Artificial Intelligence Evolution
http://ojs.wiserpub.com/index.php/AIE/

Review

A Systematic Review on Knowledge Graphs Classification and Their Various Usages

Mst. Mim Akter[*], **Md-Mizanur Rahoman**

Department of Computer Science and Engineering, Begum Rokeya University, Rangpur, Bangladesh
E-mail: mim.brur.bd@gmail.com

Received: 31 August 2023; **Revised:** 29 October 2023; **Accepted:** 13 November 2023

Abstract: A Knowledge Graph is a directive graph where the nodes state the entities and the edges describe the relationships between the entities of data. It is also referred to as a Semantic Network. The massive amount of data generated every day can be transformed into knowledge via knowledge graphs for the effective use of these data. Knowing the classification of Knowledge Graphs is required to adapt to different requirements of Knowledge Graphs. Knowledge Graphs are primarily classified concerning their building techniques and their usages. In building techniques, it is considered how the Knowledge Graph is built. For example, the graph can be constructed as a triplet, quadruplet, etc., or created from structured data, e.g., database, or unstructured data, e.g., text, image, etc. On the other hand, Knowledge Graphs can be used for various purposes. For example, Knowledge Graphs can be used for Information Retrieval, Semantic Query, etc., or different types of data visualization. Nowadays, Knowledge Graph is one of the trending topics in the modern technology-dependent world. However, clear and specific discussions on the classifications of Knowledge Graphs and their various usages are less available. In this paper, we will describe the classification of knowledge graphs and their various usages in detail so that the readers can get a clear concept of this topic.

Keywords: knowledge graph, knowledge classification, information extraction, knowledge graph construction, entity extraction, relationship extraction

1. Introduction

A Knowledge Graph is a directed labeled graph that describes the relationship between real-world entities and represents them in a network. Real-world entities indicate events, objects, concepts, situations, etc. Knowledge Graph is formed with three primary elements: node, edge, and label. The real-world entities can be considered as nodes. Edges capture the relationship between two nodes. Labels define the signification of the relationships, for example, the brotherhood between two people [1].

Historically, the term Knowledge Graph was introduced by an Austrian linguistician, E. W. Schneider, when he discussed the construction technique of Course Modularization in early 1972 [2]. Later, a project named 'Knowledge Graphs' was started unitedly with The University of Groningen and The University of Twente in the late 1980s. This project mainly gave attention to the schema of the Semantic Networks with edges connected with a definable set of relationships to simplify algebras on the graph. In the following decades, the difference between Knowledge Graph and

DOI: https://doi.org/10.37256/aie.4220233605

Semantic Network faded. In 1985, WordNet was created to represent the semantic relationships between words and their meanings [3]. In 1998, a system named ThinkBase was founded by Andrew Edmonds that delivered fuzzy logic in the graphical context [4]. In 1999, Berners-Lee disclosed his vision of the Semantic Web for the first time in his book ‘Weaving the Web’ [5]. Then, in 2001, Berners-Lee, Hendler, and Lassila published an article based on the progression from the existing Web to the Semantic Web [6]. In 2005, GeoNames was introduced by Marc Wick to represent the relationships between various geographical names and locations and linked entities [7]. In 2006, YAGO was created from the idea of combining information from Wikipedia and WordNet [8]. DBpedia and Freebase were created in 2007 and acted as knowledge repositories for the general use of knowledge [9, 10]. DBpedia mainly focused on extracting structured data from Wikipedia, whereas Freebase also consisted of an extent of public datasets [11, 12]. Neither of them ever designated themselves as ‘Knowledge Graphs’; however, they developed and narrated related concepts. In 2012, Google founded its Knowledge Graph, constructed on DBpedia and Freebase among various sources [1]. They later included Wikipedia, Wikidata, and the CIA World Factbook and also added RDFa, JSONLD, and Microdata content derived from listed web pages [13-15]. Then, the Schema.org vocabulary attached the entity type and the relationship type in this Knowledge Graph. The Google Knowledge Graph became more famous for common use day by day [1]. Thenceforth, the advertisement of using Knowledge Graphs in various large multinationals such as Facebook, Microsoft, LinkedIn, Amazon, etc, have popularized the term [16-19]. In 2019, IEEE released its International Conference on Knowledge Graph, which was mainly based on “Intelligent Computing and Data Mining” and “Big Knowledge” [20-22]. Figure 1 chronologically represents the history of Knowledge Graphs. In this figure, we can see the names of the events and the years when these events occurred.

Usually, Knowledge Graphs are formed with datasets from various sources. To form a Knowledge Graph - schemas, identities, and context should work together. Schemas made the framework of the Knowledge Graph, where identities decide the nodes, as the context determines the setting of the Knowledge Graph. A Knowledge Graph is conceivably constructed by a procedure named Semantic Enrichment. When processing the data, Semantic Enrichment identifies the individual objects and determines the relationship between different objects. In a Knowledge Graph, two nodes connect through an edge and form a triple. Once a Knowledge Graph is completed, we can use the knowledge from the Knowledge Graphs in different types of tasks such as information retrieval, question answering, recommendation, etc [23-25]. A Knowledge Graph can be as powerful as the data it can access.

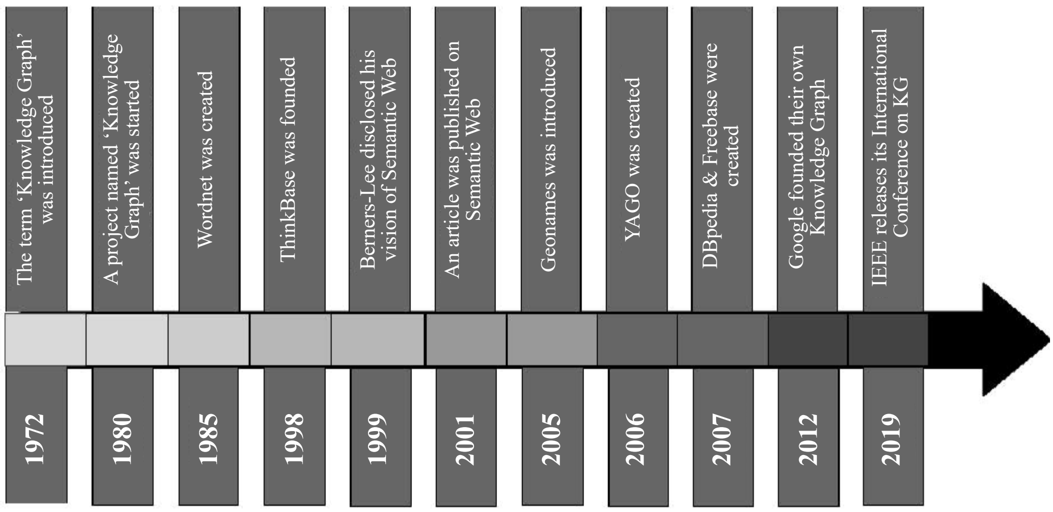

Figure 1. History of Knowledge Graphs

As we know, modern technology is completely data-driven. In every second, there is a massive amount of data. However, we can not use them effectively until we convert the data into knowledge. Knowledge Graph converts the data into knowledge so that we can use them effectively. A Knowledge Graph converts the data into machine-understandable information. Real-world knowledge is situational, layered, and changing. Here, situational, layered, and changing refer to the meaning of the knowledge that can alter depending on situations, the union of concepts, and discoveries. Knowledge Graph turns the data into knowledge to be situational, layered, and changing like real-world knowledge [26]. These knowledge technologies provide the necessary machine and human-understandable context, enable greater data adaptability, and minimize complexity and cost. Knowledge Graph is one of the best solutions for structuring the world's information systems and unifying data from multiple sources [27]. Thus, knowledge technologies enhance the influence, power, and validity of machine learning and data science. That is why, nowadays, it has become a trending topic for the world, especially for machine learning and data science engineers.

Despite being such an important topic, the discussion about the classification of Knowledge Graphs is limited and unclear. However, for effective utilization of Knowledge Graphs, it is essential to know their classifications and their usages. In this paper, we will discuss various types of Knowledge Graphs based on their building techniques and usages. By reading this paper, the readers will get:

1. A clear concept about the classification of Knowledge Graphs.
2. A detailed idea about the usage of various kinds of knowledge graphs.
3. Several future research directions on Knowledge Graphs.

The other sections of this paper are sorted as follows: Section 2 describes the generic construction method for Knowledge Graphs. Section 3 states the various kinds of Knowledge Graphs and their usage. Section 4 discusses the future research directions on Knowledge Graphs. Lastly, section 5 describes the conclusion of this paper.

2. Generic Knowledge Graph Construction method

Generally, a Knowledge Graph can be constructed using two approaches. The first one is the Top-down Knowledge Graph Construction Approach, and the other one is the Bottom-up Knowledge Graph Construction Approach. The Top-down Knowledge Graph Construction Approach starts with a predefined ontology, providing a structured and guided approach to data representation. On the other hand, the Bottom-Up Knowledge Graph Construction Approach is characterized by its agility. It doesn't depend on a predefined ontology and can adapt to diverse data sources. In this section, we will discuss comprehensively about these two approaches.

2.1 *Top-down Knowledge Graph Construction approach*

In the Top-down Approach, the ontology and schema should be identified first, and then knowledge instances will be attached to the knowledge base. For showing the original cases of knowledge graphs, this method emphasizes the precise domain ontologies [28].

The Top-down Knowledge Graph Approach first requires determining a subject domain and finding out a list of demands. Second, it needs to plan a Conceptual Model to gather the entities of interest, their inter-relationships, and the classes. Third, it requires the creation of the Logical Model and the Physical Model, which will assemble logical narration and descriptions of the gathered entities and relationships. Fourth, it requires to perform technological implementation and improvement. For this, it is required to take into account the coding language as well as the Knowledge Graph advancement platforms. The last step requires applying the Knowledge Graph as a service to bear the society reuse and give feedback. In this procedure, the knowledge of a specific domain is converted into machine-understandable statements [29-30]. Figure 2 describes the whole Top-down Knowledge Graph Construction Approach. In this figure, a step-by-step Top-down Knowledge Graph Construction Approach from domain selection to building the proper Knowledge Graph is illustrated.

The Top-down Knowledge Graph Construction Approach is well-suited for scenarios where the data representation is structured and controlled. As a result, the applications of this approach are seen in the fields of Corporate Knowledge Management, Healthcare and Medical Reports, Financial Services, Government and Public Administration, Scientific Research and Academia, etc. The Top-down Knowledge Graph Construction Approach provides structured, consistent data for efficient querying and organization. It is advantageous when working with domain-specific knowledge. However, it is disadvantageous at times because of rigidity. Sometimes, the predefined ontology can be inflexible, making it less suitable for rapidly evolving or dynamic data environments.

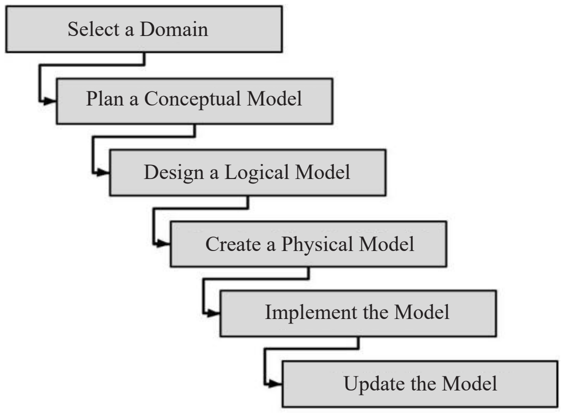

Figure 2. The top-down approach for Knowledge Graph Construction

In summary, the Top-down Knowledge Graph Construction Approach first determines the domain. Subsequently, it generates the Conceptual Model, the Logical Model, and the Physical Model, respectively. After that, it performs the implementation and evaluates the proficiency of the Knowledge Graph. At last, depending on the evaluation, it updates the Knowledge Graph frequently to meet the goals.

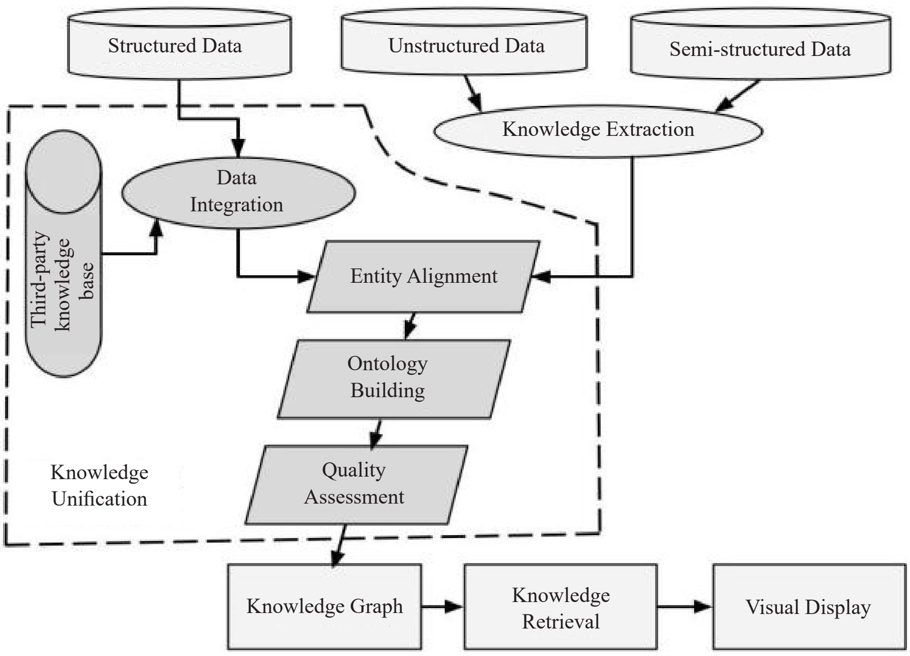

Figure 3. The bottom-up approach for Knowledge Graph Construction

2.2 *Bottom-up Knowledge Graph Construction Approach*

In the Bottom-up Knowledge Graph Construction Approach, the entities and relationships are extracted from structured metadata, unstructured and semi-structured data, and other crowed sources data. After extracting the entities and relationships, they are used progressively to build the Knowledge Graph structure. In this approach, the schema or ontology is developed based on the data rather than being predefined. The ontology is created as new entity types and relationships are discovered.

