

Review

An Overview of Knowledge Graph Reasoning: Key Technologies and Applications

Yonghong Chen ¹, Hao Li ², Han Li ¹, Wenhao Liu ¹, Yirui Wu ^{2,*}, Qian Huang ^{2,*}  and Shaohua Wan ³ ¹ Yangzhou Collaborative Innovation Research Institute Co., Ltd., Yangzhou 225006, China² College of Computer and Information, Hohai University, Nanjing 211100, China³ Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Chengdu 611731, China

* Correspondence: wuyirui@hhu.edu.cn (Y.W.); huangqian@hhu.edu.cn (Q.H.)

Abstract: In recent years, with the rapid development of Internet technology and applications, the scale of Internet data has exploded, which contains a significant amount of valuable knowledge. The best methods for the organization, expression, calculation, and deep analysis of this knowledge have attracted a great deal of attention. The knowledge graph has emerged as a rich and intuitive way to express knowledge. Knowledge reasoning based on knowledge graphs is one of the current research hot spots in knowledge graphs and has played an important role in wireless communication networks, intelligent question answering, and other applications. Knowledge graph-oriented knowledge reasoning aims to deduce new knowledge or identify wrong knowledge from existing knowledge. Different from traditional knowledge reasoning, knowledge reasoning methods oriented to knowledge graphs are more diversified due to the concise, intuitive, flexible, and rich knowledge expression forms in knowledge graphs. Based on the basic concepts of knowledge graphs and knowledge graph reasoning, this paper introduces the latest research progress in knowledge graph-oriented knowledge reasoning methods in recent years. Specifically, according to different reasoning methods, knowledge graph reasoning includes rule-based reasoning, distributed representation-based reasoning, neural network-based reasoning, and mixed reasoning. These methods are summarized in detail, and the future research directions and prospects of knowledge reasoning based on knowledge graphs are discussed and prospected.

Keywords: knowledge reasoning; knowledge graph; rule reasoning; distributed representation; neural network



Citation: Chen, Y.; Li, H.; Li, H.; Liu, W.; Wu, Y.; Huang, Q.; Wan, S. An Overview of Knowledge Graph Reasoning: Key Technologies and Applications. *J. Sens. Actuator Netw.* **2022**, *11*, 78. <https://doi.org/10.3390/jsan11040078>

Academic Editor: Ugo Fiore

Received: 28 July 2022

Accepted: 15 November 2022

Published: 22 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Although artificial intelligence has made significant progress in recent years and is outperforming humans in some tasks, the goal of a machine with the intelligence of a two- or three-year-old child is still far from being accomplished. A large reason for this distance is the lack of knowledge in machines. For example, a machine sees the word “apple” and does not know some properties of an apple, such as falling from an apple tree, edible, red, etc. In order for the machine to understand the meaning behind the text, we need to model the thing (entity) that can be described, fill in its attributes, and expand its connections with other things, that is, build the prior knowledge of the machine.

A knowledge base is a database of knowledge that can store complex structured and unstructured information (e.g., knowledge about the apple mentioned above). Knowledge graphs (KG) [1–4] are a type of knowledge base based on graph structure or topology structure modeling. They are the key technology for the application of knowledge bases and have almost become a synonym for knowledge bases. The related technologies in knowledge graphs have been widely applied in many fields such as wireless communication networks, search engines, intelligent question answering, linguistic reasoning, recommendation computing, big data decision analysis, and so on.

In wireless communication networks, with the continuous development of wireless communication technology, people have put forward higher and higher requirements for bandwidth, data transmission rate, delay, reliability, and other indicators. In view of the complexity of wireless network structures and the diversity of service types, the relationship between the endogenous factors of wireless communication network data is usually complex. Therefore, the factors that influence the quality of a wireless communication network's performance and user experience is extremely varied: they can effectively clarify the relationship between indexes and data fields, aid in understanding the structure and operation mechanism of a wireless communication network, and assess network performance degradation or decreases in the quality of the user experience and accurately pinpoint the cause of these changes. The corresponding tuning scheme is given according to the specific reasons. Based on the above requirements, many wireless communication network applications [5,6] based on knowledge graphs have been proposed.

From another point of view, the wide demand for knowledge graphs in these fields also promotes the continuous improvement of knowledge graph technology. For example, considering insufficient data and difficulty in knowledge updating, Liu et al. [7] proposed a new Chinese healthcare knowledge graph based on deep learning methods which can be applied to mobile devices based on IoT (Internet of Things) and WoT (Web of Things). In order to solve the problem of insufficient data, 600 thousand pieces of data were processed to support the training of the model and a good performance was achieved. Cloud computing was also used in their practice, through IoT and WoT-based mobile devices for the convenience of doctors and patients. New data are calculated and stored in the cloud, which involves text and image computing, as well as processing and storage capabilities.

Because of the importance of knowledge graph reasoning (KGR) in reasoning systems based on knowledge and facts, the research and development of knowledge reasoning has been continuous for decades. Example KGs include open source ones such as DBpedia [8], Freebase [9], YAGO [10], and proprietary ones such as those developed by Google and Microsoft [11]. In the past few years, interest in KGR has exploded in the research community and industry due to its important role in artificial intelligence applications such as recommender systems [12–15], question answering [16–20], commonsense reasoning [21–24], personalized medicine [25], and drug discovery [26,27].

KGR aims to discover new knowledge from existing knowledge [2]. As shown in Figure 1, new knowledge can be divided into two types: new entities and new relationships. The technical areas involved in new entities are usually natural language processing or knowledge mapping technologies related to entity extraction, entity disambiguation, entity fusion, etc. The new relational technology involves relational extraction and knowledge reasoning. KGR refers to the derivation of potential or new relationships between entities through reasoning technology in the established knowledge graph and the discovery of new knowledge. In graph databases, graph theory, and other related fields, it is often called link prediction.

