These metrics demonstrate that Complex excels at both retrieving relevant data and minimizing false positives. With an Accuracy of 0.96 and an AUC of 0.99, it comes out as one of the best-performing models, showing strong reliability and robustness across all evaluation metrics.

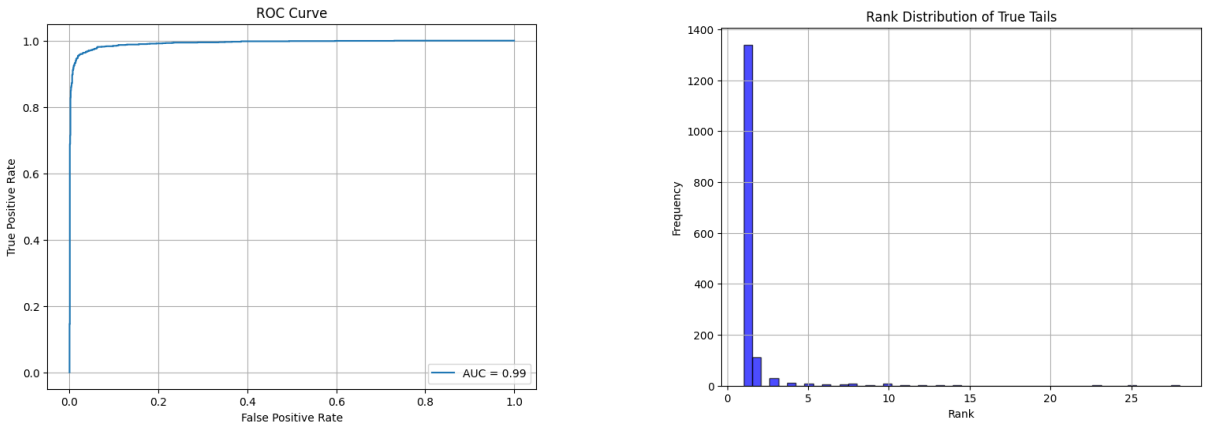

Figure 5.2: ROC and Rank Distribution of True Tails for Complex Model

3. **TransE**: While TransE achieves a decent Accuracy (0.96), its other metrics are moderate. Precision (0.48), Recall (0.50), and F1 Score (0.49) highlight its struggles with balancing true positives and false positives. The relatively low AUC (0.56) further underscores its limited effectiveness in distinguishing between relevant and irrelevant results. TransE might be better suited for applications requiring moderate recall and precision without high computational demands.