Figure 3 describes the Bottom-Up Approach to Knowledge Graph building. From this figure, specific concepts are gained on the stages of Knowledge acquisition, Knowledge Unification, Knowledge Storing, and Recovery that cover the overall idea of the Bottom-Up Approach.

Here, we see that knowledge building in the Bottom-up Knowledge Graph Construction Approach is an iteratively improving procedure with Knowledge Acquisition, Knowledge Unification, Knowledge Storing, and Recovery. In the Knowledge Acquisition procedure, the knowledge from various data sources needs to be collected through Knowledge Extraction. Numerous tools have been built for Knowledge Extraction through the years, such as Stanford NER [31], OpenNLP [32], AIDA [33], ReVerb [34], CiceroLite [35], Open Calais [36], Wikipedia Miner toolkit [37], etc. After extracting knowledge, the next and most important phase is Knowledge Unification. It is an iterative procedure that has to continuously build an ontology and assess the quality of the ontology. The objective of Knowledge Unification is to determine Entity Alignment and Ontology Building iteratively. Ontology Building will not terminate until the outcomes of the Quality Assessment fulfill the needs. Entity Alignment decides whether various entities direct the same object or not. If the Quality Assessment of the Ontology and the Knowledge Graph do not fulfill the demand, the procedure of building and unification of the Knowledge Graph will be repeated [28, 38].

The Bottom-up Knowledge Graph Construction Approach is well-suited for extracting knowledge from various data sources like web content, scientific literature, and social media for research, recommendations, and data enrichment. It is capable of processing a massive amount of data and rapidly creating an extensive knowledge graph. However, in this procedure, an existing challenge is the proper rational assertion, representations for the entities, and relationships in the resulting Knowledge Graph. Most of the time, the entities and the relationships need to be marked out by the domain specialists and knowledge engineers, where existing Knowledge Graphs can be recycled [29].

3. Different types of Knowledge Graphs and their usages

The Knowledge Graphs can be classified into various classifications depending on their domain, modality, openness, temporality, implementation techniques, and construction methodologies. In this section, we will discuss the classification of Knowledge Graphs depending on their various building techniques and usages.

For a clearer concept, the classification of various kinds of Knowledge Graphs and their usage is represented in Table 1. In this table, the first column describes the name of the categories, the second column indicates the name of the subcategories of these categories, and the third column represents the usages of these subcategories.

3.1 *Implementation-based Knowledge Graph*

Based on the implementation techniques, all the Knowledge Graphs can be categorized into two categories. The first one is the Resource Description Framework (RDF), and the other one is Labeled Property Graphs (LPG).

3.1.1 *Resource Description Framework (RDF)*

The Resource Description Framework is a universal framework used to illustrate structured metadata, especially in the territory of the World Wide Web. It manifested under the support of the World Wide Web Consortium (W3C). It qualifies the structured metadata to encrypt, interchange, and reuse. It does not determine any specific semantics, syntax, and structure of the data; instead, it gives the authority for the resource description communities to describe the elements of metadata as they need to meet their specific goals [39].

Table 1. Various types of Knowledge Graphs and their usages

Name of Categories	Name of the Subcategories	Usages
Implementation-based Knowledge Graph	Resource Description Framework (RDF)	Semantic Web Technologies, DBpedia, Freebase, OpenCyc, Wikidata, YAGO, Linked Data, Government Data, Scientific Data, Financial Data, Geospatial Data, Education, Media and Entertainment, Life Sciences and Healthcare, Cultural Heritage
	Labeled Property Graphs (LPG)	Neo4j Graph Database, DBpedia, Freebase, Content Management, Social Networks, Biological Networks, Geospatial Data, Recommendation Engines, E-commerce, Fraud Detection, Semantic Web Applications
Modality-based Knowledge Graph	Text Knowledge Graphs	WordNet, ConceptNet, Freebase, GeoNames, Google Knowledge Graph, Wikidata, DBpedia, YAGO, Facebook Graph, Diffbot, Educational Knowledge Graphs, Legal Knowledge Graphs, Medical Knowledge Graphs, Scientific Literature, Sentiment Analysis, Semantic Search, News and Media Analysis, Government and Public Data, E-commerce and Product Knowledge, Customer Support and Chatbots
	Visual Knowledge Graphs	Google Knowledge Graph, DBpedia, YAGO, Wikidata, Diffbot, Facebook Graph, GeoNames, Computer Vision and Object Recognition, Virtual and Augmented Reality, Automotive and Transportation, Medical Imaging and Healthcare, Social Media Analysis, Fashion and Retail, Environmental Science, Media and Entertainment, Astronomy and Space Exploration
	Multi-modal Knowledge Graphs	Google Knowledge Graph, DBpedia, YAGO, Diffbot, Wikidata, GeoNames, Facebook Graph, Education and e-learning, Healthcare and Medical Imaging, Autonomous Vehicles, Robotics and Automation, Media and Entertainment, Cultural Heritage and Museums, Language Learning and Translation, Virtual Reality and Augmented Reality, Environmental Monitoring
Openness-based Knowledge Graph	Open Knowledge Graphs	DBpedia, Data Commons, Wikidata, Freebase, OpenCyc, YAGO, ConceptNet, WordNet, Scientific Research, Linguistics and Natural Language Processing, Healthcare and Medical Research, Education and e-Learning, Agriculture and Food Systems, Business and Industry, Public Health, Social Sciences, Government and Public Data
	Proprietorial Knowledge Graphs	Google Knowledge Graph, Google Knowledge Vault, Facebook Graph, Cyc, Microsoft Bing Knowledge Graph, Apple's Siri Knowledge Graph, Amazon Product Graph, LinkedIn's Economic Graph, Walmart Retail Knowledge Graph, IBM Watson Knowledge Graph, Uber's Trip Data Graph, Automotive Industry, Insurance Industry, Financial Institutions, Pharmaceutical and Healthcare Industry
Temporal-based Knowledge Graph	Static Knowledge Graphs	DBpedia, ConceptNet, WordNet, Artificial Intelligence and Natural Language Processing (NLP), Historical Research, Cross-disciplinary Research, Language and Linguistics, Content Recommendation, Education and Textbook Creation, Language Learning and Translation, Museum and Art Curation, Library and Information Science
	Dynamic Knowledge Graphs	DBpedia Live, Freebase, Wikidata, YAGO, OpenCyc, Google Knowledge Graph, Diffbot, Google Knowledge Vault, Stock Market and Financial Analysis, News and Media Monitoring, Social Media Analytics, Healthcare and Medical Records, Supply Chain and Logistics, Weather and Climate Data, Real Estate and Property Data, Agriculture and Crop Monitoring, Agriculture and Crop Monitoring, Transportation and Traffic Management
Domain scope-based Knowledge Graph	General Knowledge Graphs	YAGO, Google Knowledge Graph, Diffbot, DBpedia, Freebase, Wikidata, WordNet, ConceptNet, NELL, BableNet, Education and E-Learning, Cross-domain Semantic Search, Data Integration and Interoperability, Semantic Web and Linked Data, Text Analytics and Natural Language Processing (NLP), Information Retrieval and Question Answering, Media and Entertainment, Cultural and Historical Research, Smart Assistants and Chatbots, Market Research and Competitive Analysis
	Domain Knowledge Graphs	GeoNames, MusicBrainz, CIA World Factbook, OpenCyc, DrugBand, Travel and Tourism, Food and Culinary Arts, Sports and Athletics, Language Learning and Linguistics, Home Automation and IoT, Fashion and Apparel, Astronomy and Space Exploration, Aviation and Air Travel, Legal and Law, Pet and Animal Care
Construction Methodology-based Knowledge Graph	Rule-based Knowledge Graphs	It helps to build the Knowledge Graph by using different rules, parsing techniques, and patterns. For example, YAGO extracts relationships using regular expressions and identifies the relationships between entities using pattern-matching
	Learning-based Knowledge Graphs	Google Knowledge Graph, Facebook Graph, WordNet, ConceptNet, Search Engines, Recommendation Systems, Chatbots and Virtual Assistants, Speech and Audio Processing, Autonomous Vehicles, Image and Video Analysis, Virtual Reality and Augmented Reality, Sentiment Analysis, Game Development, Content Generation, Content Moderation

The Resource Description Framework is a robust framework for data description and is utilized in the construction of Knowledge Graphs. The Resource Description Framework assembles the data via triples (three positional statements). A Resource Description Framework triple usually contains a subject, predicate, and object. The Resource Description Framework discloses the relationships between resources where resources have their properties. The nodes in a Resource Description Framework Graph can be resources described by a unique resource identifier (URI), literals, or auxiliary blank nodes. The edges refer to the predicates. Named graphs or contexts handle the elements of the graph. Each edge in the graph represents information, and this information can be visualized as a quadruple (subject, predicate, object, context). By the Resource Description Framework descriptions, different types of resources can be manifested using a uniform structure building with three interconnected data pieces [40-41].

The classes, predicates, and named graphs are all identified by their URIs. Thus, they can appear as nodes and edges in the graph, respectively, receiving their designations. With these designations, both data and schema can be accessed and operated in an identical model [42]. Figure 4 represents an instance of the Resource Description Framework Knowledge Graph. The nodes in this figure are named for better understandability; all of those must have their own URIs, like the list that is given below:

Lena = https://www.linkedin.com/in/lenasmith.

Lucas = https://www.linkedin.com/in/lucasrobert.

Jonas = https://www.linkedin.com/in/Jonasdial.

San Diego = https://en.wikipedia.org/wiki/San Diego.

California = https://en.wikipedia.org/wiki/California.

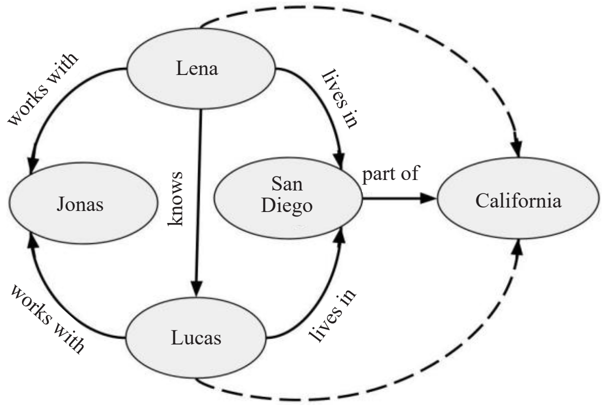

Figure 4. An example of the Resource Description Framework Knowledge Graph

Knowledge Graph that, illustrated in the Resource Description Framework, gives the best infrastructure for data interconnection, composition, integration, and reuse. However, the Resource Description Framework has some limitations. It doesn't allow the labels or properties to be joined with the edges in the graph, and this is realized as an inconvenience compared to Labeled Property Graphs (will be discussed in the following section). However, we can say that by interconnecting structured metadata, the Resource Description Framework gives the capability to convert the web into a more effective and proficient data resource [43-44].

The usages of the Resource Description Framework are observed to construct rich and extensive Knowledge Graphs, as well as for solving a diverse range of real-life problems. Some examples of such applications are Semantic Web Technologies, DBpedia, Freebase, OpenCyc, Wikidata, YAGO, Linked Data, Government Data, Scientific Data, Financial Data, Geospatial Data, Education, Media and Entertainment, Life Sciences and Healthcare, Cultural Heritage, etc. [8, 45-46].

3.1.2 *Labeled Property Graphs (LPG)*

A Labeled Property Graph (LPG) is one kind of graph database. The Labeled Property Graph contains nodes and edges where any individual node or edge has a label and properties of its own. The key difference between the Labeled Property Graphs and the Resource Description Framework is the Labeled Property Graph has the ability to assemble properties at the nodes and edges of the network; however, the Resource Description Framework does not [47-48].

The Labeled Property Graph Model is the best universal objective data model illustration that we have at the present time. The Labeled Property Graph is built using the labels and properties. The individual names of the nodes and edges in a graph are known as labels, whereas the individual properties of these nodes and edges are known as properties. The properties of the nodes or edges can be keys or, values or any attributes. In a Labeled Property Graph Model, the most significant things are the labels and the properties that are joined with the nodes and the edges of the graph. The edges have two characteristics; they have always direction and have a start and an end node, thus making the graph a directed graph [49-50].