Approaches to KGR can be broadly classified into four main categories: *embedding-based reasoning*, *symbolic-based reasoning*, *neural network-based reasoning*, and *mixed reasoning*. The central idea of the **embedding-based reasoning** methods is the determination of a mapping function that maps a symbolic representation to a vector space for numerical representation, thus reducing the dimension disaster and capturing the implicit association between entities and relations. The key point is that it can be calculated directly and quickly. The common approaches are TransE [28] (translating embedding) series algorithms, RESCAL [29], DistMul [30], etc. They can be applied to downstream tasks such as node classification and link prediction. **Symbolic-based reasoning** methods mine and reason new knowledge or facts through the definition or learning of rules existing in knowledge. AMIE [31] and AMIE+ algorithms derived from the early ILP (inductive logic programming) [32] system emphasize the rapid and effective learning of high confidence rules from large-scale knowledge graphs through automated rule learning methods and have been applied for reasoning tasks. **Neural network-based reasoning** methods have a strong

expression ability and achieve good results in relation (link) prediction and other tasks. The design of a network structure is diverse and can meet different reasoning requirements. For example, NTN [33] uses a bilinear tensor layer to replace the standard linear neural network to directly relate two entity vectors and calculate the score of the probability of a relationship between the two entities. R-GCN [34] applied GCN to relational networks, especially for relational prediction and entity classification, and introduced the method of weight sharing and coefficient constraint so that it could be applied to networks with numerous relationships.

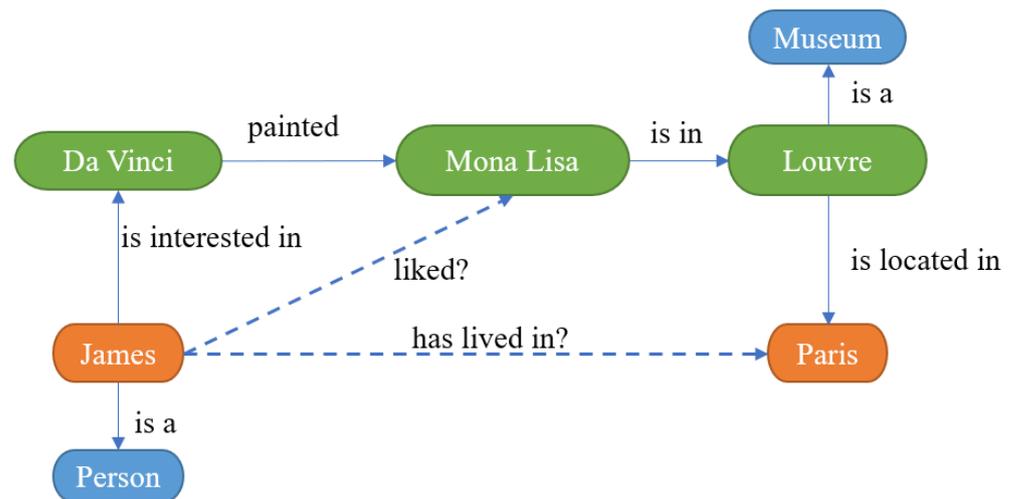


Figure 1. Example of an incomplete knowledge graph which contains missing links (dashed lines) that can possibly be inferred from existing facts (solid lines).

Mixed reasoning can combine the abilities of symbolic-based reasoning, embedding-based reasoning, and neural network-based reasoning to achieve complementary advantages and improve the accuracy and interpretability of reasoning results at the same time. For example, neural logic programming (Neural LP) [35] is a differentiable knowledge graph reasoning method, which integrates relational representation learning, rule learning, and recurrent neural network. LSTMs generate implicit variables in multi-step reasoning, and generate attention to each relationship in multi-step reasoning through implicit variables. DeepPath [36] and MINERVA [37] used reinforcement learning to learn a path selection strategy in the multi-step reasoning process of knowledge graphs. RUGE (rule-guided embedding) [38] iteratively inputs the existing inference rules into the representation learning process of knowledge graphs, and achieves better inference results by constraining and influencing the representation learning results of knowledge graphs.

In this paper, we attempt to establish the basics for future research through a thorough literature review of knowledge graph reasoning approaches. More specifically, firstly, we review the basic concepts of knowledge graphs and classify and summarize the methods related to knowledge graphs from three aspects: knowledge representation, knowledge extraction, and knowledge fusion. Then, we start from the definition of knowledge graph reasoning and explore its applications in industry. Then, we classify knowledge graph reasoning methods and introduce and compare the various kinds of knowledge graph reasoning algorithms from classical models to the latest developments. Finally, we discuss the pending problems and future directions of knowledge graph reasoning.

The rest of the paper is organized as follows: In Section 2, we detail the application of knowledge graph reasoning to different tasks. In Section 3, for the convenience of readers who are not familiar with knowledge graphs, we first quickly review the related technologies of knowledge graphs. In Section 4, we introduce four KGR technologies mentioned above in detail. Then, in Section 5, we compare and analyze various KGR technologies and introduce the application of KGR technology from the two perspectives

of technology and industry. Finally, in Section 6, we present a brief summary and the challenges and future research directions of knowledge graph reasoning.

2. Brief Introduction to Knowledge Graphs

2.1. Knowledge Representation

Typically, knowledge is derived from open data on the Internet or data within the domain or enterprise, which often leads to a generic knowledge graph or domain knowledge graph. After obtaining the data, the knowledge should be stored on the computer in a standard format. Early experts used Semantic Web and logical description to describe explicit, discrete, and thus endogenically interpretable knowledge (such as first-order predicate logic, Horne clauses, and Horne logic).

With the development of the Semantic Web and the Internet, knowledge representation has ushered in a new opportunity and challenge (Figure 2). The opportunity is that the Semantic Web provides a good application scenario for knowledge representation. The challenge is that Semantic Web-oriented knowledge representation needs to provide a set of standard language that can be used to describe various information on the Web. The early standard languages of the Web, HTML and XML, could not adapt to the requirements of the knowledge representation of the Semantic Web, so W3C proposed new standard languages: RDF, RDFS, and OWL.