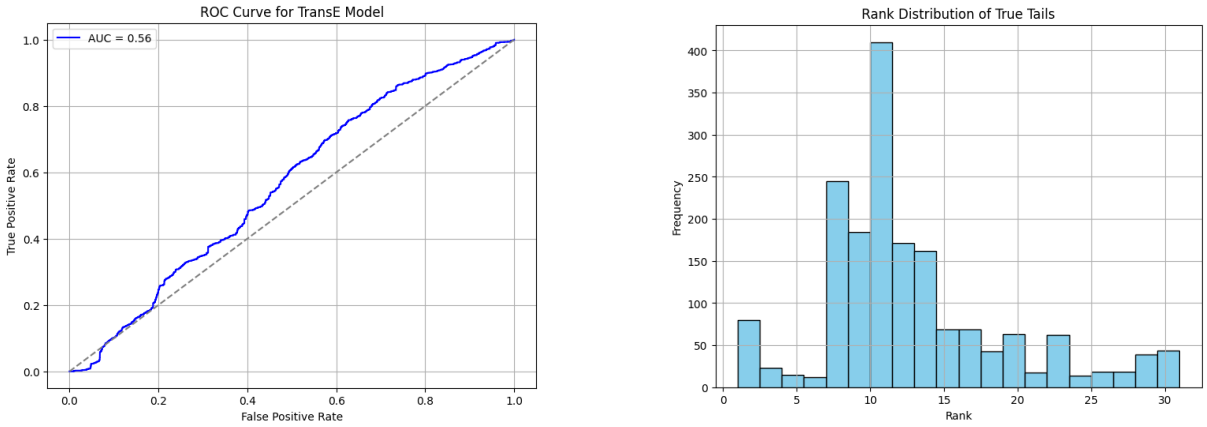

Figure 5.3: ROC and Rank Distribution of True Tails for TransE Model

4. **DistMult**: Similar to Complex, DistMult delivers excellent results, with a Precision of 0.99 and an F1 Score of 0.92. Although its Recall (0.86) is slightly lower than Complex, it maintains strong Accuracy (0.93) and AUC (0.99). This makes DistMult a reliable choice, particularly for tasks prioritizing high precision and robust model performance.

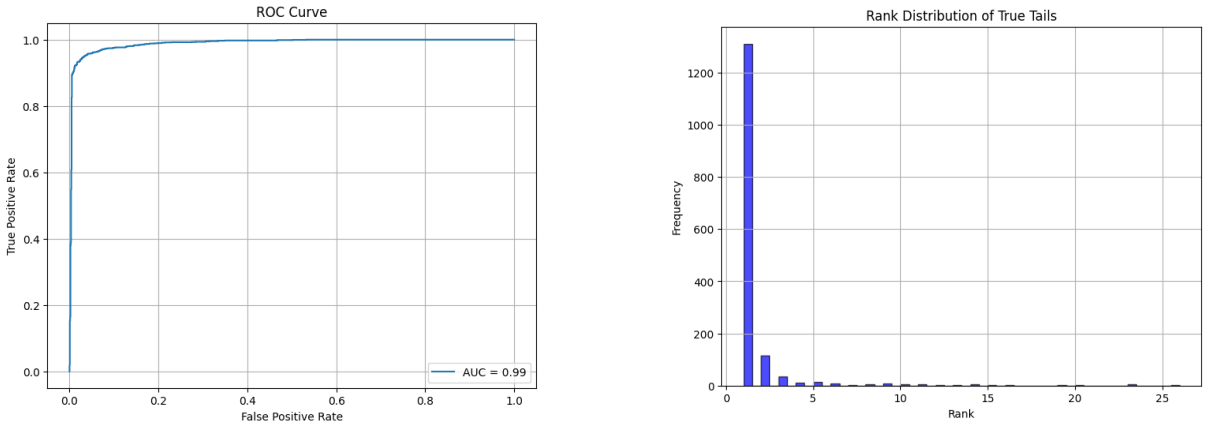

Figure 5.4: ROC and Rank Distribution of True Tails for DistMult Model

5. **RESCAL**: RESCAL shows moderate performance across all metrics, with Precision, Recall, and F1 Score at 0.50. Its Accuracy (0.91) and AUC (0.48) indicate that it performs better than RotatE but falls short of the other models. RESCAL may benefit from further optimization to improve its balance of precision and recall.

In summary, Complex and DistMult are the best-performing models on the AiSecKg dataset, with Complex having a slight advantage due to its balanced metrics. TransE and RESCAL