For a better understanding, Figure 5 describes an example of the Labeled Property Graph. In Figure 5, a directed graph can be seen, which consists of three nodes and four edges. These nodes provide interconnected information that is directed by the edges. The nodes and edges of this graph may have individual labels and properties that help to access more information directly. For example, because of the label and property of node - ‘Person 1’, the name, birth year, and occupation of this node can be accessed easily. Similarly, the labels and properties of other nodes or edges help to obtain more information about them immediately.

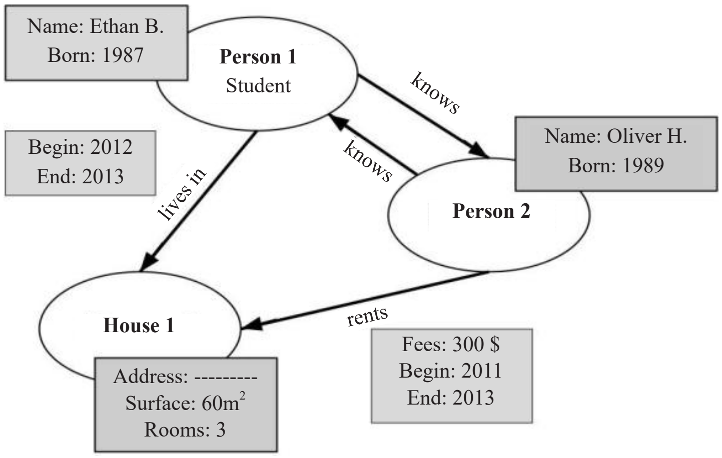

Figure 5. An example of the Labeled Property Graph

Labeled Property Graphs do not have an ideal guideline for data representation. Instead, institutions operating with Labeled Property Graphs generate their semantics as there are no ideal ontologies. Each Labeled Property Graph attains its identical querying language [51]. In summary, by assembling properties and labels to the nodes and edges, the Labeled Property Graph gives the ability to access more information about the nodes and edges directly.

Labeled Property Graphs are used to build various comprehensive Knowledge Graphs, as well as for resolving real-life issues in multiple domains. Some examples of such usages are Neo4j Graph Database, DBpedia, Freebase, Content Management, Social Networks, Biological Networks, Geospatial Data, Recommendation Engines, E-commerce, Fraud Detection, Semantic Web Applications, etc. [45].

3.2 *Modality-based Knowledge Graph*

Depending upon the modality of the metadata and other resources of data, the Knowledge Graphs can be categorized into three different categories: Text Knowledge Graph, Visual Knowledge Graph, and Multi-modal

Knowledge Graph [52].

3.2.1 *Text Knowledge Graph*

A Knowledge Graph constructed from pure textual data is known as a Text Knowledge Graph. This type of Knowledge Graph comprises text information and databases. The Text Knowledge Graphs are also referred to as Conventional Knowledge Graphs.

The Text Knowledge Graphs are mainly a formation of structured human knowledge, a Semantic Network that indicates the relationships between different entities in the text. The relationships between the entities are represented by triples (subject, predicate, object). The triple is formed with nodes and edges. The nodes represent the entities, and the edges represent the relationships between the entities. The Text Knowledge Graph can be a single text entity or a summation of different text entity relations [53-54].

Let's consider a text: "Noah is a girl. She is a student. Clara studies with her. They live in New York and study at New York University". The Text Knowledge Graph for the given text is illustrated in Figure 6. In this figure, the text is converted into a Knowledge Graph by triples, where the nodes of the triple indicate the entities and the edges indicate the relationship between these nodes.

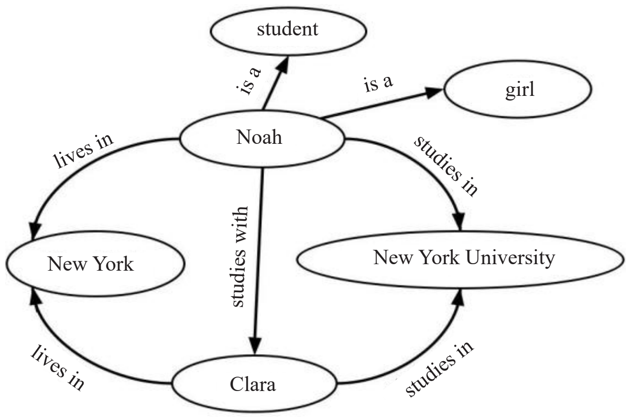

Figure 6. An illustration of the Text Knowledge Graph

The Text Knowledge Graph concentrates on the entities and their relationships to database and text knowledge. It extracts knowledge from the text data and constructs a Knowledge Graph to illustrate the relationships between the entities with the help of triples [55-56].

The usages of Text Knowledge Graphs are observed to construct rich and comprehensive Knowledge Graphs, as well as for resolving real-life problems in diverse domains. Some examples of such applications are Wordnet, ConceptNet, Freebase, GeoNames, Google Knowledge Graph, Wikidata, DBpedia, YAGO, Diffbot, Facebook Graph, Educational Knowledge Graphs, Legal Knowledge Graphs, Medical Knowledge Graphs, Scientific Literature, Sentiment Analysis, Semantic Search, News and Media Analysis, Government and Public Data, E-commerce and Product Knowledge, Customer Support and Chatbots, etc. [8, 45, 57-58].

3.2.2 *Visual Knowledge Graph*

A Knowledge Graph that is built with visual data such as images and videos is known as a Visual Knowledge Graph. Visual Knowledge Graphs mainly refer to Image Knowledge Graphs and Video Knowledge Graphs or their combination graphs [59].

Image Knowledge Graphs are constructed from images and the description of the images. At first, a description text and scene graph of the image have been generated. After that, by going through the different types of knowledge extraction processes such as image entity recognition, feature image extraction, visual relation detection, etc, the scene graph and description text have been transformed into the Image Knowledge Graph [53]. In the same way, we can construct Video Knowledge Graphs with videos and sequential information, images, and audio of the videos. By going through the different types of knowledge extraction processes for videos, audio, and images, such as Action Detection, Time Range Acquisition, Sound Classification, Image Entity Recognition, Feature Image Extraction, Visual Relation Detection, etc, the sequential information, images, audios and the videos construct the Video Knowledge Graph. Figure 7 represents an example of the Visual Knowledge Graph. This is mainly an Image Knowledge Graph, which has been built with images and descriptions of the images.

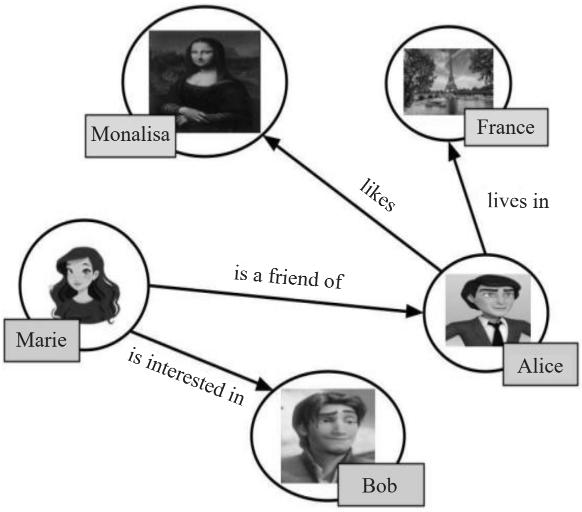

Figure 7. An example of the Visual Knowledge Graph

The Visual Knowledge Graphs consist of visual knowledge that is extracted from the visual data. The visual data can be images or videos. Therefore, with the help of images and videos, we can generate the Visual Knowledge Graph.

Visual Knowledge Graphs are used in building rich and extensive Knowledge Graphs, as well as for solving real-life problems in different domains. Some examples of such usages are Google Knowledge Graph, DBpedia, Diffbot, YAGO, Wikidata, Facebook Graph, GeoNames, Computer Vision and Object Recognition, Virtual and Augmented Reality, Automotive and Transportation, Medical Imaging and Healthcare, Social Media Analysis, Fashion and Retail, Environmental Science, Media and Entertainment, Astronomy and Space Exploration, etc. [8, 45, 57].

3.2.3 *Multi-modal Knowledge Graph*

A Multi-modal Knowledge Graph consists of both textual and visual data. A Knowledge Graph that is constructed from multi-modal data is known as a Multi-modal Knowledge Graph. Multi-modal involves the styles of viewing, listening, and touching in the sensory view or understood by various sensors. In general, multi-modal comprises different data media ordinations such as text, image, video, audio, etc. By using the heterogeneous multi-model data, we can create a Multi-modal Knowledge Graph [53, 60].

A Multi-modal Knowledge Graph is a combination of Text Knowledge Graphs, Image Knowledge Graphs, Video Knowledge Graphs, and Audio Knowledge Graphs. At first, the Text Knowledge Graphs, the Image Knowledge Graphs, the Video Knowledge Graphs, and the Audio Knowledge Graphs will be created, respectively. After that, the unification of these graphs will build the Multi-modal Knowledge Graph. However, while processing the data, the description texts for the images and videos will use the process of knowledge extraction from text. Figure 8 represents a Multi-modal Knowledge Graph. In this figure, the graph has been constructed using multi-modal data such as textual data, visual data, and audio data.

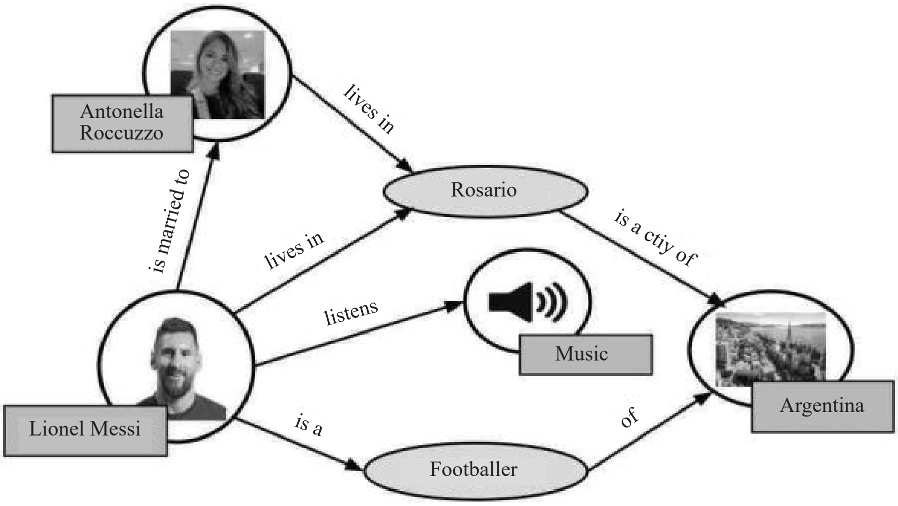

Figure 8. A Multi-modal Knowledge Graph

The Multi-modal Knowledge Graph is a connective Semantic Network for arranging and handling multi-modal data, which can efficiently arrange crumbled, cross-modal, and cross-database data. By concentrating on Knowledge Graph construction with a single modality, we are efficiently turning down a heap of other information that Knowledge Graphs are supposed to have. And if we were capable of involving it in the learning procedure, it has the strength to progress the aggregate achievement of our models. The Multi-modal Knowledge Graph is the best solution for this type of problem [55, 61].

The usages of Multi-modal Knowledge Graphs are observed to construct rich and comprehensive Knowledge Graphs. They are also used for resolving real-life problems in diverse domains. Some examples of such applications are Google Knowledge Graph, Diffbot, DBpedia, YAGO, Wikidata, GeoNames, Facebook Graph, Education and e-learning, Healthcare and Medical Imaging, Autonomous Vehicles, Robotics and Automation, Media and Entertainment, Cultural Heritage and Museums, Language Learning and Translation, Virtual Reality and Augmented Reality, Environmental Monitoring, etc. [8, 45, 57, 62].

3.3 *Openness-based Knowledge Graph*

Depending on the availability of metadata, crowed-sources data like social media and other resources, Knowledge Graphs are classified into two categories. They are narrated as either Public Knowledge Graph and Private Knowledge Graph or Open Knowledge Graph and Proprietorial Knowledge Graph [27].

3.3.1 *Open Knowledge Graph*

The Knowledge Graphs can be constructed with openly accessible metadata, crowed-sources data, and different data sources. The Knowledge Graphs with openly accessible resources refer to those Knowledge Graphs that are built from the knowledge of the resources that are available for everyone. Anyone can freely access and use the knowledge gained from this type of Knowledge Graph; that is why we designate these types of graphs as Open Knowledge Graphs [63-65].

A leading example of the Knowledge Graph with openly accessible resources is Wikidata. Wikidata is a jointly accomplished Open Knowledge Graph that supplies data for Wikipedia and for different kinds of applications on the web. Wikidata uses multilingual data. A Wikidata Knowledge Graph can aid in enlarging and advancing the proficiency of data in Wikipedia. Figure 9 represents a portion of the Wikidata Knowledge Graph.