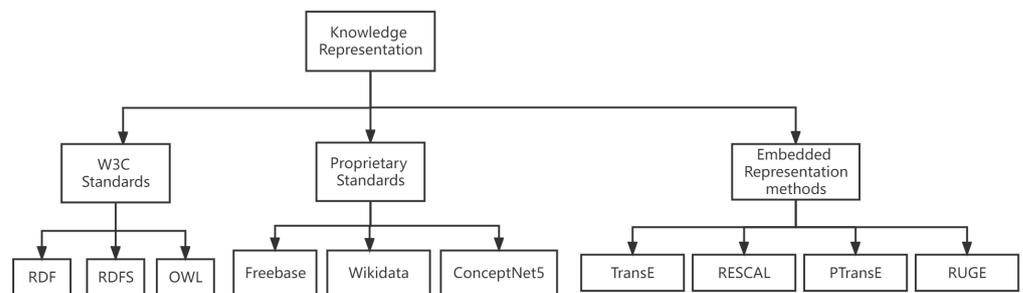
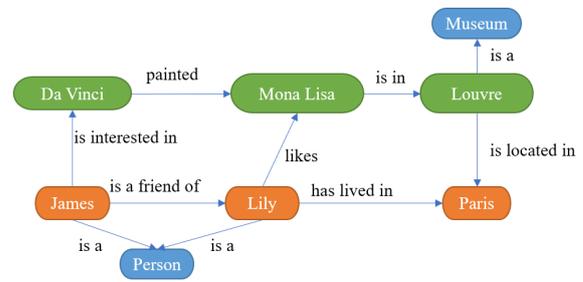


Figure 2. Categorization of knowledge representation methods.

Proprietary knowledge graphs usually have specific knowledge representation methods. For example, The Freebase [9] knowledge representation framework mainly includes the following elements: objects, facts, types, and properties. Object indicates an entity. Each object has a unique ID, called a machine ID (MID). An “Object” can have one or more “Types”. “Properties” describes “Facts”. Wikidata’s [39] knowledge representation framework mainly includes the following elements: pages, entities, items, properties, statements, qualifiers, references, etc. Furthermore, the knowledge representation framework of ConceptNet5 [40] consists of concepts, words, phrases, assertions, relations, and edges. Figure 3 shows an example of a knowledge base and knowledge graph.

- (Da Vinci, **Painted**, Mona Lisa)
- (Mona Lisa, **is in**, Louvre)
- (Louvre, **is a**, Museum)
- (Louvre, **is located in**, Paris)
- (James, **is a**, Person)
- (James, **is interested in**, Da Vinci)
- (James, **is a friend of**, Lily)
- (Lily, **is a**, Person)
- (Lily, **likes**, Mona Lisa)
- (Lily, **has lived in**, Paris)



(a) Factual triples in knowledge base.

(b) Entities and relations in knowledge graph

Figure 3. An example of knowledge base and knowledge graph.

Of the representation methods of knowledge graphs mentioned above, most of them are based on triples to organize knowledge. Although this discrete symbolic expression can

be very effective in structuring the data, these symbols cannot express the corresponding semantic level of information in the computer, nor can they carry out semantic calculation, which is not friendly to some downstream applications.

With the development of representation learning and the appearance of embedding technology such as word vectors in NLP, people have been inspired to represent knowledge as low-dimensional dense vectors similar to word vectors. By embedding the entities and relations in the knowledge graph into a low-dimensional continuous vector space, we can learn a low-dimensional vector representation for each entity and relationship. This kind of knowledge representation based on continuous vector can realize the discovery of new facts and new relations through numerical operation and can discover more implicit knowledge and potential hypotheses more effectively. This implicit knowledge is usually subjective and difficult to observe and summarize. More importantly, as a type of prior knowledge, knowledge graph embedding is usually input into many deep neural network models to constrain and supervise the training process of a neural network.

The main methods of knowledge graph embedding are as follows: (1) transfer distance methods, such as TransE [28]; (2) semantic matching methods, such as RESCAL [29]; and (3) methods considering additional information, such as PTransE [41] and RUGE [38]. In the transfer distance methods, word vectors are spatially translational. For example, TransE's design principle is that if a triplet (H, R, T) is true, then they need to conform to the $H + R \approx T$ relation, for example: $\text{Vec}(\text{Rome}) + \text{Vec}(\text{is-Capital-of}) \approx \text{Vec}(\text{Italy})$.

RESCAL [29] represents binary relational data as a three-dimensional tensor, where two dimensions are entities and the third dimension is the relation, and the element of the corresponding position in the tensor is 1 if the triplet exists, otherwise it is set to 0. The tensor value corresponding to each triplet in the knowledge graph is decomposed into the matrix representation of entity and relation, so that the decomposition entity matrix and relation matrix multiplication is as close to the original tensor value as possible.

In addition to constructing models of knowledge graph embedding based on only triples in the knowledge base, some models have been upgraded to consider additional information. For example, in path-based TransE, also known as PTransE [41], adding, multiplying, and RNN are considered to be three ways to express relational paths. Besides, Guo [38] proposed a rule-guided knowledge graph embedding method, wherein the proposed soft rule refers to the rules with confidence mined in the knowledge graph by AMIE+ rule learning method. The overall framework of the approach is an iterative process consisting of two parts, called Soft Label Prediction and Embedding correction. To put it simply, rule learning and knowledge graph embedding learning iterate each other and finally make knowledge graph embedding integrate certain rule information.

2.2. Knowledge Extraction

The data sources of knowledge extraction can be structured data (e.g., relational databases, spreadsheets, and CSV files), semi-structured data (e.g., HTML and XML) or unstructured data (e.g., RDF and linked data). For different types of data sources, the key technologies involved in knowledge extraction and the technical difficulties to be solved are different.