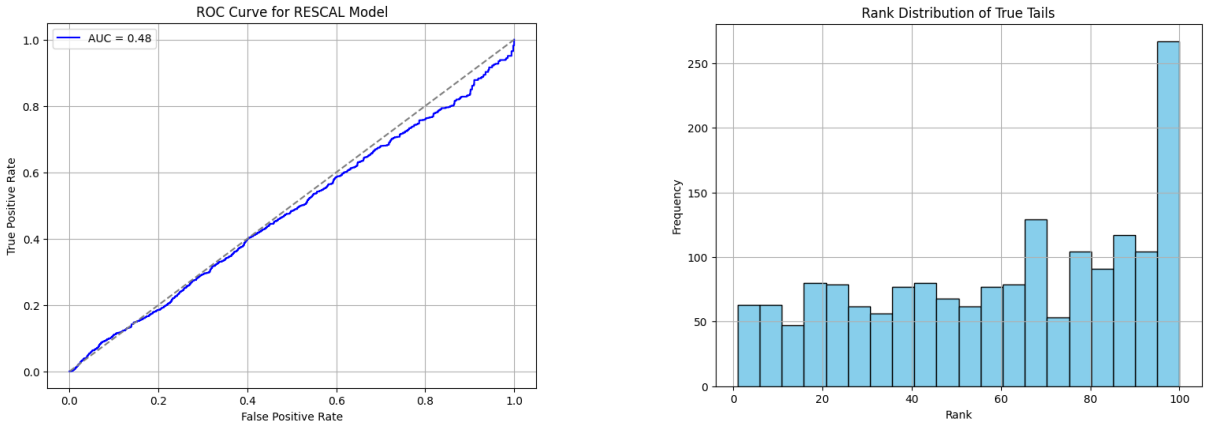

Figure 5.5: ROC and Rank Distribution of True Tails for RESCAL Model

offer moderate capabilities but are less competitive, while RotatE, despite its perfect Recall, underperforms due to poor precision and overall accuracy. Future research could explore hybrid approaches that combine the strengths of multiple models or refine underperforming models like RotatE and RESCAL to improve their precision and overall effectiveness.

5.2 Analysis with dataset - DRKG

5.2.1 Result

Projections to Lower-Dimensional Spaces

For the sake of examining how relation embeddings are positioned in space, t- SNE was used to visualize the high dimensional relation embeddings onto a 2D plane. The visualization produced is found in figure 5.6. Each point on the graph denotes relation embedding and each graph point has a specific color according to the dataset.

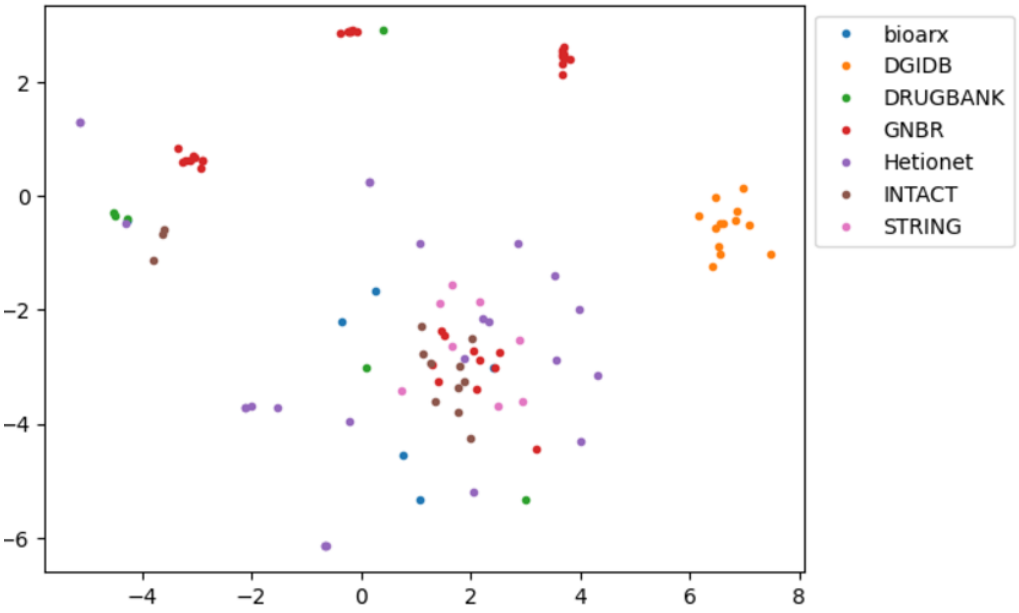

Figure 5.6: t-SNE visualization of relation embeddings in 2D space

Pair-wise Cosine Similarity Analysis

In order to quantify the semantic similarity between the relation embeddings, we compared each pair of relations using a cosine similarity metric. We present in figure 5.7 the distribution of the resulting cosine similarity scores while table 5.2 summarizes the top 10 most similar relation pairs.