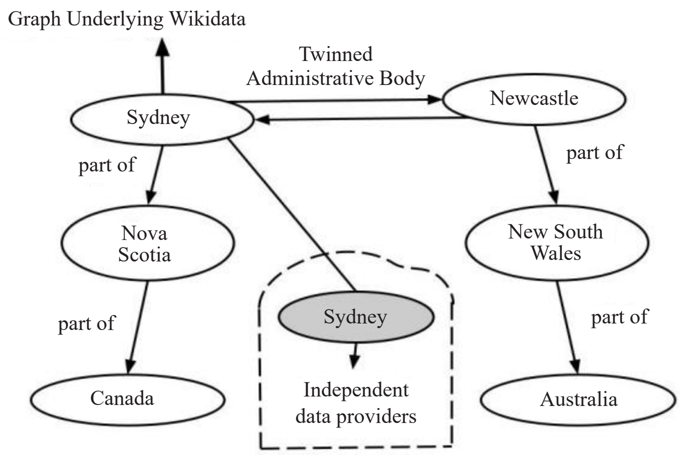

Figure 9. A portion of the Wikidata KG

In Figure 9, the Wikipedia page for the city of Sydney has been considered, which is situated in the New South Wales state of Australia. New South Wales also has a city named Newcastle. Sydney is a city in New South Wales, Australia. Wikipedia also has an entry of a city named Sydney, which is situated in Nova Scotia, Canada. As we can see, Australia and Canada have two cities with the same name, Sydney. Therefore, the relationship between these two cities is called "Twin Towns". Newcastle City is a sister city to Sydney. That is why their relationship is known as "Sister Towns". The Wikidata illustration of Sydney comprises a relationship called Twinned Administrative Body, which comprises the town of Newcastle.

A current case of another Open Knowledge Graph is from the Data Commons endeavor, whose objective is to create publicly obtainable data that is easily accessible and usable. Data Commons fulfills the essential cleaning and adding of data from different publicly available data sources and supplies availability to the resulting Knowledge Graph [27].

In Brief, the usages of the Open Knowledge Graphs are noticed to build rich and extensive Knowledge Graphs, as well as for solving various types of real-life problems. Some examples of such usages are DBpedia, Data Commons, Wikidata, Freebase, OpenCyc, YAGO, ConceptNet, WordNet, Scientific Research, Linguistics and Natural Language Processing, Healthcare and Medical Research, Education and e-learning, Agriculture and Food Systems, Business and Industry, Public Health, Social Sciences, Government and Public Data, etc. [27, 45, 58].

3.3.2 *Proprietorial Knowledge Graph*

The Proprietorial Knowledge Graph indicates those Knowledge Graphs that are constructed with the proprietary resources of private enterprises or organizations. These Knowledge Graphs are built with datasets that are not accessible to everyone; however, they can be accessible to the people of a particular organization or enterprise [27]. The Proprietorial Knowledge Graphs are not freely available and usable for ordinary people. They are available and usable for the people of the enterprise or the organization that has ownership of the graphs [66-68].

In today's world, corporate data usually exists with many independent databases and unstructured resources. So, Data Integration is essential to handle the operation of modern enterprises and organizations. Moreover, the extensive shift to online activities for around every enterprise has produced the gathering of highly massive amounts of worthy user behavior data from different localities. Furthermore, the rapid increase of data attainable from third-party data sellers is providing enterprises with invaluable information that must be integrated with their data for more feasible business performances. With the help of the Proprietorial Knowledge Graphs, enterprises integrate these valuable data and convert them into intelligence.

Let's go through an instance of how a company can benefit from its Proprietorial Knowledge Graph. Suppose a piece of news has been published narrating that company "A" has been recorded for bankruptcy because of the pandemic. With this news, the suppliers of this company will encounter financial tension. Now, if company "B", which is a supplier to the company "A", is suffering economic tension, it might be hoped that the same tension is also endured successively by suppliers to "B". This is known as the Supply-chain Relationship of a company, which is essential to trace for the company. As we know, a customer can create his '360-degree view' of a company with the data about that customer inside and outside of the company. Here, the data from outside the company refer to third-party information and resources. In the same way, the company can also build a '360-degree view' of every customer by assembling third-party data with the data of the customers from the company's databases. This helps a company to build its own Proprietorial Knowledge Graph. With the help of the Proprietorial Knowledge Graphs of a company, the company can trace the 'Supply-chain Relationship' for that company and support finding the stressed suppliers whose trouble may deserve observation [27, 66]. Figure 10 describes an example of a "360-degree view". This figure shows the effect of news reports on customers or the company and the relationships of the company or customers with the suppliers.

Nowadays, Knowledge Graphs are becoming a popular and easy solution for converting data into intelligence in enterprises because of the relative comfort of generating and visualizing the formation and the availability of built-in analytical operations. The Proprietorial Knowledge Graph allows an enterprise to create and use its Knowledge Graph, which is built with its own data, integrating with the data available from third-party sellers [27].

The usages of Proprietorial Knowledge Graphs are seen in rich and extensive Knowledge Graphs of different enterprises or organizations. Some examples of such usages are Google Knowledge Graph, Google Knowledge Vault, Facebook Graph, Cyc, Microsoft Bing Knowledge Graph, Apple's Siri Knowledge Graph, Amazon Product Graph, LinkedIn's Economic Graph, Walmart Retail Knowledge Graph, IBM Watson Knowledge Graph, Uber's Trip Data Graph, Automotive Industry, Insurance Industry, Financial Institutions, Pharmaceutical and Healthcare Industry, etc. [45].

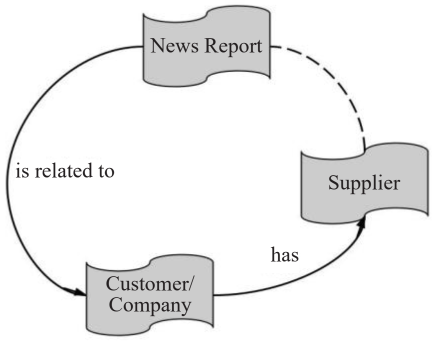

Figure 10. An example illustration for a "360-degree view"

3.4 *Temporal-based Knowledge Graph*

The digital world is completely technology-based, and information is the most powerful weapon in this modern world. To store the knowledge of the information structurally, we often create Knowledge Graphs. Some Knowledge Graphs can be updated with time, whereas some others remain unchanged over time. According to the timeliness of contained knowledge, the Knowledge Graphs are classified into two classes. The first one is the Static Knowledge Graph, and the other one is the Temporal or Dynamic Knowledge Graph [52, 69].

3.4.1 *Static Knowledge Graph*

A static Knowledge Graph is the Knowledge Graph that is generated from genuine knowledge, and it does not change over time. In these graphs, we can not add, remove, edit, or change the nodes or edges over time as per our requirements. Since these types of graphs can not be updated over time, it is essential to build such types of graphs with original and permanent information [70-72].

Figure 11 shows the visualization of a Static Knowledge Graph by which the concept of this topic will be more clear. In this figure, at the time of t_1, there is a Knowledge Graph G, and the graph remains the same at the time of t_2, t_3,

t_4, t_5, ... $t_n - 1$ and t_n, respectively. Therefore, the Knowledge Graph, G, will be a static Knowledge Graph. Over time, the information and relationships between entities will not be changed in this graph.

The Static Knowledge Graph does not depend on time. With the changing of time, the entities and relationships of the static Knowledge Graphs will remain unchanged. The knowledge of the Static Knowledge Graph can not be updated. That is why genuine information should be used to build a Static Knowledge Graph [72-74].

The usages of Static Knowledge Graphs are noticed in various comprehensive Knowledge Graphs, as well as for solving multiple problems in different domains. Some examples of such applications are DBpedia, ConceptNet, WordNet, Artificial Intelligence and Natural Language Processing (NLP), Historical Research, Cross-disciplinary Research, Language and Linguistics, Content Recommendation, Education and Textbook Creation, Language Learning and Translation, Museum and Art Curation, Library and Information Science, etc. [45, 58].

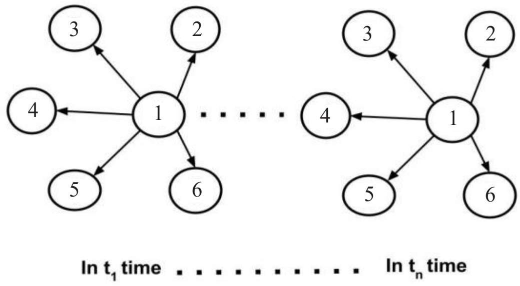

Figure 11. A Static Knowledge Graph

3.4.2 *Dynamic Knowledge Graph*

Knowledge Graph that changes over time as per requirement is known as a Dynamic Knowledge Graph. Dynamic Knowledge Graphs are also called Temporal Knowledge Graphs, Time-varying Knowledge Graphs, or Evolving Knowledge Graphs. In these types of graphs, we can add, remove, edit, or change the nodes or edges over time as per our requirements [70, 75].

If we want to make an application highly efficient that operates on a Knowledge Graph, it is necessary to construct a proper Knowledge Graph in the first place. Though there are a lot of people involved in the data gathering, it is pretty challenging to supply consummate data primarily. Different problems can appear, such as unfinished data, noises, obscurity, or lost data. Besides, there exists an abundance of data that narrates complicated, ongoing earth in which entities and relationships update through time. So, it is essential for the changing world to build such types of Knowledge Graphs from which we can add, remove, edit, or change the information over time. This is why Dynamic Knowledge Graphs are getting popular day by day [70, 73].

Figure 12 visualizes a Dynamic Knowledge Graph to give a clear idea of the Dynamic Knowledge Graph. In this figure, at the time of t_1, there is a Knowledge Graph G, and the graph changes with the changing of time from t_1 to t_2, t_2 to t_3, t_3 to t_4, t_4 to t_5 ...... $t_n - 1$ to t_n respectively. Therefore, the Knowledge Graph G will be a Dynamic Knowledge Graph. In this graph, the connection between nodes and edges will not remain the same all the time. The information and relationships between entities will be updated with time.

If a Knowledge Graph has a set of entities and relationships that are dependent on time, then we can consider such type of Knowledge Graph as the Dynamic Knowledge Graph. Constructing these graphs qualifies us to handle the circumstance where time dependency is needed to learn thoroughly a Knowledge Graph with time-dependent entities. In a Dynamic Knowledge Graph, as time passes, a node can update itself by changing the respective relationships with other nodes [76-77].

The usages of Dynamic Knowledge Graphs are observed in building rich and extensive Knowledge Graphs, as well as for resolving various problems in diverse domains. Some examples of such applications are DBpedia Live, Freebase, Wikidata, YAGO, OpenCyc, Google Knowledge Graph, Diffbot, Google Knowledge Vault, Stock Market and Financial

Analysis, News and Media Monitoring, Social Media Analytics, Healthcare and Medical Records, Supply Chain and Logistics, Weather and Climate Data, Real Estate and Property Data, Agriculture and Crop Monitoring, Agriculture and Crop Monitoring, Transportation and Traffic Management, etc. [45, 78].

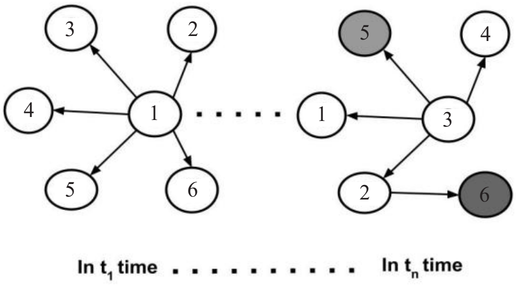

Figure 12. A Dynamic Knowledge Graph

3.5 *Domain scope-based Knowledge Graph*

Depending on the domain scope, the Knowledge Graphs are divided into two different classes: General Knowledge Graph and Domain Knowledge Graph [52, 79-80]. The General Knowledge Graph commonly gathers domain-independent information or knowledge from different domains to construct the Knowledge Graph, whereas the Domain Knowledge Graph collects domain-dependent information or knowledge from a particular domain for building the Knowledge Graph.

3.5.1 *General Knowledge Graph*

The Knowledge Graph, created with domain-independent knowledge or information, is known as General Knowledge Graph. General Knowledge Graphs are also referred to as Open-world Knowledge Graphs or Cross-domain Knowledge Graphs or Domain-independent Knowledge Graphs. These types of Knowledge Graphs can be generated with the information of different domains [80-81].

A General Knowledge Graph is usually portrayed as a directive graph where the nodes of the graph describe the real-world entities, and the edges of the graph indicate the relationships between the entities. The relationship between two entities can be illustrated as an RDF triple. Here, the triple consists of two nodes that narrate the entities and an edge that narrates the relationship between the entities [82]. Figure 13 represents a General Knowledge Graph for its entities and relations. This figure shows a Knowledge Graph built with the knowledge of different domains.

The Knowledge Graph is a Semantic Web that states the relationships of the nodes through edges where nodes narrate the entities and edges narrate the relationships between the entities. Examples of different Domain-independent Knowledge Graphs are NELL [83-84], YAGO [85-86], BabelNet [87-88], Cyc [89-92], DBPedia Knowledge Base [10-11, 93], Google Knowledge Graph [94-95], etc. These Knowledge Graphs collect the knowledge from various domains and combine the knowledge to build a General Knowledge Graph [96].

In Short, the applications of General Knowledge Graphs are noticed in rich and comprehensive Knowledge Graphs, as well as for resolving a diverse range of real-life problems. Here are some examples of such usages: YAGO, Google Knowledge Graph, Diffbot, DBpedia, Freebase, Wikidata, WordNet, ConceptNet, NELL, BableNet, Education and ELearning, Cross-domain Semantic Search, Data Integration and Interoperability, Semantic Web and Linked Data, Text Analytics and Natural Language Processing (NLP), Information Retrieval and Question Answering, Media and Entertainment, Cultural and Historical Research, Smart Assistants and Chatbots, Market Research and Competitive Analysis, etc. [45, 58].