Specifically, as shown in Figure 4, knowledge extraction includes the following sub-tasks:

1. Named entity recognition : the detection of named entities from text and classification of them into predefined categories, such as person, organization, place, time, etc. In general, named entity recognition is the basis of other knowledge extraction tasks.
2. Relationship extraction: the identification and extraction of entities and relationships between entities from the text. For example, from the sentence "[Steve Jobs] is one of the founders of [Apple]", the entities "[Steve Jobs]" and "[Apple]" are identified as having a "is-founder-of" relationship.
3. Event extraction: the identification of the information about the event in the text and presentation of it in a structured form. For example, information such as location, time, target, and victim can be identified from news reports of terrorist attacks.

In addition, some work has used methods such as multitasking learning to extract entities and relationships together [42]. Some recent studies have used reinforcement learning to reduce manual labeling and automatically reduce noise [43]. A large amount of data exists in the form of unstructured data (i.e., free text), such as news reports, scientific literature, and government documents, etc. Below, we introduce the main technologies and methods of knowledge extraction oriented for text data.

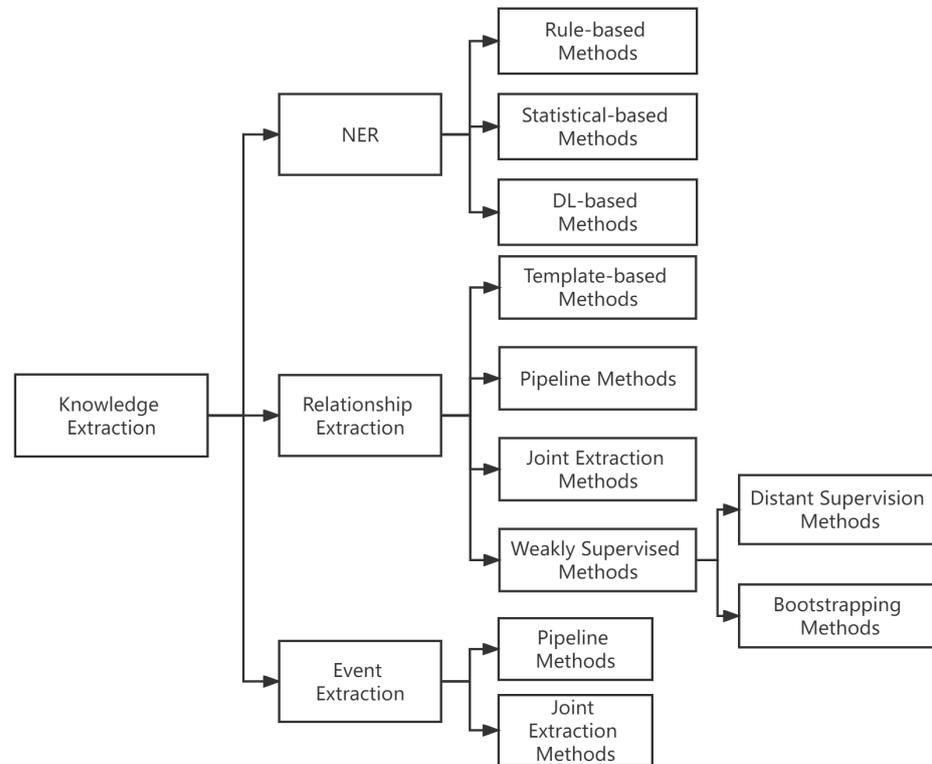


Figure 4. Categorization of knowledge extraction methods.

2.2.1. Named Entity Recognition

Named entity recognition, also known as entity extraction, aims to extract entity information elements from text. In order to extract entities from text, it is necessary to identify and locate the entities from the text, and then classify the identified entities into predefined categories.

The research on entity extraction was carried out relatively early, and a large number of methods have been accumulated in this field. In general, existing methods can be divided into rule-based methods, statistical model-based methods, and deep learning-based methods.

(1) Rule-based methods:

Rule-based methods are early adoptions of manual rules, such as regular matching. They are usually manually constructed by experts with certain domain knowledge. The rules are then matched against the text string to identify a named entity. This entity extraction method can achieve high accuracy and recall rate on small data sets, but with an increase in data sets, the construction cycle of rule sets becomes longer and the portability is poor.

(2) Statistical-based methods:

Statistical model-based methods require fully annotated or partially annotated corpus for model training. The main models adopted include hidden Markov models [44] (conditional Markov model and maximum entropy model) and conditional random fields. This method treats named entity recognition as a sequence annotation problem. Compared with ordinary classification problems, the prediction of the current tag in the sequence labeling problem is not only related to the current input feature, but also related to the previous

prediction tag, that is, the prediction tag sequence is strongly interdependent. Recognizing entities from natural text is a typical sequence annotation problem. The construction of named entity recognition method based on statistical model mainly involves three aspects: training corpus annotation, feature definition, and model training.

In order to construct the training corpus of a statistical model, people usually use an inside–outside–beginning (IOB) or Inside–outside (IO) annotation system to annotate texts manually. In the IOB annotation system, each word in the text is marked as the initial word (B) of the real name, the subsequent word (I) of the entity name, or the external word (D) of the entity name. In the IO annotation system, words in text are marked as words inside the entity name (I) or words outside the entity name (D).

Before training the model, the statistical model needs to calculate a set of features for each word input to the model. These features include word-level features, dictionary features, and document-level features. Word-level features include whether the initial letter is capitalized, whether it ends with a period, whether it contains a number or a part of speech, the n-gram of the word, etc. Dictionary characteristics depend on external dictionary definitions, such as predefined word lists, lists of place names, and so on. Document-level features are calculated based on the whole corpus document set, such as word frequency and co-occurrence words in the document set.

(3) Deep learning-based methods:

With the wide application of deep learning in the field of natural language processing, deep neural networks have been successfully applied to named entity recognition and have achieved good results. Compared with traditional statistical models, the method based on deep learning directly takes the vector of words in the text as input and realizes end-to-end named entity recognition through neural networks, no longer relying on manually defined features.

At present, convolutional neural networks (CNN), recurrent neural networks (RNN), and attentional mechanisms are the main neural networks used in named entity recognition. Generally, different neural network structures play the role of encoders in named entity recognition. Based on the initial input and the context information of the word, they obtain the new direction table of each word. Finally, the annotation result of each word is output through a CRF model.