Frobenius Distance for Similarity Analysis

In addition to cosine similarity, we also determined the pairwise similarity on the basis of the Frobenius distance.The distribution of the Frobenius distances is presented in the figure 5.8.The pair-wise distances between the embeddings were computed, and the top 10 most similar relation pairs are summarized in Table 5.3.

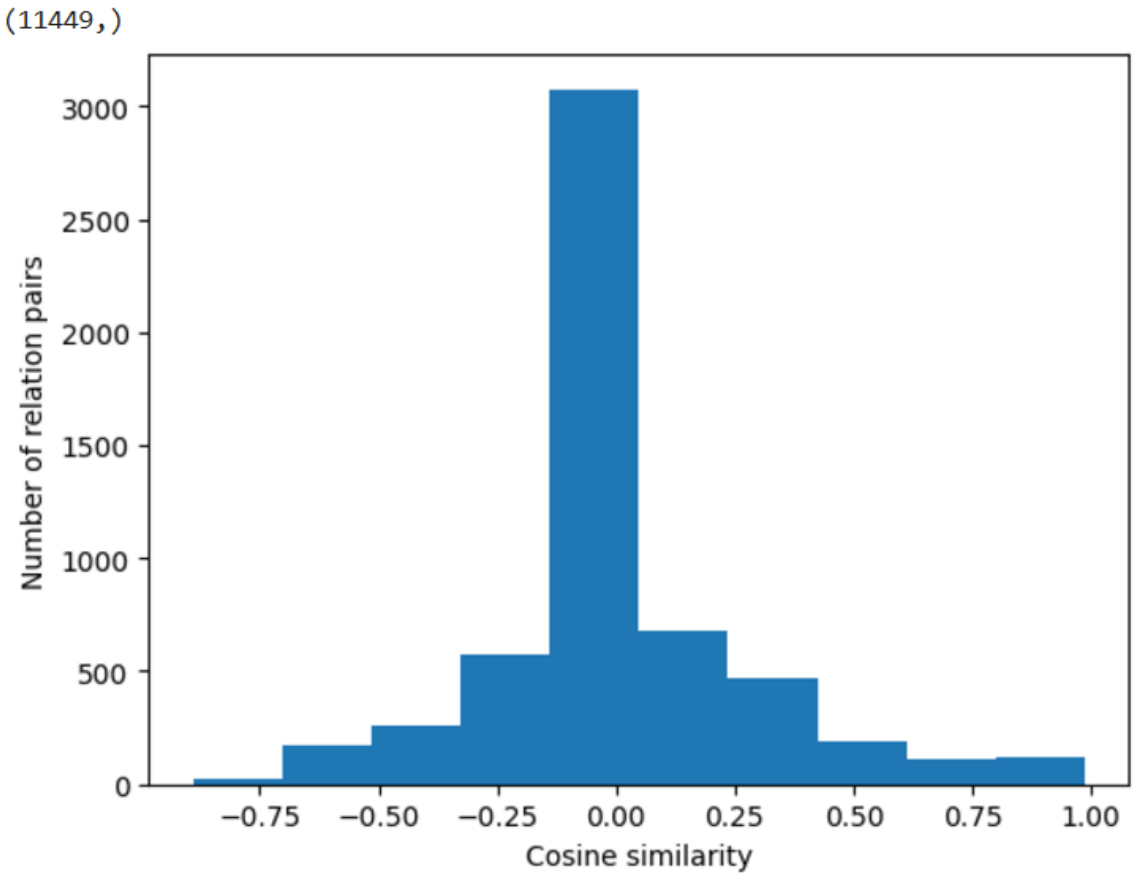

Figure 5.7: Histogram of pair-wise cosine similarity scores.