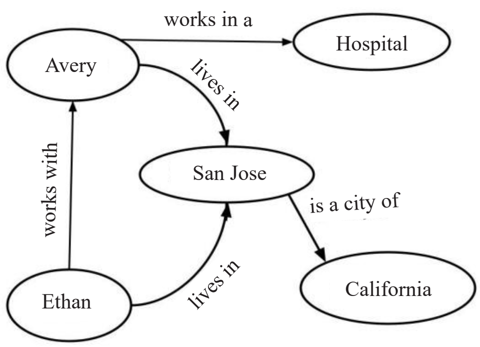

Figure 13. A general Knowledge Graph

Entertainment, Cultural and Historical Research, Smart Assistants and Chatbots, Market Research and Competitive Analysis, etc. [45, 58].

3.5.2 *Domain Knowledge Graph*

The Knowledge Graph, which is constructed depending on a particular domain, is known as the Domain Knowledge Graph. The knowledge to build this kind of graph is collected from the particular domain. Information from any other domain is not allowed in this kind of graph. The Domain Knowledge Graphs are also referred to as Domain-specific Knowledge Graphs [80-81, 97].

The definition of the Domain Knowledge Graph indicates three main aspects:

1. The Logical Model of this graph is illustrated by a distinctive and predefined domain ontology introduced to cover the domain of interest.

2. The Domain Knowledge Graphs have to be resolutely contextual to accost a piece of specific subject-matter information.

3. The Physical Model of the Domain Knowledge Graph is illustrated as a labeled graph in which data semantics are improved with a particular theoretical description of entities and the relations between these entities.

The domain of the Domain Knowledge Graphs can be any specific area or subject, such as Medicine, Movies, Science and Engineering, Travel, Food, Education, Society, Service, Finance, Religion, Poems, Politics, Literature, Books, etc. [98-108].

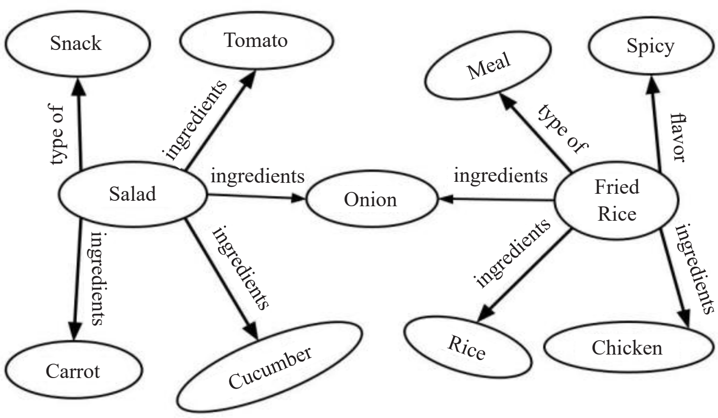

Figure 14. The portion of the food domain Knowledge Graph

Figure 14 depicts a Knowledge Graph in the food domain. All of the knowledge that constructed the above Knowledge Graph has been included from a specific domain, i.e., the food domain. Any information or knowledge that does not exist in the food domain has not been allowed for building this Domain Knowledge Graph. In this way, a Domain Knowledge Graph represents knowledge of a specific domain so that anyone can attain any information required easily about that particular domain. The Domain Knowledge Graph can illustrate complicated domain knowledge in an arranged manner and has acquired huge enrichment in applied applications [81-82].

The usages of the Domain Knowledge Graphs are observed to build rich and extensive Knowledge Graphs, as well as for solving real-life problems in different domains. Some instances of Domain Knowledge Graphs are GeoNames, MusicBrainz, CIA World Factbook, OpenCyc, and DrugBand [45]. Here, GeoNames contains only geographical information, MusicBrainz contains only music-related information, DrugBand contains only drug and drug-targets-related information, and so on. Moreover, some additional usages of the Domain Knowledge Graphs are Travel and Tourism, Food and Culinary Arts, Sports and Athletics, Language Learning and Linguistics, Home Automation and IoT, Fashion and Apparel, Astronomy and Space Exploration, Aviation and Air Travel, Legal and Law, Pet and Animal Care, etc.

3.6 *Construction methodology-based Knowledge Graph*

Depending on the construction methodology, Knowledge Graphs can be categorized into two different categories. The first one is the Rule-based Knowledge Graph, and the other one is the Learning-based Knowledge Graph. The Knowledge Graph that has been generated following different types of rules is known as a Rule-based Knowledge Graph, and the knowledge graph that has been created by continuous learning is called a Learning-based Knowledge Graph.

3.6.1 *Rule-based Knowledge Graph*

The Rule-based Knowledge Graph follows particular rules, especially different parsing techniques or patterns, to build the Knowledge Graph. The Rule-based Knowledge Graph is mainly categorized into two categories. They are Pattern- following Knowledge Graphs and Parser-based Knowledge Graphs. The Pattern-following Knowledge Graphs have been constructed depending on various patterns that are followed by the Knowledge Graphs. The Parser-based Knowledge Graphs have been built with different parsing techniques that are followed by the Knowledge Graphs [109-111].

3.6.1.1 *Pattern-following Knowledge Graph*

In a Knowledge Graph, patterns define the various ways to discover the relationships between the entities. For constructing a Knowledge Graph, we can consider different kinds of patterns. In this section, we will discuss some of the more often-used patterns. Among all of the patterns of the Knowledge Graph, the more frequently usable patterns are the Transitive Relation Pattern, Transitive Closure Pattern, Cyclic Detection, Defining a relation as the composition of other relations, Ordering Pattern, etc. [112-119].

Generally, Transitive Relation refers to a relation imposed between consecutive elements of a series; it should also be imposed between any other two elements obeying the sequence. For instance, “if X is greater than Y and Y is greater than Z, then X is greater than Z” - this is a Transitive Relation. This kind of relationship can be affixed temporarily in the Knowledge Graph, but it will be illogical for the bigger Knowledge Graphs, where the relationships will seem semantically incorrect. Suppose we have two cities with the same name in different countries. For example, ‘Sydney is situated in Australia’ and ‘Sydney is situated in Canada’, then as per the Transitive Relation, we will say Australia is situated in Canada, which is semantically incorrect. To solve this problem, we will follow the Transitive Relation Pattern of the Knowledge Graph, which will affix the entities of the Knowledge Graph with logical and semantic consequences and conclude that “As Australia and Canada are not situated in the same boundary, Australia is not situated in Canada”.

The Transitive Closure Patterns work to implement the Transitive Relations that haven’t been present till now in the Knowledge Graph but could be proficient. For example, if A is related to B, B is related to C, and C is related to D, then we can say A is related to C and D in the same way B is related to D. This is how the Transitive Closure Pattern works.

The Cyclic Detection is a pattern used in the Knowledge Graphs to determine the cyclic relationships of the knowledge. Besides, we can use the pattern which can identify a relation as the mixture of various relations. Ordering Pattern is one of the most frequently used patterns in Knowledge Graph building. Many relationships usually indicate some sequence in which we should find the primary and final members of such sequence. This is where the Ordering Pattern works.

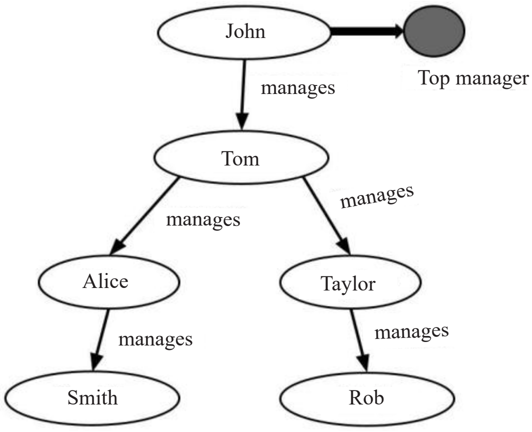

Figure 15. An example of the Ordering Pattern

Figure 15 describes the Ordering Pattern of a Knowledge Graph. It indicates the managerial infrastructure of a company. If we like to discover the top manager of the company, we can follow a rule to describe the top manager as an individual who manages somebody but is not managed by anybody else. In this criteria, the pattern will return 'John' as the top manager of the company. Similarly, we can discover the junior managers of the company who are managed by someone and do not manage anybody else. In this case, the graph will return 'Smith' and 'Rob' as the output.

3.6.1.2 *Parser-based Knowledge Graph*

The Knowledge Graph can be generated by performing several parsing techniques. The Knowledge Graph, whose entities and relationships between entities have been derived from different parsing techniques. Depending on the parsing techniques, the Knowledge Graphs are classified into two categories. They are Dependency Parser-based Knowledge Graphs and Constituency Parser-based Knowledge Graphs.

3.6.1.2.1 *Dependency parser-based Knowledge Graph*

Dependency parser-based Knowledge Graphs refer to those Knowledge Graphs that have been created performing dependency parsing on the knowledge. This type of Knowledge Graph follows the Dependency Parser to define the entities and the relationships between entities in the graph [120-123].

Dependency parsing is a procedure of resolving the dependencies within the words in a sentence to explore the grammatical structure of the sentence. Different types of tags have been used in dependency parsing to describe the relationships between any two words in a sentence [124-127]. These tags are referred to as the dependency tags. For example, let's think about the phrase 'silent environment'. Here, the word 'silent' modifies the noun 'environment'. Hence, a dependency is present from 'environment' to 'silent'. In this case, the word 'environment' will behave as a head, and 'silent' will act as a dependent or a child. To describe this dependency, we use the 'amod' tag, which indicates the adjectival modifier. Till now, the Universal Dependency (version 2) taxonomy has used 37 universal dependency relations, and every dependency relation has an individual dependency tag. For instance, for case marking, the 'case' tag is used; for indicating auxiliary, the 'aux' tag is used; for referring to the classifier 'clf' tag is used; for determiner 'det' tag is used, for unspecified dependency 'dep' tag is used, for passive auxiliary 'aux:pass' tag is used, for marker 'mark' tag is used, for object ‘obj’ tag is used, etc.

Figure 16 describes the dependency layout of a sentence. In this structure, we have used ‘nsubj’ tag, ‘root’ tag, ‘dobj’ tag, ‘amod’ tag, ‘det’ tag, ‘nmod’ tag and ‘case’ tag to describe the nominal subject, root, direct object, adjectival modifier, determiner, nominal modifier and case marking respectively. Following the dependency parsing process, the data and information can be transformed into dependency parser-based Knowledge Graphs.

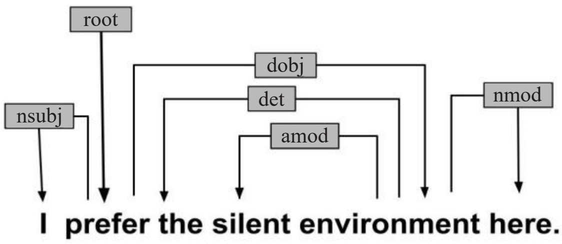

Figure 16. Dependency layout of a sentence

3.6.1.2.2 *Constituency parser-based Knowledge Graph*

The Knowledge Graphs generated by performing constituency parsing on the knowledge are known as constituency parser-based Knowledge Graphs. These Knowledge Graphs follow the constituency parsing process to identify the entities and describe the relationships between entities [128-129].

Constituency parsing is the procedure of exploring sentences by splitting them into different sub-phrases where each sub-phrase belongs to an individual grammar class, namely noun phrase or NP, verb phrase or VP, etc. [54, 130]. For a better understanding, let’s consider an example. Suppose we have a sentence - “I bought a book”. After performing the constituency parsing on the sentence, we have a constituency parse tree. Figure 17 represents this constituency parse tree.

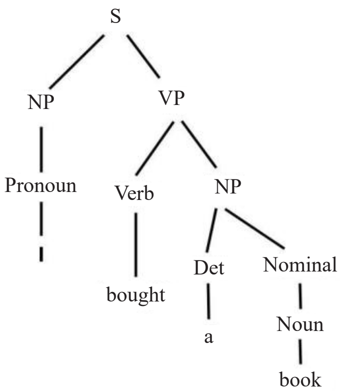

Figure 17. A visual layout of constituency parse tree

In Figure 17, the sentence has been divided into two sub-phrases - noun phrase and verb phrase. After that, the verb phrase is again split into verb and noun phrases. In constituency parsing, sentences are thus divided into different grammatical categories. Following the constituency parsing, the knowledge can be converted into Constituency-based Knowledge Graphs [54, 130].

The usages of Rule-based Knowledge Graphs are observed to help build Knowledge Graphs using different rules, parsing techniques, and patterns. For example, YAGO extracts relationships using regular expressions and identifies the relationships between entities using pattern matching [131].

3.6.2 *Learning-based Knowledge Graph*

The Learning-based Knowledge Graph can be generated and updated by learning. In the learning-based technique, the data and information have to go through a Machine Learning or Deep Learning Model to extract the knowledge and build the Knowledge Graph. Based on this, the Learning-based Knowledge Graphs are classified into two categories. They are Machine Learning-based Knowledge Graphs and Deep Learning-based Knowledge Graphs. To generate a Knowledge Graph using Machine Learning or Deep Learning, the information has to go through some process such as Relation Extraction, Entity Extraction, etc.