BIO labeling is the main method used in named entity recognition to represent the result of sequence labeling. There are many labeling methods in sequence labeling, the most important of which are BIO and BIOES. B is the beginning of an entity word, I is the middle of an entity word, E is the end of an entity word, and S indicates that the entity word contains only the current word.

In addition, if in some scenarios entity categories (such as question answering systems) are not considered, the process is completed. However, in many scenarios, entity categories need to be considered (such as extracting subjects, objects, places, institutions, and so on in event extraction), and the BIO's tag list must be extended. A B and I label is assigned to each "entity type", such as "B-brand", to represent "the beginning of the entity word and the entity type is brand". When there are too many entity categories, the size of the BIOES tag list can rapidly increase.

In 2015, Huang et al. [45] proposed a Bi-LSTM + CRF model structure. As shown in Figure 5, in Bi-LSTMs (bi-directional LSTM) there are two LSTM cells, one of which runs left to right to obtain the first layer representation vector, and the other runs right to left to obtain the second layer vector. Then, the two layers of vectors are added together to obtain the third layer of vector. If CRF is not used, the third layer can be directly connected with softmax to output the results. If CRF is used, the third layer of vector must be entered into the CRF layer, which determines the final result after a professional and rigorous calculation by CRF.

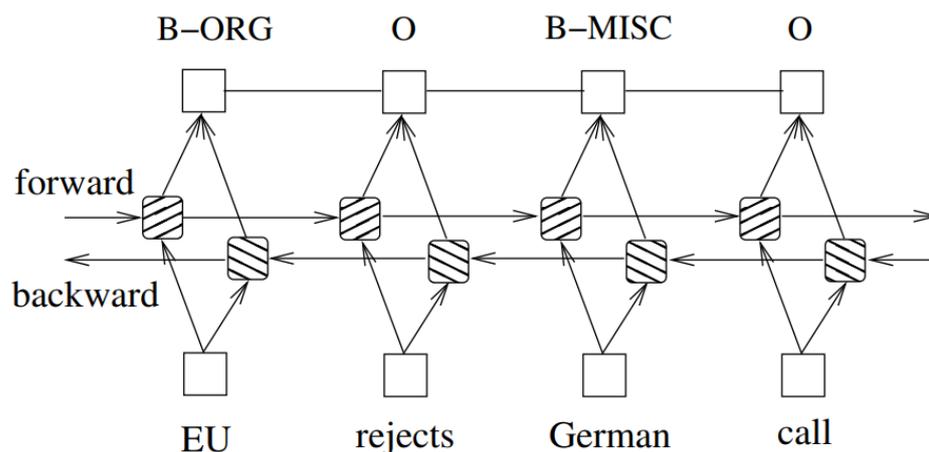


Figure 5. A bidirectional LSTM network for named entity recognition.

Google's recently proposed BERT [46] model takes advantage of Transformer's multi-layer self-attention bi-directional modeling capabilities by randomly blocking 15% of words. In various NLP downstream tasks (such as sentence pair classification tasks, single sentence classification tasks, question answering tasks), good results have been obtained. However, because BERT did not have modeling based on lexical structure and syntactic structure, it was difficult to provide a good vector representation for the new words, while ERNIE [47] greatly enhanced the general semantic representation ability through unified modeling of lexical structure, grammatical structure, and semantic information in the training data. In a number of tasks, the results have greatly exceeded BERT.

2.2.2. Relationship Extraction

Relationship extraction is one of the important sub-tasks of knowledge extraction. It is oriented toward unstructured text data and extracts semantic relationships between two or more entities from text. Relationship extraction is closely related to entity extraction. In general, after identifying the entities in the text, the possible relationships between the entities are extracted. There are also many joint models that do these two tasks together.

Here are the main methods of existing relationship extraction:

(1) Template-based relationship extraction methods:

Using the above template for matching in the text, a new entity with a "husband and wife" relationship can be obtained. The advantage of the template-based relationship extraction method is that the template is simple to construct and the relationship extraction system can be implemented quickly on small-scale data sets. Similarly, building templates by hand can take a lot of time for domain experts when the amount of data is large. In addition, the template-based relationship extraction system has poor portability, and the template needs to be rebuilt when facing a relationship extraction problem in another domain. Finally, due to the limited number of manually constructed templates and the insufficient coverage of templates, the recall rate of template-based relational extraction system is generally not high.

(2) Pipelined methods:

At present, the existing relational extraction methods based on supervised learning mainly include the pipelined method and the joint extraction method [48].

The pipelined methods treat entity identification and relation extraction as two separate processes, and they do not affect each other. However, relationship extraction is carried out on the basis of entity extraction results, so the results of relationship extraction also depend on the results of entity extraction, such as MTB [49] and BR-CNN [50]. When inputting a sentence, the named entity is first recognized, the identified entities are then paired together, the relationship is classified, and finally the triplet with the entity relationship is taken as the input. The disadvantages of the assembly line approach are as follows: (1) Error propagation: the error of entity recognition module will affect the

subsequent relationship classification performance; (2) The relationship between the two subtasks is ignored. For example, if there is a country–president relationship in the figure, we can know that the former entity must belong to the location type and the latter to the person type, and the pipeline-based method cannot make use of such information; (3) Unnecessary redundant information is generated. Due to pin–pair pairing of the identified entities and then relationship classification, those unrelated entity pairs will bring redundant information and increase the error rate.

(3) Joint extraction methods:

In joint extraction methods, entity extraction and relation extraction are combined, a sentence is input, and the related entity triplet can be obtained directly through the joint model of entity recognition and relation extraction. This method can overcome the shortcomings of the above pipeline method, but it may have more complex structures, such as BERT [46], LSTM-RNN [51], and DGCNN [52].