Table 5.2: Top 10 Most Similar Relation Pairs Based on Cosine Similarity

Rank	Relation Pair 1	Relation Pair 2	Cosine Similarity
1	GNBR::E::Compound:Gene	GNBR::K::Compound:Gene	0.9859
2	GNBR::E::Compound:Gene	GNBR::E+::Compound:Gene	0.9829
3	GNBR::N::Compound:Gene	GNBR::E-::Compound:Gene	0.9698
4	GNBR::E::Compound:Gene	GNBR::E-::Compound:Gene	0.9653
5	GNBR::K::Compound:Gene	GNBR::E+::Compound:Gene	0.9565
6	GNBR::E+::Compound:Gene	GNBR::E-::Compound:Gene	0.9502
7	GNBR::L::Gene:Disease	GNBR::G::Gene:Disease	0.9419
8	GNBR::K::Compound:Gene	GNBR::E-::Compound:Gene	0.9407
9	GNBR::J::Gene:Disease	GNBR::Md::Gene:Disease	0.9319
10	GNBR::J::Gene:Disease	GNBR::Te::Gene:Disease	0.9318

5.2.2 Discussion

From the visualization of **projections to lower dimensional spaces**,several observations can be made-

1.Distinct Cluster: Relations from the DGIDB dataset (orange) form a tight cluster on the right side of the plot, indicating high similarity among embeddings in this dataset.Relations

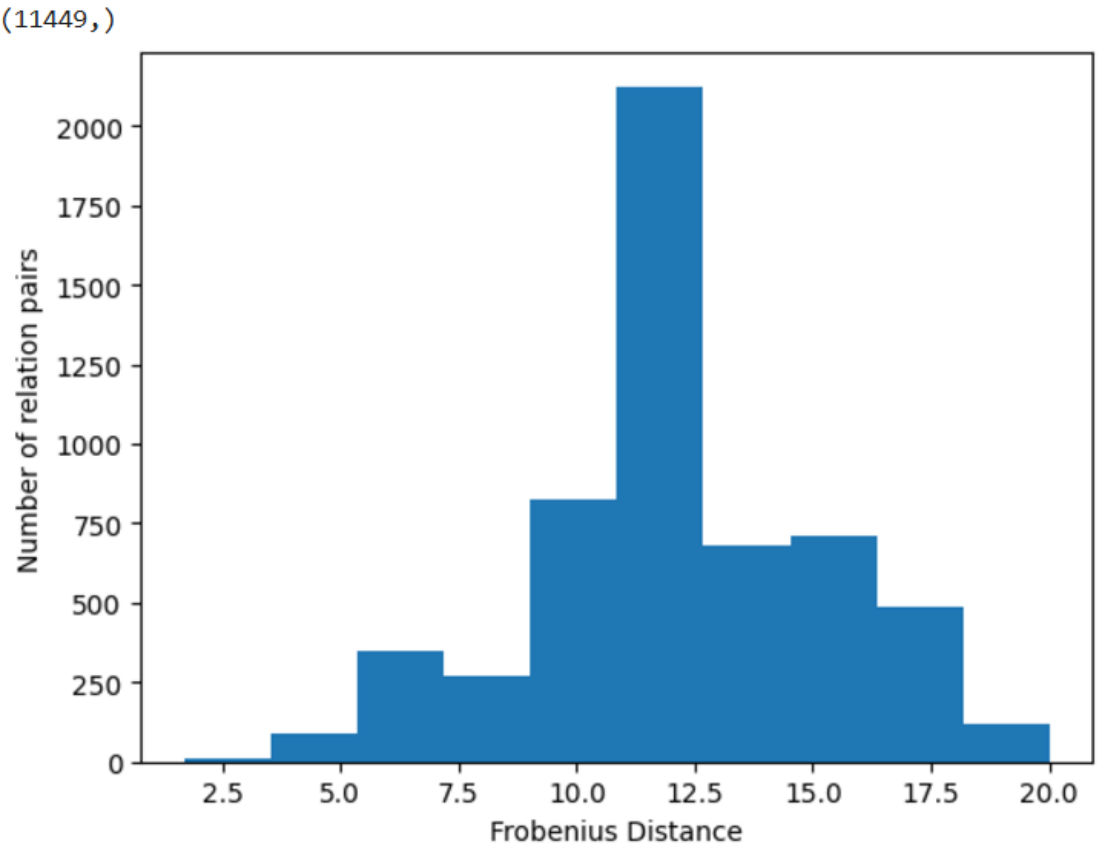

Figure 5.8: Histogram of pair-wise Frobenius similarity scores.