3.6.2.1 *Machine Learning-based Knowledge Graph*

Machine Learning-based Knowledge Graphs indicate those Knowledge Graphs that have been constructed and updated as required using a Machine Learning Model. Machine Learning is a field of Artificial Intelligence that makes the system capable of learning from data, recognizing patterns, and making decisions with minimum human interference. With the help of a Machine Learning Model, it has become easier to generate a Knowledge Graph.

For generating a Knowledge Graph using Machine Learning, the data has to go through several processes. At first, the data needed to be collected and preprocessed. Then, the Machine Learning model works with the data to extract the entities and the relationships between entities by Entity Extraction and Relation Extraction methods. Then, with the help of the extracted entities and relationships, the Machine Learning Model creates triples and finally generates the Knowledge Graph. Machine Learning also trains the Knowledge Graph by inferring the absent entities and relationships of the graph. Finally, Machine Learning assesses the generated model, more specifically the Knowledge Graph, and makes the development of the model if required [132-133].

After creating a Knowledge Graph using Machine Learning, the created Knowledge Graph can also act like a Machine Learning Model, and by using this model, much better and richer data can be fed to the Machine Learning Algorithms. Consequently, various important Natural Language Processing-based Machine Learning tasks, including Question Answering, Recommendation, Information Extraction, Information Retrieval, Semantic Query, etc, can be performed spontaneously [134]. At the same time, when working with Machine Learning Models, Machine Learning engineers need to deal with a huge number of data. The Machine Learning-based Knowledge Graph Models help to handle such huge data without difficulty.

3.6.2.2 *Deep Learning-based Knowledge Graph*

Deep Learning-based Knowledge Graph introduces those Knowledge Graphs that are generated and updated if required by the Deep Learning Models. Deep Learning is a subcategory of Machine Learning. The Deep Learning Model needs lower human interference as compared to Machine Learning. This is because the Deep Learning Model learns independently from the environment and past experiences, whereas the Machine Learning Model requires more human interference to learn [135]. This is one of the main differences between the Deep Learning and the Machine Learning. Besides, the layout of the Machine Learning algorithms is rather simple, such as Decision Tree, Linear Regression, etc, where the Deep Learning Models are built with Multi-layered Artificial Neural Networks.

To construct a Knowledge Graph using Deep Learning, at first, we have to collect the structured and unstructured information and preprocess these data. Then, the entities and the relations between entities will be identified, represented as nodes and edges in the graph, respectively. Afterward, the entities and the relationships will be converted into numerical vectors called embeddings. Then, a Deep Learning Model, such as a Graph Neural Network, will be trained on the embeddings to identify the missing entities or relationships. Using the trained model, new entities and relationships will be inferred to enlarge the graph [136]. At last, based on a test dataset, the performance of the generated model will be assessed, and development will be made if required. This is the procedure to convert data into a Knowledge Graph using Deep Learning.

The usages of Learning-based Knowledge Graphs are observed to construct rich and comprehensive Knowledge

Graphs, as well as for resolving a diverse range of real-life problems. Some examples of such usages are Google Knowledge Graph, Facebook Graph, WordNet, ConceptNet, Search Engines, Recommendation Systems, Chatbots and Virtual Assistants, Speech and Audio Processing, Autonomous Vehicles, Image and Video Analysis, Virtual Reality and Augmented Reality, Sentiment Analysis, Game Development, Content Generation, Content Moderation, etc.

4. Discussion

In this paper, we have systematically discussed the classification of Knowledge Graphs and their various usages so that the readers can get a clear concept of the topic. In modern engineering, numerous Knowledge Graphs, including Wordnet, Freebase, DBpedia, and YAGO, have been developed. These Knowledge Graphs play an enabling role in gathering, organizing, and effectively managing knowledge from large-scale data. Their usage can be realized widely in various Artificial Intelligence-based tasks, namely Recommendation Systems, Search Engines, and Question-Answering systems. Considering the broad application aspects, Knowledge Graphs have become a research focus in Natural Language Processing in recent years [137]. Therefore, in section 4.1, we want to share some potential research directions on Knowledge Graphs. In section 4.2, we will discuss some challenges that we encountered while classifying the Knowledge Graphs.

4.1 *Future research directions*

Future researchers can focus on four primary areas within the field of Knowledge Graphs:

1. Construction of Knowledge Graphs.
2. Quality Assessment of Knowledge Graphs.
3. Schema/Ontology Alignment of Knowledge Graphs.
4. Scalability Enhancement of Knowledge Graphs.

Knowledge Graph Embedding: The Knowledge Graph Embedding provides a mechanism for the seamless integration of knowledge from a Knowledge Graph into real-world applications. The goal is to map entities and relations in the Knowledge Graph to a continuous vector space in such a way that the geometric relationships between these vectors reflect the semantic relationships in the graph [138].

• **Knowledge Graph Creation:** Knowledge Graph Creation involves the manual or semi-manual process of collecting, extracting, and structuring data from various sources to create a Knowledge Graph [139].

• **Generative AI-driven Knowledge Graph Generation:** Generative AI-driven Knowledge Graph Generation uses artificial intelligence and NLP techniques to automate the creation of Knowledge Graphs from unstructured text data [139].

• **Enhancing Large Language Model Explainability through Knowledge Graph Integration:** Large Language Models generate text-based responses, but their decision-making processes can be opaque. Integrating knowledge graphs allows these models to draw from structured information to provide more transparent and accurate responses [140].

On the other hand, to control the quality of the Knowledge Graphs, it is necessary to evaluate the credibility of the Knowledge Graphs [141]. Apart from this, Knowledge Graph Cleansing, Error Checking, and Error Rectifying are also essential factors for ensuring the quality of the Knowledge Graphs. To cleanse the Knowledge Graph, check and rectify the errors, and evaluate the credibility of Knowledge Graphs, there are several processes to go through that requires time and effort. Therefore, automatic Quality Assessment of Knowledge Graph can be an area of interest for future researchers. In the area of Quality Assessment of Knowledge Graph, researchers have the opportunity to explore several promising directions for future research, including:

• **Knowledge Graph Cleansing:** Knowledge Graph Cleansing involves the removal of missing or erroneous data values within a knowledge graph to enhance its overall data quality [142].

• **Credibility Evaluation of Knowledge Graph:** Credibility Evaluation evaluates the accuracy and trustworthiness of data sources contributing to the Knowledge Graph to ensure its overall quality [143].

• **Error Detection and Correction of Knowledge Graphs:** Error Detection and Correction in Knowledge Graphs aims to identify and rectify inconsistencies or mismatches in triples of a Knowledge Graph involving the head entity, tail entity, and relations to enhance the overall quality of the Knowledge Graph [144-145].

Moreover, when expanding a Knowledge Graph by linking multiple Knowledge Graphs, it is crucial to ensure the proper alignment of schemas and ontologies between them. The Ontology Alignment of Knowledge Graphs requires time and effort. As a result, automatic Ontology Alignment of Knowledge Graphs can be a potential area of interest for future researchers. In the realm of Ontology Alignment of Knowledge Graphs, future research directions may include:

• **Schema Matching:** Schema Matching is a method of identifying attributes in a Knowledge Graph that are either linguistically similar or represent equivalent information for enabling data integration and interoperability [146].

• **Entity Matching:** Entity Matching is the process of aligning entities or concepts in different knowledge graphs that refer to the same real-world objects, enabling cross-graph data linkage and search [147].

• **Relation Matching:** Relation Matching is a method that aligns relationships between entities in a graph-based schema, enhancing the accuracy of relation alignment tasks in multilingual datasets [148].

In addition, with the continuous influx of data from diverse sources, Knowledge Graphs need to expand to manage this increasing volume of data efficiently. As a result, for accommodating this larger volume of data, the Scalability Enhancement of Knowledge Graphs is an essential factor. Therefore, the Scalability Enhancement of constructed Knowledge Graphs can be one of the potential future research directions. Furthermore, a potential future research direction for researchers lies in conducting a comparative study on the time complexity analysis of Information Extraction or Information Reasoning from constructed Knowledge Graphs versus other Information Extraction techniques like Named Entity Recognition, Relationship Extraction, etc.

4.2 *Challenges in Knowledge Graphs classification*

In the paper, we have discussed the Classification of Knowledge Graphs and their various usages in detail. However, it may have some challenges because dilemmas have been encountered in some places while making the classification. For example, Chen et al. [52] refer to three types of modality-based Knowledge Graphs, whereas Peng et al. [54] refer to modality-based Knowledge Graphs as two types. As many categories as possible have been discussed to avoid such ambiguities. Actually, the Classification of Knowledge Graphs can be represented in several ways. In this paper, our main purpose is to give an outline of the Classification of Knowledge Graphs so that the readers can understand and use this concept for their future work.

5. Conclusion

Although Knowledge Graph has an immense effect on today's world by reconfiguring the system of storing, processing, and delivering knowledge, the Classification of Knowledge Graphs is still a less talked-about topic. Despite being an important topic to discuss, the Classification of Knowledge Graphs is an unclear topic because of its low discussion. Moreover, the discussion about the various usages of Knowledge Graphs is also limited. In this paper, we have given a detailed and clear idea of the Classification of Knowledge Graphs and their various usages.

In our work, we have combined all the possible categories of Knowledge Graphs from different perspectives based on their building techniques and usages. After finishing this paper, the readers will get a clear concept of the different categories of Knowledge Graphs depending on their various construction techniques and applications. At the same time, the readers will get several future research directions on Knowledge Graphs.

Conflict of interest

The authors have no conflict of interest related to this publication.

References

[1] Ehrlinger L, Wöß W. Towards a definition of knowledge graphs. In: *Joint Proceedings of the Posters and Demos*

Track of 12th International Conference on Semantic Systems - SEMANTiCS2016 and 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS16). Leipzig, Germany; 2016. p. 2.
[2] Schneider EW. *Course Modularization Applied: The Interface System and Its Implications For Sequence Control and Data Analysis*. 1973.
[3] Fellbaum C. *WordNet: An Electronic Lexical Database*. Massachusetts: MIT Press; 1998.
[4] Transworld Research & Development Co. Ltd. Corporation Bermuda, Flatts Village Hamilton #3 St. James Court FL04 Bermuda. (1998, November 16). ThinkBase, US Trademark No. 75589756. United States Patent and Trademark Office.
[5] Berners-Lee T, Fischetti M. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. San Francisco: Harper; 1999.
[6] Berners-Lee T, Hendler J, Lassila O. The semantic web. *Scientific American*. 2001; 284(5): 34-43.
[7] Bol PK. On an infrastructure for historical spatial analysis. *Perspectives on History*. 2012.
[8] Rebele T, Suchanek F, Hoffart J, Biega J, Kuzey E, Weikum G. YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. In: *The Semantic Web-ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings*. Springer, Cham; 2016. p. 177-185. Available from: doi: 10.1007/978-3-319-46547-0_19.
[9] Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: a collaboratively created graph database for structuring human knowledge. In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. 2008. p. 1247-1250. Available from: doi: 10.1145/1376616.1376746.
[10] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. Dbpedia: A nucleus for a web of open data. In: *The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007+ ASWC 2007, Busan, Korea, November 11-15, 2007*. Proceedings Springer; 2007. p. 722-735. Available from: doi: 10.1007/978-3-540-76298-0_52.
[11] Lehmann J, Isele R, Jakob M, Jentzsch A, Kontokostas D, Mendes PN, et al. Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*. 2015; 6(2): 167-195.
[12] Pellissier Tanon T, Vrandečić D, Schaffert S, Steiner T, Pintscher L. From freebase to wikidata: The great migration. In: *Proceedings of the 25th International Conference on World Wide Web*. 2016. p. 1419-1428. Available from: doi: 10.1145/2872427.2874809.
[13] Wikipedia. *Wikipedia*. Pedia Press; 2004.
[14] Vrandečić D. Wikidata: A new platform for collaborative data collection. In: *Proceedings of the 21st International Conference on World Wide Web*. New York, NY, USA: Association for Computing Machinery; 2012. p. 1063-1064.
[15] Agency CI. *The CIA World Factbook 2010*. Skyhorse Publishing Inc.; 2009.
[16] Bellomarini L, Fakhoury D, Gottlob G, Sallinger E. Knowledge graphs and enterprise AI: the promise of an enabling technology. In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. Macao, China: IEEE; 2019. p. 26-37. Available from: doi: 10.1109/ICDE.2019.00011.
[17] Meroño-Peñuela A, Lisena P, Martínez-Ortiz C. Knowledge Graphs of linked data. In: *Web Data APIs for Knowledge Graphs: Easing Access to Semantic Data for Application Developers*. Switzerland: Springer, Cham; 2021. p. 1-9.
[18] Nigam VV, Paul S, Agrawal AP, Bansal R. A review paper on the application of knowledge graph on various service providing platforms. In: *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. Noida, India: IEEE; 2020. p. 716-720.
[19] McHugh J, Cuddihy PE, Williams JW, Aggour KS, Kumar VS, Mulwad V. Integrated access to big data polystores through a knowledge-driven framework. In: *2017 IEEE International Conference on Big Data (Big Data)*. Boston, MA, USA: IEEE; 2017. p. 1494-1503.
[20] Gao Y, Möller R, Wu X, Kotagiri R. *2019 IEEE International Conference on Big Knowledge, ICBK 2019, Beijing, China, November 10-11, 2019*. IEEE; 2019.
[21] Liao F, Ma L, Yang D. Research on construction method of knowledge graph of US military equipment based on BiLSTM model. In: *2019 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS)*. Shenzhen, China: IEEE; 2019. p. 146-150.
[22] Wu X, Wu J, Fu X, Li J, Zhou P, Jiang X. Automatic knowledge graph construction: A report on the 2019 icdm/icbk contest. In: *2019 IEEE International Conference on Data Mining (ICDM)*. Beijing, China: IEEE; 2019. p. 1540-1545.
[23] Wang C, Yu H, Wan F. Information retrieval technology based on knowledge graph. In: *2018 3rd International Conference on Advances in Materials, Mechatronics and Civil Engineering (ICAMMCE 2018)*. Hangzhou, China:

Atlantis Press; 2018. p. 291-296.
[24] Yasunaga M, Ren H, Bosselut A, Liang P, Leskovec J. QA-GNN: Reasoning with language models and knowledge graphs for question answering. *arXiv* [Preprint]. 2021. Available from: doi: 10.48550/arXiv.2104.06378.
[25] Wang H, Zhang F, Zhang M, Leskovec J, Zhao M, Li W, et al. Knowledge-aware graph neural networks with label smooth- ness regularization for recommender systems. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Anchorage, AK, USA: Association for Computing Machinery; 2019. p. 968-977.
[26] Xie K, Jia Q, Jing M, Yu Q, Yang T, Fan R. Data analysis based on Knowledge Graph. In: Barolli L, Takizawa M, Enokido T, Chen HC, Matsuo K. (eds.) *Advances on Broad-Band Wireless Computing, Communication and Applications*. BWCCA 2020. Lecture Notes in Networks and Systems, vol. 159. Switzerland: Springer, Cham; 2020. p. 376-385. Available from: doi: 10.1007/978-3-030-61108-8_37.
[27] Chaudhri VK, Baru CK, Chittar N, Dong XL, Genesereth MR, Hendler JA, et al. Knowledge Graphs: Introduction, history and, perspectives. *AI Magazine*. 2022; 43(1): 17-29. Available from: doi: 10.1609/aimag.v43i1.19119.
[28] Zhao Z, Han SK, So IM. Architecture of Knowledge Graph construction techniques. *International Journal of Pure and Applied Mathematics*. 2018; 118(19): 1869-1883.
[29] Ma X. Knowledge graph construction and application in geosciences: A review. *Computers & Geosciences*. 2022; 161: 105082. Available from: doi: 10.1016/j.cageo.2022.105082.
[30] Tan J, Qiu Q, Guo W, Li T. Research on the construction of a knowledge graph and knowledge reasoning model in the field of urban traffic. *Sustainability*. 2021; 13(6): 3191. Available from: doi: 10.3390/su13063191.
[31] Costa CM, Veiga G, Sousa A, Nunes S. Evaluation of stanford ner for extraction of assembly information from instruction manuals. In: *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (Icarsc)*. Coimbra, Portugal: IEEE; 2017. p. 302-309.
[32] Karthick S, Arun C, Selvakumarasamy S, Arora G. Query processing using open NLP tool. *Indian Journal of Science and Technology*. 2016; 9: 32. Available from: doi: 10.17485/ijst/2016/v9i32/98668.
[33] Wibisono A, Roos M, Marshall MS. The AIDA Toolkit-a tool for users and developers. In: *Proceedings of the Workshop on Semantic Web Applications and Tools for Life Sciences*. Berlin, Germany; 2010.
[34] Fader A, Soderland S, Etzioni O. Identifying relations for open information extraction. In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK: Association for Computational Linguistics; 2011. p. 1535-1545.
[35] Gangemi A. A comparison of knowledge extraction tools for the semantic web. In: *The Semantic Web: Semantics and Big Data: 10th International Conference, ESWC 2013*. Berlin, Heidelberg: Springer; 2013. p. 351-366.
[36] Butuc MG. Semantically enriching content using opencalais. *Editia*. 2009; 9: 77-88.
[37] Milne D, Witten IH. An open-source toolkit for mining Wikipedia. *Artificial Intelligence*. 2013; 194: 222-239.
[38] Meilicke C, Chekol MW, Ruffnelli D, Stuckenschmidt H. Anytime bottom-up rule learning for Knowledge Graph completion. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*. Macao, China: AAAI Press; 2019. p. 3137-3143.
[39] Miller E. An introduction to the resource description framework. *D-Lib Magazine*. 1998; 13: 3-11.
[40] Corby O, Dieng R, Hebert C. A conceptual graph model for W3C resource description framework. In: *Conceptual Structures: Logical, Linguistic, and Computational Issues: 8th International Conference on Conceptual Structures, ICCS 2000*. Berlin, Heidelberg: Springer; 2000. p. 468-482.
[41] Pan JZ. Resource description framework. *Handbook on ontologies*. Berlin, Heidelberg: Springer. 2009. p. 71-90.
[42] Powers S. *Practical RDF: solving problems with the resource description framework*. California: O'Reilly Media, Inc.; 2003.
[43] Hartig O. Reconciliation of RDF* and property graphs. *arXiv* [Preprint]. 2014. Available from: doi: 10.48550/arXiv.1409.3288.
[44] Candan KS, Liu H, Suvarna R. Resource description framework: metadata and its applications. *ACM SIGKDD Explorations Newsletter*. 2001; 3(1): 6-19. Available from: doi: 10.1145/507533.507536.
[45] Färber M, Ell B, Menne C, Rettinger A. A comparative survey of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web Journal*. 2015; 1(1): 1-5.
[46] Le Grange JJ, Lehmann J, Athanasiou S, Garcia-Rojas A, Giannopoulos G, Hladky D, et al. The geoknow generator: Managing geospatial data in the linked data web. In: *Proceedings of the Linking Geospatial Data Workshop*. 2014.
[47] Angles R. The property graph database model. In: *Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management*. Cali, Colombia; 2018.

[48] Baken N. Linked data for smart homes: Comparing RDF and labeled property graphs. In: *Proceedings of 8th Linked Data in Architecture and Construction Workshop-LDAC2020*. Dublin, Ireland; 2020. p. 23-36.
[49] Hou Y, Chen H, Li C, Cheng J, Yang MC. A representation learning framework for property graphs. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining KDD '19*. New York, NY, USA: Association for Computing Machinery; 2019. p. 65-73. Available from: doi: 10.1145/3292500.3330948.
[50] Bunakov V. Metadata integration with labeled-property graphs. In: *Metadata and Semantic Research: 13th International Conference, MTSR 2019*. Rome, Italy: Springer; 2019. p. 441-448.
[51] Anikin D, Borisenko O, Nedumov Y. Labeled property graphs: SQL or NoSQL? In: *2019 Ivannikov Memorial Workshop (IVMEM)*. Velikiy Novgorod, Russia: IEEE; 2019. pp. 7-13. Available from: doi: 10.1109/IVMEM.2019.00007.
[52] Chen Z, Wang Y, Zhao B, Cheng J, Zhao X, Duan Z. Knowledge Graph completion: A review. *IEEE Access*. 2020; 8: 192435-192456.
[53] Peng Jh, Hu X, Huang W, Yang J. What is a multi-modal Knowledge Graph: A survey. *Big Data Research*. 2023; 32: 100380.
[54] Martinez-Rodriguez JL, Lopez-Arevalo I, Rios-Alvarado AB. Openie-based approach for knowledge graph construction from text. *Expert Systems with Applications*. 2018; 113: 339-355.
[55] Wilcke W, Bloem P, de Boer V, van't Veer R. End-to-end learning on multimodal Knowledge Graphs. *arXiv* [Preprint]. 2023. Available from: doi: 10.48550/arXiv.2309.01169.
[56] Kertkeidkachorn N, Ichise R. T2kg: An end-to-end system for creating knowledge graph from unstructured text. In: *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI Press; 2017.
[57] Upshall M. Extracting Meaning from Web Content. *eLucidate*. 2016; 13(1).
[58] Speer R, Chin J, Havasi C. Conceptnet 5.5: An open multilingual graph of general knowledge. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. San Francisco, California, US: AAAI Press; 2016. vol. 31.
[59] Lu W, He W, Wu J, Zhang Y. A critical review of additive manufacturing technology in rehabilitation medicine via the use of visual knowledge graph. *Virtual and Physical Prototyping*. 2023; 18(1): e2132265.
[60] Kannan AV, Fradkin D, Akrotirianakis I, Kulahcioglu T, Canedo A, Roy A, et al. Multimodal knowledge graph for deep learning papers and code. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. New York, United States: Association for Computing Machinery; 2020. p. 3417-3420.
[61] Mousselly-Sergieh H, Botschen T, Gurevych I, Roth S. A multimodal translation-based approach for knowledge graph representation learning. In: *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. New Orleans, Louisiana: Association for Computational Linguistics; 2018. p. 225-234.
[62] Chen D, Li Z, Gu B, Chen Z. Multimodal named entity recognition with image attributes and image knowledge. In: *Database Systems for Advanced Applications: 26th International Conference, DASFAA 2021*. Taipei, Taiwan: Springer; 2021. p. 186-201.
[63] Hogan A, Blomqvist E, Cochez M, d'Amato C, Melo Gd, Gutierrez C, et al. Knowledge graphs. *ACM Computing Surveys (CSUR)*. 2021; 54(4): 1-37.
[64] Garijo D, Osorio M, Khider D, Ratnakar V, Gil Y. OKG-Soft: An open knowledge graph with machine readable scientific software metadata. In: *2019 15th International Conference on eScience (eScience)*. San Diego, CA, USA: IEEE; 2019. p. 349-358.
[65] Fu C, Chen T, Qu M, Jin W, Ren X. Collaborative policy learning for open knowledge graph reasoning. *arXiv* [Preprint]. 2019. Available from: doi: 10.48550/arXiv.1909.00230.
[66] Aasman J. Transmuting information to knowledge with an enterprise knowledge graph. *IT Professional*. 2017; 19(6): 44-51.
[67] Gomez-Perez JM, Pan JZ, Vetere G, Wu H. Enterprise knowledge graph: An introduction. In: *Exploiting Linked Data and Knowledge Graphs in Large Organisations*. Switzerland: Springer, Cham; 2017. p. 1-14.
[68] Song D, Schilder F, Hertz S, Saltini G, Smiley C, Nivarthi P, et al. Building and querying an enterprise knowledge graph. *IEEE Transactions on Services Computing*. 2017; 12(3): 356-369.
[69] Ji S, Pan S, Cambria E, Marttinen P, Philip SY. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*. 2021; 33(2): 494-514.
[70] Le T, Nguyen H, Le B. A survey of the link prediction on static and temporal knowledge graph. *Journal on Information Technologies & Communications*. 2021; 2021(2): 51-84.
[71] Liang K, Meng L, Liu M, Liu Y, Tu W, Wang S, et al. Reasoning over different types of Knowledge Graphs: Static, temporal and multi-modal. *arXiv* [Preprint]. 2022. Available from: doi: 10.48550/arXiv.2212.05767.
[72] Trivedi R, Dai H, Wang Y, Song L. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In: International conference on machine learning. In: *Proceedings of the 34th International Conference on Machine*