(4) Methods based on weakly supervised learning:

The relationship extraction method based on supervised learning needs a large amount of training corpus, especially the method based on deep learning, so the optimization of the model depends more on a large amount of training data. When the training corpus is insufficient, a weakly supervised learning method can only use a small amount of annotated data for model learning. Relationship extraction methods based on weakly supervised learning mainly include **distant supervision methods** and **bootstrapping methods**.

Distant supervision methods can automatically construct a large amount of training data by aligning the knowledge graph with unstructured text, reduce the dependence of the model on manual annotation data, and enhance the cross-domain adaptability of the model. The basic assumption of distant supervision methods is that if there is a relationship between two entities in the knowledge graph, the sentence containing both entities expresses this relationship. For example, if there is a physical relationship named founder (Steve Jobs, Apple Inc.) in a knowledge graph, the sentence “Steve Jobs is the co-founder and CEO of Apple Inc.” containing the physical relationship founder can be used as a training example.

Therefore, the general steps of the distant supervision methods are as follows: (1) The entity pairs with a target relationship are extracted from the knowledge graph. (2) Sentences containing entity pairs are extracted from unstructured texts as training examples. (3) Training supervised learning model for relation extraction occurs.

Distant supervision methods can obtain training data by using rich knowledge graph information and effectively reduce the workload of manual annotation. However, based on the assumption of remote supervision, a large amount of noise will be introduced into the training data, resulting in semantic drift. Among the subsequent improvements, PCNNs (piecewise convolutional neural networks) [53] introduced the method of multi-example learning to improve remote supervision. Here, multi-example learning refers to selecting the relationship of the sentence with the highest credibility from BAG (multiple sentences of the same entity pair) as the relationship of all sentences in BAG. CNN-RL based on reinforcement learning was proposed in 2018 for noise processing [43]. The author modeled the noisy problems in the remote supervision RE as RL problems and can classify them at the sentence level, in contrast to the previous model which is at the bag level.

Bootstrapping methods originate from the autonomous sampling method in statistics. They use a small number of instances as the initial seed set, then learn the template extracted by the relationship from the seed set, and then extract more instances using the template and add them to the seed set. By iterating over time, the bootstrapping method can extract a large number of relationship instances from the text. There are many entity relationship extraction systems that use bootstrapping methods, such as Snowball [54] and NELL [55], but they are older. The advantages of the bootstrapping method are that the relationship extraction system has a low construction cost, is suitable for large-scale relationship extraction tasks, and has the ability to discover new relationships.

However, due to its sensitivity to initial seeds, semantic drift, and low accuracy of results, bootstrapping has gradually reduced in popularity.

2.2.3. Event Extraction

Event extraction is the presentation of unstructured text containing event information in a structured form, which is widely used in automatic abstracting, automatic question answering, information retrieval, and other fields. In recent years, event extraction has been attracting the attention of many research institutions and researchers. The Message Understanding Conference (MUC) and the Automatic Content Extraction Conference (ACE) are typical measurement conferences that contain extraction task events.

An event in an ACE definition consists of an event trigger and an argument that describes the event structure. Trigger is a word that can trigger the occurrence of an event. It is the most important characteristic word that determines the event type and determines the event category/sub-category. The element is used to populate the event template, which completely describes the event itself. For example, the sentence in Figure 6 contains two events: the first event is “die”, and the trigger word is “died”. Its argument contains: victim—cameraman, place—Baghdad, and instrument—American tank. The second event is an attack (fired) and its argument includes: target—Palestine Hotel, place—Baghdad, target—cameraman, and attacker—American tank.

Event extraction tasks can be composed of the following two steps: (1) event detection: ACE2005 defines eight event categories and 33 sub-categories, each of which corresponds to a unique event template. (2) Argument detection: The event element is the participant of the event. According to the event template, extract the corresponding element and label it with the correct element.

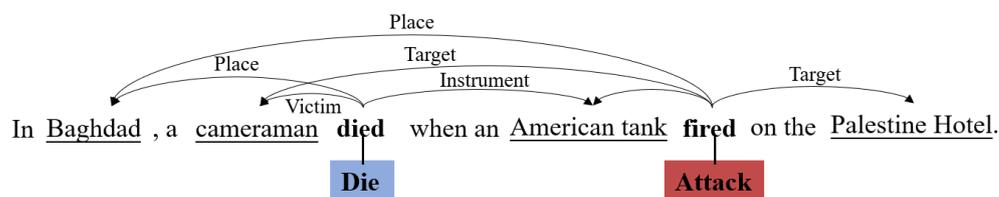


Figure 6. A sample of event extraction.

2.3. Knowledge Fusion

Knowledge fusion refers to the fusion of knowledge extracted from multiple data sources to form a huge knowledge base. Knowledge fusion includes two parts: entity link and knowledge merge. **Entity link** refers to the operation of linking the entity object extracted from text to the corresponding entity object in the knowledge base. The basic idea is the selection of a group of candidate entity objects from the knowledge base according to the given entity reference items, and then the connection of the reference necklace to the correct entity object through similarity calculation. **Knowledge merger** is mainly aimed at structured data, which can be divided into two ideas: one idea is the merger of an external knowledge base, which mainly deals with the conflict between the data layer and pattern layer, and the second idea consists of merging relational databases, such as the RDB2RDF approach.

Wu et al. [56] use the two-stage embedding method to realize entity linking. The main idea is that embedding is realized by embedding the query where the entity is located, stitching the title and description of all entities in the knowledge graph, and then computing their similarity and sorting to realize entity alignment. The overall approach is similar to the recommendation system sorting pattern.

In order to realize knowledge updating and fusion of dynamic data sources, Li et al. [57] from Tsinghua University proposed a semi-supervised entity alignment method through a knowledge embedding model and crossover graph model, which successfully integrated complementary knowledge graphs from different sources or languages. Zeng et al. [58] described the entity alignment problem as a classic stable matching problem, which realized

robust knowledge fusion by mining strong dependencies between entities. Zhu et al. [59] realized entity alignment through joint knowledge embedding, encoded entities, and relations between different knowledge graphs into a unified low-dimensional semantic space and improved alignment performance through iteration and parameter sharing.