Table 5.3: Top 10 Most Similar Relation Pairs Based on Frobenius Distance

Rank	Relation Pair 1	Relation Pair 2	Frobenius Distance
1	GNBR::E::Compound:Gene	GNBR::K::Compound:Gene	1.6985
2	GNBR::E::Compound:Gene	GNBR::E+::Compound:Gene	1.8601
3	GNBR::N::Compound:Gene	GNBR::E-::Compound:Gene	2.3698
4	GNBR::E::Compound:Gene	GNBR::E-::Compound:Gene	2.6062
5	GNBR::K::Compound:Gene	GNBR::E+::Compound:Gene	2.9946
6	GNBR::E+::Compound:Gene	GNBR::E-::Compound:Gene	3.1560
7	GNBR::L::Gene:Disease	GNBR::G::Gene:Disease	3.4119
8	GNBR::K::Compound:Gene	GNBR::E-::Compound:Gene	3.4543
9	GNBR::J::Gene:Disease	GNBR::Md::Gene:Disease	3.6072
10	GNBR::J::Gene:Disease	GNBR::Te::Gene:Disease	3.6244

from GNBR (red) form small, well-separated clusters, suggesting internal cohesion within GNBR relations but dissimilarity from other datasets.

2.Overlapping and Centralized Distributions: Relations from Hetionet, INTACT, and STRING are more spread out but also have a large portion of their points overlapping closer to the center, which reflects a shared semantic space or similarity among these datasets.

3.Outliers:A few outliers can be seen, especially for the DRUGBANK and Hetionet, on the

far left. These could represent unique relations that are very different from the rest of the embeddings.

From the evaluation of embedding similarity using **cosine distance**,we can observe-

1.Score Distribution: The histogram shows that, while most relation pairs have low similarity, a subset have significant similarity (cosine similarity > 0.9).

2.Top Similar Pairs:GNBR::E::Compound:Gene and GNBR::K:: show the highest similarity (0.9859).Compound:Gene indicates that these two relations have very similar embeddings. Notably, the majority of the top similar pairs are from the GNBR dataset, specifically the Compound-Gene and Gene-Disease relations. This demonstrates the internal semantic consistency of GNBR relations.

3.Implications: High similarity scores indicate redundancy or shared semantics among different relation types. Analyzing these pairs can help identify closely related embeddings, which may assist in understanding dataset structure and relation semantics.

we calculate the pair-wise embedding similarity using **Frobenius Distance** and can observe-

1.Score Distribution: The histogram of the Frobenius distances tells us that most relation pairs have their distances concentrated within the range of 10-12, peaking around 12.5. In this case, the distribution makes it clear that while most of the embeddings are only somewhat similar, only a minority subset attains really small distances (high similarity).

2.Top Similarity Pairs: The distances must-have shown the variability of relations as modeled in the data. Expectedly, the most closely related pairs such as GNBR::E::Compound:Gene and GNBR::K::Compound:Gene (distance=1.6985) had the smallest Frobenius distance. Such a finding was in line with the view that relations characterized by comparable meaning or roles tended to be located closely in the embedding space. Therefore, the closeness between the two embeddings demonstrated a very close similarity, reflecting some aspects of their common semantic or functional characteristics.

3.Implications: The findings of the research establish that the embeddings successfully capture the semantic relationships that exist within the dataset. Smaller Frobenius distances then correspond to a higher semantic similarity and would thus indicate that the embedding model has learned meaningful representations for the relation pairs.

Chapter 6

Conclusion & Future Works

6.1 Conclusion

This research highlights the transformative potential of Knowledge Graphs (KGs) as a powerful tool for semantic data representation, integration, and analysis. By focusing on two domain-specific datasets—AISecKG for cybersecurity education and DRKG for drug repurposing—the study demonstrates how KGs can enhance domain-specific knowledge discovery, decision-making, and application development. The research explored various embedding models such as TransE, RotatE, ComplEx, DistMult, and RESCAL, comparing their performance using metrics like Mean Rank and HITS@k. Advanced visualization techniques, including t-SNE and similarity measures, provided insights into the latent structure and relationships within datasets.