Learning. Sydney, NSW, Australia: JMLR.org; 2017. p. 3462-3471.
[73] Cai B, Xiang Y, Gao L, Zhang H, Li Y, Li J. Temporal knowledge graph completion: a survey. *arXiv* [Preprint]. 2022. Available from: doi: 10.48550/arXiv.2201.08236.
[74] Goel R, Kazemi SM, Brubaker M, Poupart P. Diachronic embedding for temporal knowledge graph completion. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2020. p. 3988-3995. Available from: doi: 10.1609/aaai.v34i04.5815.
[75] Tang X, Yuan R, Li Q, Wang TY, Yang HZ, Cai YD, et al. Timespan-aware dynamic knowledge graph embedding by incorporating temporal evolution. *IEEE Access*. 2020; 8: 6849-6860.
[76] Das R, Munkhdalai T, Yuan X, Trischler A, McCallum A. Building dynamic knowledge graphs from text using machine reading comprehension. *arXiv* [Preprint]. 2018. Available from: doi: 10.48550/arXiv.1810.05682.
[77] García-Durán A, Dumančić S, Niepert M. Learning sequence encoders for temporal knowledge graph completion. *arXiv* [Preprint]. 2018. Available from: doi: 10.48550/arXiv.1809.03202.
[78] Nadgeri A, Bastos A, Singh K, Mulang IO, Hoffart J, Shekarpour S, et al. Kgpool: Dynamic knowledge graph context selection for relation extraction. *arXiv* [Preprint]. 2021. Available from: doi: 10.48550/arXiv.2106.00459.
[79] Shapiro SC. *A net structure for semantic information storage, deduction and retrieval*. University of Wisconsin-Madison, Department of Computer Sciences. 1971.
[80] Lin J, Zhao Y, Huang W, Liu C, Pu H. Domain knowledge graph-based research progress of knowledge representation. *Neural Computing and Applications*. 2021; 33: 681-690.
[81] Abu-Salih B. Domain-specific knowledge graphs: A survey. *Journal of Network and Computer Applications*. 2021; 185: 103076.
[82] Yu H, Li H, Mao D, Cai Q. A relationship extraction method for domain knowledge graph construction. *World Wide Web*. 2020; 23: 735-753.
[83] Carlson A, Betteridge J, Kisiel B, Settles B, Hruschka E, Mitchell T. Toward an architecture for never-ending language learning. *AAAI*. 2010; 24(1): 1306-1313. Available from: doi: doi.org/10.1609/aaai.v24i1.7519.
[84] Mitchell T, Fredkin E. Never-ending language learning. In: *2014 IEEE International Conference on Big Data (Big Data)*. Washington, DC, USA: IEEE; 2014. p. 1-1. Available from: doi: 10.1109/BigData.2014.7004203.
[85] Suchanek FM, Kasneci G, Weikum G. Yago: a core of semantic knowledge. In: *Proceedings of the 16th International Conference on World Wide Web*. New York, United States: Association for Computing Machinery; 2007. p. 697-706.
[86] Kasneci G, Suchanek F, Weikum G. *Yago-a Core of Semantic Knowledge*. 2006.
[87] Navigli R, Ponzetto SP. BabelNet: Building a very large multilingual semantic network. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics; 2010. p. 216-225.
[88] Navigli R, Bevilacqua M, Conia S, Montagnini D, Cecconi F. Ten years of BabelNet: A survey. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. 2021. p. 4559-4567. Available from: doi: 10.24963/ijcai.2021/620.
[89] Lenat DB, Prakash M, Shepherd M. CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI Magazine*. 1985; 6(4): 65.
[90] Lenat DB, Guha RV, Pittman K, Pratt D, Shepherd M. Cyc: toward programs with common sense. *Communications of the ACM*. 1990; 33(8): 30-49.
[91] Lenat DB. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*. 1995; 38(11): 33-38.
[92] Matuszek C, Witbrock M, Cabral J, DeOliveira J. An introduction to the syntax and content of cyc. In: *2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*. Stanford, CA; 2006.
[93] Bizer C, Lehmann J, Kobilarov G, Auer S, Becker C, Cyganiak R, et al. Dbpedia-a crystallization point for the web of data. *Journal of Web Semantics*. 2009; 7(3): 154-165.
[94] Fensel D, Şimşek U, Angele K, Huaman E, Kärle E, Panasiuk O, et al. Introduction: what is a knowledge graph? In: *Knowledge graphs: Methodology, Tools and Selected Use Aases*. Springer, Cham; 2020. p. 1-10.
[95] Steiner T, Verborgh R, Troncy R, Gabarro J, Van de Walle R. Adding realtime coverage to the google knowledge graph. In: *11th International Semantic Web Conference (ISWC 2012)*. Boston, USA: CEUR-WS.org; 2012. p. 65-68.
[96] Fallatah O, Zhang Z, Hopfgartner F. A gold standard dataset for large knowledge graphs matching. In: *15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference*

(ISWC 2020). 2020. p. 24-35.
[97] Kejriwal M. *Domain-Specific Knowledge Graph Construction*. Springer; 2019.
[98] Xu J, Kim S, Song M, Jeong M, Kim D, Kang J, et al. Building a PubMed knowledge graph. *Scientific Data*. 2020; 7(1): 205.
[99] Li L, Wang P, Yan J, Wang Y, Li S, Jiang J, et al. Real-world data medical knowledge graph: construction and applications. *Artificial Intelligence in Medicine*. 2020; 103: 101817.
[100]Huang X, Zhang J, Xu Z, Ou L, Tong J. A knowledge graph based question answering method for medical domain. *PeerJ Computer Science*. 2021; 7: e667.
[101]Lei Q, Liu Y. Constructing movie domain Knowledge Graph based on LOD. In: *2019 Twelfth International Conference on Ubi-Media Computing (Ubi-Media)*. Bali, Indonesia: IEEE; 2019. p. 54-57.
[102]Chen P, Lu Y, Zheng VW, Chen X, Yang B. Knowedu: A system to construct knowledge graph for education. *IEEE Access*. 2018; 6: 31553-31563.
[103]Haussmann S, Seneviratne O, Chen Y, Ne'eman Y, Codella J, Chen CH, et al. FoodKG: a semantics-driven knowledge graph for food recommendation. In: *The Semantic Web-ISWC 2019: 18th International Semantic Web Conference*. Auckland, New Zealand: Springer; 2019. p. 146-162.
[104]Wang S, Wang D, Zhang D, Zhang F. Construction and querying of ancient poet Knowledge Graph. *Journal of Physics: Conference Series*. 2019; 1237: 032047.
[105]Abu-Salih B, Al-Tawil M, Aljarah I, Faris H, Wongthongtham P, Chan KY, et al. Relational learning analysis of social politics using knowledge graph embedding. *Data Mining and Knowledge Discovery*. 2021; 35(4): 1497-1536.
[106]Wu T, Gao C, Qi G, Zhang L, Dong C, Liu H, et al. KG-Buddhism: The Chinese Knowledge Graph on Buddhism. In: *Semantic Technology: 7th Joint International Conference, JIST 2017*. Gold Coast, QLD, Australia; Springer; 2017. p. 259-267.
[107]Lan J, Shi R, Cao Y, Lv J. Knowledge Graph-based conversational recommender system in travel. In: 2*022 International Joint Conference on Neural Networks (IJCNN)*. Padua, Italy: IEEE; 2022. p. 1-8.
[108]Auer S, Kovtun V, Prinz M, Kasprzik A, Stocker M, Vidal ME. Towards a knowledge graph for science. In: *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*. New York, NY, United States: Association for Computing Machinery; 2018. p. 1-6.
[109]Norabid IA, Fauzi F. Rule-based text extraction for multimodal Knowledge Graph. *International Journal of Advanced Computer Science and Applications*. 2022; 13(5). Available from: doi: 10.14569/IJACSA.2022.0130535.
[110]Bakhshi M, Nematbakhsh M, Mohsenzadeh M, Rahmani AM. SParseQA: Sequential word reordering and parsing for answering complex natural language questions over knowledge graphs. *Knowledge-Based Systems*. 2022; 235: 107626.
[111]Wilcke W, de Boer V, de Kleijn M, van Harmelen F, Scholten HJ. User-centric pattern mining on knowledge graphs: An archaeological case study. *Journal of Web Semantics*. 2019; 59: 100486.
[112]Song T, Luo J, Huang L. Rot-pro: Modeling transitivity by projection in knowledge graph embedding. *Advances in Neural Information Processing Systems*. 2021; 34: 24695-24706.
[113]Zhou Z, Liu S, Xu G, Zhang W. On completing sparse knowledge base with transitive relation embedding. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI; 2019. p. 3125-3132.
[114]Percebois C, Strecker M, Tran HN. Rule-level verification of graph transformations for invariants based on edges' transitive closure. In: *Software Engineering and Formal Methods: 11th International Conference, SEFM 2013*. Madrid, Spain: Springer; 2013. p. 106-121.
[115]Krötzsch M, Thost V. Ontologies for knowledge graphs: Breaking the rules. In: *The Semantic Web-ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016*. Japan: Springer; 2016. p. 376-392.
[116]Wang X, Wang S, Xin Y, Yang Y, Li J, Wang X. Distributed pregel-based provenance-aware regular path query processing on RDF knowledge graphs. *World Wide Web*. 2020; 23: 1465-1496.
[117]Zhang Q, Wang R, Yang J, Xue L. Knowledge graph embedding by translating in time domain space for link prediction. *Knowledge-Based Systems*. 2021; 212: 106564.
[118]Lewis RM, Torczon V. *Rank Ordering and Positive Bases in Pattern Search Algorithms*. Institute For Computer Applications in Science And Engineering Hampton VA; 1996.
[119]Wang Z, Fan B, Wu F. Local intensity order pattern for feature description. In: *2011 International Conference on computer vision*. Barcelona: IEEE; 2011. p. 603-610.
[120]Kim AY, Song HJ, Park SB, Lee SJ. A re-ranking model for dependency parsing with knowledge graph

embeddings. In: *2015 International Conference on Asian Language Processing (IALP)*. Suzhou, China: IEEE; 2015. p. 177-180.
[121]Zhao X, Xing Z, Kabir MA, Sawada N, Li J, Lin SW. Hdskg: Harvesting domain specific knowledge graph from content of webpages. In: *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (Saner)*. Klagenfurt, Austria: IEEE; 2017. p. 56-67.
[122]Zhang H, Liu X, Pan H, Song Y, Leung CWK. ASER: A large-scale eventuality knowledge graph. In: *Proceedings of the Web Conference 2020*. New York, NY, USA: Association for Computing Machinery; 2020. p. 201-211. Available from: doi: 10.1145/3366423.3380107.
[123]Wang YS, Liu C, Zeng X, Yuille A. Scene graph parsing as dependency parsing. *arXiv* [Preprint]. 2018. Available from: doi: 10.48550/arXiv.1803.09189.
[124]Nivre J. Dependency parsing. *Language and Linguistics Compass*. 2010; 4(3): 138-152.
[125]Tapanainen P, Jarvinen T. A non-projective dependency parser. In: *Fifth Conference on Applied Natural Language Processing*. Washington, DC, USA: The Association for Computational Linguistics; 1997. p. 64-71.
[126]Nivre J, Hall J, Nilsson J, Chanev A, Eryigit G, Kübler S, et al. MaltParser: A language-independent system for data- driven dependency parsing. *Natural Language Engineering*. 2007; 13(2): 95-135.
[127]Naseem T, Barzilay R, Globerson A. Selective sharing for multilingual dependency parsing. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Jeju, Republic of Korea: Association for Computational Linguistics; 2012. p. 629-637.
[128]Nayak A, Kesri V, Dubey RK. Knowledge graph based automated generation of test cases in software engineering. In: *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*. New York, NY, United States: Association for Computing Machinery; 2020. p. 289-295.
[129]Hu Z, Wang Z, Wang Y, Tan AH. Aspect sentiment triplet extraction incorporating syntactic constituency parsing tree and commonsense Knowledge Graph. *Cognitive Computation*. 2023; 15(1): 337-347.
[130]Nicholson DN, Greene CS. Constructing knowledge graphs and their biomedical applications. *Computational and Structural Biotechnology Journal*. 2020; 18: 1414-1428.
[131]Chao K, Tao L, Li M, Guoyu J, Yu Z, et al. Construction and application research of knowledge graph in spacecraft launch. *Journal of Physics: Conference Series*. 2021; 1754(1): 012180.
[132]Dessì D, Osborne F, Recupero DR, Buscaldi D, Motta E. Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain. *Future Generation Computer Systems*. 2021; 116: 253-264.
[133]Ko H, Witherell P, Lu Y, Kim S, Rosen DW. Machine learning and knowledge graph based design rule construction for additive manufacturing. *Additive Manufacturing*. 2021; 37: 101620.
[134]Nickel M, Murphy K, Tresp V, Gabrilovich E. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*. 2015; 104(1): 11-33.
[135]Xia F, Sun K, Yu S, Aziz A, Wan L, Pan S, et al. Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*. 2021; 2(2): 109-127.
[136]Castellano G, Digeno V, Sansaro G, Vessio G. Leveraging knowledge graphs and deep learning for automatic art analysis. *Knowledge-Based Systems*. 2022; 248: 108859.
[137]Chen X, Jia S, Xiang Y. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*. 2020; 141: 112948.
[138]Choudhary S, Luthra T, Mittal A, Singh R. A survey of knowledge graph embedding and their applications. *arXiv* [Preprint]. 2021. Available from: doi: 10.48550/arXiv.2107.07842.
[139]Ye H, Zhang N, Chen H, Chen H. Generative knowledge graph construction: A review. *arXiv* [Preprint]. 2022. Available from: doi: 10.48550/arXiv.2210.12714.
[140]Pan S, Luo L, Wang Y, Chen C, Wang J, Wu X. Unifying large language models and Knowledge Graphs: A roadmap. *arXiv* [Preprint]. 2023. Available from: doi: 10.48550/arXiv.2306.08302.
[141]Wang X, Chen L, Ban T, Usman M, Guan Y, Liu S, et al. Knowledge graph quality control: A survey. *Fundamental Research*. 2021; 1(5): 607-626.
[142]Ge C, Gao Y, Weng H, Zhang C, Miao X, Zheng B. Kgclean: An embedding powered knowledge graph cleaning framework. *arXiv* [Preprint]. 2020. Available from: doi: 10.48550/arXiv.2004.14478.
[143]Wit M. *CARD: Credibility assessment model for the Google Knowledge Graph Card*. University of Twente. 2019.
[144]Liu Y. *Error Detection in Knowledge Graphs*. PhD thesis. 2021.
[145]Rahoman M, Ichise R. Automatic erroneous data detection over type-annotated linked data. *IEICE Transactions on Information and Systems*. 2016; 99(4): 969-978.

[146]Sahay T, Mehta A, Jadon S. Schema matching using machine learning. In: 2*020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*. Noida, India: IEEE; 2020. p. 359-366.
[147]Peeters R, Bizer C. Using ChatGPT for entity matching. *arXiv* [Preprint]. 2023. Available from: doi: 10.48550/arXiv.2305.03423.
[148]Yang J, Wang D, Zhou W, Qian W, Wang X, Han J, et al. Entity and relation matching consensus for entity alignment. In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. New York, NY, USA: Association for Computing Machinery; 2021. p. 2331-2341.