3. Knowledge Graph Reasoning

This section begins with an introduction to knowledge graph reasoning, describing the problems it solves and its effectiveness in some knowledge graph applications. After that, various knowledge graph reasoning techniques are introduced.

3.1. Introduction

Reasoning is a form of human logical thinking; machines that have a reasoning ability similar to humans have always been the goal of the development of artificial intelligence. Symbolic reasoning and expert systems were early attempts. After the knowledge graph was proposed and developed, reasoning technology based on knowledge graphs was also developed and became one of the most popular fields in artificial intelligence, and is also considered to be a key technology for giving artificial intelligence the same level of reasoning and decision-making abilities as human beings.

Knowledge graph reasoning aims to discover new knowledge from existing knowledge. For knowledge graphs, new knowledge can be divided into two types: new entities and new relationships. The technical areas involved in new entities are usually natural language processing or knowledge mapping technologies related to entity extraction, entity disambiguation, entity fusion, etc. The new relational technology involves relational extraction and knowledge reasoning. Knowledge graph reasoning, or knowledge reasoning, refers to the derivation of potential or new relationships between entities through reasoning technology in the established knowledge graph and the discovery of new knowledge. In graph databases, graph theory, and other related fields, it is often called link prediction.

In knowledge graphs, there generally exists knowledge deficiency, that is, the incompleteness of the knowledge graph. **Knowledge graph completion** is the most widely used field of knowledge reasoning. A large number of knowledge graph reasoning algorithms have been proposed for application to knowledge graph completion, such as TransR [60], CapsE [61], RGhat [62], etc. All the methods mentioned above can determine whether there is a certain relationship between any entities through reasoning in vector space and then realize the completion of knowledge graph.

In knowledge graph reasoning, the knowledge graph itself provides a summary of human knowledge and experience, while the reasoning technology realizes the discovery of potential and unknown knowledge based on the existing knowledge in the knowledge graph, which greatly expands the abilities such as knowledge question and answer, personalized search, and intelligent recommendation. At the same time, in industry applications, domain knowledge graphs and reasoning technology have been combined to realize auxiliary analysis and decision support. In Section 5, we detail the application of knowledge graph reasoning in these tasks.

3.2. Methods of Knowledge Graph Reasoning

Approaches to KGR can be broadly classified into four main categories: *embedding-based reasoning*, *symbolic-based reasoning*, *neural-network-based reasoning*, and *mixed reasoning*.

3.2.1. Embedding-Based Reasoning

The central idea of **embedding-based reasoning** methods is the determination of a mapping function that maps a symbolic representation to a vector space for numerical representation, thus reducing the dimension disaster and capturing the implicit association between entities and relations. The key point is that it can be calculated directly and quickly. The common approaches are TransE [28] (translating embedding) series algorithms,

RESCAL [29], DistMul [30], etc. They can be applied to downstream tasks such as node classification and link prediction.

Embedding-based reasoning methods are extended from Word2vec [63], embedding knowledge graphs into low-dimensional geometric spaces (usually Euclidean space, but also hyperbolic space, etc.) through geometric operations such as translation or rotation. Among them, translation is represented as vector addition, rotation as a Hadamard product, and embedding is a concept related to manifold in mathematics, which expresses that an instance of a mathematical structure is included in another instance through mapping.

The main idea of the translational distance model is the transformation of the rationality of the triplet in the vectorized knowledge graph into the distance between the head entity and the tail entity. The emphasis of this approach is on the design of score functions, which are often designed to make use of relations to transfer the rationality of head entities to tail entities. Inspired by word vectors, the semantic relationship between words in vector space can be extended to the relationship between head entities and tail entities in vector space in knowledge graphs. In other words, the head entities and tail entities in the knowledge graph can also be considered to be mapped into the vector space, and the relations between them can also be considered as the relations in the triplet.

By considering embedding of entities and relationships of multirelational data into low-dimensional vector spaces, Bordes et al. [28] proposed TransE, which constructs a canonical model to interpret relationships as translations operations on the low-dimensional embeddings of the entities. They claim their methods to be simple and powerful in experiments for linking predictions on knowledge graphs. As shown in Figure 7, for a specific relation (head, relation, and tail), the vector representation of the relation is interpreted as a translation vector from the vector of the head entity to the vector of the tail entity. In other words, if a certain triplet is established in a knowledge graph, its entities and relations need to be full of $head + relation \approx tail$.

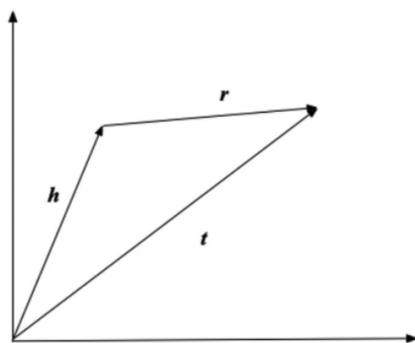


Figure 7. Simple illustration of TransE. The *h*, *r*, and *t* represent head, relationship, and tail in the triplet, respectively.

Regarding previous works such as TransE, TransH, and TransR/CTransR as rough relation translation methods from head entity to tail entity, Ji et al. [64] proposed TransD, which acts as a more fine-grained model on the basis of TransR/CTransR. Specifically, they adopt two vectors for entity and relation representation, where the first one describes the entity or relation, and the other dynamically represents the mapping matrix. Involving the construction of diversity of both relations and entities, they claim their methods to possess the advantages of less parameters and an absence of multiplication operations, which offers the possibility to directly apply TransD on large-size graph dataset.

Focusing on the task of performing link prediction between entities, i.e., knowledge graph completion, Lin et al. [60] proposed TransR to build entity and relation embeddings in separate entity spaces and relation spaces. Their ideas are derived from the fact that most former methods regard both entities and relations located within the same semantic space, and thus fail to be informative and sufficient for the effective modeling of entities and relations. Specifically, they first project entities from an entity space to the corresponding relation space. Afterwards, they build translations between projected entities. Three

separate tasks of link prediction, triple classification, and relational fact extraction prove the effectiveness of TransR model.