The findings underscore the importance of domain-specific ontology design and KG construction methodologies in tackling complex, real-world problems. AISecKG proved effective in simplifying complex cybersecurity concepts, while DRKG offered valuable insights for biomedical research, particularly in drug repurposing. Despite these successes, challenges such as scalability, data heterogeneity, and the need for enhanced usability persist. Overall, the research contributes to the growing body of knowledge on KG applications and provides a foundation for further exploration and innovation.

6.2 Future Works

The study opens avenues for further exploration and improvement in the field of Knowledge Graphs (KGs). One significant direction involves the development of dynamic and scalable knowledge graphs capable of efficiently handling large-scale datasets with evolving relation-

ships. As data grows in volume and complexity, scalable methodologies will be essential for real-time updates and querying in various applications.

Another promising area is cross-domain integration. Combining knowledge graphs from diverse fields, such as healthcare and cybersecurity, can unlock opportunities for interdisciplinary innovation. Developing frameworks that allow seamless integration of these graphs while preserving domain-specific insights will be a crucial challenge to address.

Enhanced ontology management is another area that warrants attention. Automating the process of ontology evolution will enable knowledge graphs to adapt dynamically to emerging trends and data sources, ensuring their relevance and usability over time. This will require innovative tools that incorporate real-time data validation and ontology updates.

Lastly, improving the usability of knowledge graphs through intuitive visualization and querying tools is vital. Such tools can bridge the gap between technical complexity and user accessibility, empowering non-expert users to interact with and derive meaningful insights from knowledge graphs more effectively.

These future directions aim to address the current limitations of KGs while expanding their applicability to real-world problems across various domains.

References

[1] G. Agrawal, K. Pal, Y. Deng, H. Liu, and C. Baral, "Aiseckg: Knowledge graph dataset for cybersecurity education," *AAAI-MAKE 2023: Challenges Requiring the Combination of Machine Learning 2023*, 2023.

[2] B. Abu-Salih, "Domain-specific knowledge graphs: A survey," *King Abdullah II School of Information Technology, The University of Jordan*, 2021.

[3] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[4] X. Zou, "A survey on application of knowledge graph," *Journal of Physics: Conference Series*, vol. 1487, no. 012016, 2020.

[5] Z. Yan and J. Liu, "A review on application of knowledge graph in cybersecurity," 2020.

[6] L. Obrst, P. Chase, and R. Markeloff, "Developing an ontology of the cyber security domain," in *Proceedings of the 2012 Conference on Semantic Technology in Intelligence, Defense and Security (STIDS)*, 2012.

[7] M. Iannacone, S. Bohn, G. Nakamura, J. Gerth, K. Huffer, R. Bridges, E. Ferragut, and J. Goodall, "Developing an ontology for cyber security knowledge graphs," in *Proceedings of the 2015 Cyber and Information Security Research Conference (CISR '15)*, 2015.

[8] C. Andrew, C. Lim, and E. Budiarto, "Knowledge graphs for cybersecurity: A framework for honeypot data analysis," in *2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs)*, pp. 1–10, 2023.

[9] N. Kurbatova and R. Swiers, "Disease ontologies for knowledge graphs," *BMC Bioinformatics*, vol. 22, no. 377, 2021.

[10] Z. Wu, Y. Wan, H. Ma, Q. Chen, J. Gerth, K. Huffer, R. Bridges, E. Ferragut, and J. Goodall, "Construction of knowledge graph of neurodegenerative diseases based

on ontology," in *Proceedings of the 19th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2023.

[11] R. Kulkarni and Y. Haribhakta, "Building the knowledge graph from medical conversational text data and its applications," in *2022 4th International Conference on Advances in Computing, Communication Control, and Networking (ICAC3N)*, pp. 1508–1513, 2022.

[12] S. Zhang, Y. Tang, J. Yan, L. Li, T. Li, J. Li, P. Xie, Y. Gu, J. Xu, Z. Feng, W. Zhang, J. Xia, W. Mayer, H.-Y. Zhang, G.-C. He, and K. He, "A graph-based approach for integrating biological heterogeneous data based on connecting ontology," in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 600–607, 2021.

[13] H. Li, Z. Shi, C. Pan, D. Zhao, and N. Sun, "Cybersecurity knowledge graphs construction and quality assessment," *Complex Intelligent Systems*, 2023.

[14] N. F. Noy and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology," *Stanford University, Stanford, CA, 94305*.

[15] M. Krötzsch, "Ontologies for knowledge graphs?," in *Proceedings of the Center for Advancing Electronics Dresden (cfaed)*, (TU Dresden, Dresden, Germany).

[16] B. Smith, W. Ceusters, B. Klagges, J. Köhler, A. Kumar, J. Lomax, C. Mungall, F. Neuhaus, A. Rector, and C. Rosse, "Relations in biomedical ontologies," *Genome Biology*, vol. 6, no. 5, p. R46, 2005.

[17] O. Shinde and A. Khobragade, "Knowledge graph creation on windows malwares and completion using knowledge graph embedding," in *2022 4th International Conference on Advances in Computing Communication Control and Networking (ICAC3N)*, pp. 1980–1984, 2022.

[18] A. Bordes *et al.*, "Translating embeddings for modeling multi-relational data," *Advances in Neural Information Processing Systems*, 2013.

[19] Z. Sun *et al.*, "Rotate: Knowledge graph embedding by relational rotation in complex space," *International Conference on Learning Representations*, 2019.

[20] T. Trouillon *et al.*, "Complex embeddings for simple link prediction," *International Conference on Machine Learning*, 2016.

[21] B. Yang *et al.*, "Embedding entities and relations for learning and inference in knowledge bases," *International Conference on Learning Representations*, 2014.

[22] R. Patro, "Building knowledge graph," 2021.

[23] V. N. Ioannidis, X. Song, S. Manchanda, M. Li, X. Pan, D. Zheng, X. Ning, X. Zeng, and G. Karypis, "Drkg - drug repurposing knowledge graph for covid-19." https://github.com/gnn4dr/DRKG/, 2020.

[24] P. SANAGAPATI, "Knowledge graph nlp tutorial-(bert,spacy,nltk)," 2021.

[25] K. Mohamed, I. Rodríguez-Rodríguez, and A. Clare, "Healthcare information systems and public health: Supporting the covid-19 response," *Health Informatics Journal*, vol. 26, no. 3, pp. 2263–2275, 2020.

[26] M. Chikina, Y. Fridman, A. Sokolov, R. Sharan, and E. Ruppin, "Towards a more accurate and ecologically relevant evaluation of algorithms for gene function prediction," in *Gene Function Analysis* (W. Dubitzky, M. Granzow, and D. Berrar, eds.), vol. 604 of *Methods in Molecular Biology*, pp. 231–245, Humana Press, 2010.

[27] J. Zhang, S. Köhler, J. D. Overton, C. Smail, A. Özgür, O. Bodenreider, P. N. Robinson, and W. A. Baumgartner, "Ontology-aware deep learning enables ultrarapid biomedical literature classification," *Bioinformatics*, vol. 35, no. 20, pp. 4108–4115, 2019.

[28] P. Xuan, Y. Cao, and T. Zhang, "Drug repositioning through integration of prior knowledge and projections of drugs and diseases," *Bioinformatics*, vol. 35, no. 20, pp. 4108–4119, 2019.

[29] S. Mei and K. Zhang, "A multi-label learning framework for drug repurposing," *Pharmaceutics*, vol. 11, no. 9, p. 466, 2019.

Generated using Undegraduate Thesis LaTeX Template, Version 1.4. Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.

This thesis was generated on Wednesday 1st January, 2025 at 5:58pm.