Starting from the point of balancing between model capacity and efficiency, Wang et al. [65] proposed TransH, which models a relation as a hyperplane together with a translation operation. Their methods take quantities of mapping properties of relations, such as one-to-many, many-to-one, and many-to-many, into consideration, thus embedding a highly efficient method, even when facing a large-scale knowledge graph. Moreover, they try to reduce false negative labels in training, which helps to move a practical knowledge graph towards completion with relatively low complexity. Experiments show that TransH achieves remarkable improvements over TransE in both accuracy and capability, testing with tasks of link prediction, triplet classification, and fact extraction on WordNet and Freebase datasets.

The core idea of RESCAL [29] is the encoding of the whole knowledge graph into a three-dimensional tensor, from which a core tensor and a factor matrix are decomposed. In the core tensor, each slice of the two-dimensional matrix represents a relationship, and each row in the factor matrix represents an entity. The result restored by the core tensor and factor matrix is regarded as the probability of the corresponding triad. If the probability is greater than a certain threshold, the corresponding triad is correct. Otherwise it is not correct.

3.2.2. Symbolic-Based Reasoning

Symbolic-based reasoning methods mainly refer to reasoning new entity relations using rules through first-order predicate logic and description logic. Typical methods include ILP [32] and AMIE [31]. The efficiency of logical reasoning on large-scale knowledge graphs is limited by its discreteness. Cohen proposed a differentiable rule reasoning machine TensorLog [66]. The main advantage of rule-based reasoning method is that the rules are usually similar to the reasoning processes that humans use to think about problems, and its reasoning conclusions can be explained. As a result, it is friendly to people. The rules deposited in the knowledge graph have good deductive ability.

ILP [32] is a symbol rule learning algorithm based on first-order logic induction and expressed in first-order logic. Entity relations in the knowledge graph can be regarded as facts described by binary predicates, so first-order logic rules can also be learned from the knowledge graph using the ILP method. Given the background knowledge and target predicates (relations in the knowledge graph), the ILP system can learn a set of logical rules describing the target predicates. FOIL is an early representative ILP system and a famous first-order rule learning algorithm. It follows the sequential coverage framework and adopts a top-down rule induction strategy.

Association rule mining under incomplete evidence (AMIE) [31] predicts the rules of each relationship by learning in sequence. For each relationship, the rule body is expanded by three operations, and the candidate (closed) rules whose support is greater than the threshold are reserved. The three operations are as follows: (1) Add hanging edges: hanging edges are where one end of the edge is a variable that does not appear, and the other end (a variable or constant) is a variable that does not appear in the rule; (2) Add instance edges: instance edges are similar to hanging edges in that one side of the edge is also a variable or constant that appears in the rule, but the other side is a constant that does not appear, which is an entity in the knowledge base; (3) Add closed edges: closed edges are edges that connect two elements (variables or constants) that already exist in the rule. Its advantage is that it can be interpreted and can automatically discover reasoning rules, but its disadvantage is that it has a large search space and a low coverage of generated rules. In addition, the prediction effect of the final model is poor.

Wang et al. [67,68] proposed a first-order probabilistic language model ProPPR (Programming with Personalized PageRank) for knowledge reasoning on knowledge graph. ProPPR constructs a directed proof graph, in which nodes correspond to the connecting or reasoning targets of clauses in the form of "relation (head entity variable, tail entity vari-

able)", where the starting node is the query clause, and edges correspond to rules, namely an reasoning step, which reduces one clause to another. Edge weights are associated with feature vectors. When introducing a feature template, edge weights can depend on the partial instantiation results of the template, such as the specific value of a variable in the clause. At the same time, a self-loop from each destination tail node to itself and a self-start edge from each node to the start node are added to the graph. The self-loop is used to increase the weight of the tail node of the target, and the self-starting edge makes the traversal tend to increase the weight of the reasoning with fewer reasoning steps. Finally, reasoning in ProPPR is implemented based on personalized Web page ratings on the graph.

Cohen [66] further proposed TensorLog, which uses differentiable processes for reasoning. In TensorLog, each entity is associated with a one-hot vector, and a 0, 1 operation matrix is defined for each relation. If there is a corresponding relation between the i th entity and the J th entity, the value at the position (i, j) is 1, otherwise it is 0. These expressions are fixed and not updated. The reasoning from this logical rule can be formalized as matrix multiplication. Given an entity and a relation, when predicting another entity, for each possible path, the entity one-hot vector is multiplied by the product of the relation operations on the path (the transpose of the product is given for tail entities), and the weighted sum of the results of all paths with the confidence parameters to be learned can obtain the corresponding score vectors of all entities. Because the one-HOT representation is used, the score of the candidate entity can be obtained by multiplying the score vector by the transpose of its one-hot vector. For confidence parameters, we learn by maximizing the score of triples in the knowledge graph.

Paulheim and Bizer [69] proposed two algorithms, SDType and SDValidate, which use the statistical distribution of attributes and types to complete type triples and identify wrong triples. SDType deduces the type of the entity through the statistical distribution of the type of the entity at the head and tail of the attribute. Similar to the weighted voting mechanism, SDType assigns a weight to the vote of each attribute. SDValidate first calculates the frequency of tail entity relations. Low-frequency triples are further scored with a statistical distribution of attributes and types. Triples scoring less than a given threshold are considered to be potentially incorrect. Jang et al. [70] proposed a model-based triplet quality assessment method for knowledge graphs. This method directly selects patterns with a high rate according to the hypothesis that more frequent patterns are more reliable for the analysis of knowledge graph data models and then uses these patterns for triplet quality analysis.

3.2.3. Neural Network-Based Reasoning

Neural network-based reasoning methods have a strong expression ability and achieve good results in relation (link) prediction and other tasks. The design of network structure is diverse and can meet different reasoning requirements. For example, NTN [33] uses a bilinear tensor layer to replace the standard linear neural network to directly relate two entity vectors and calculate the score of the probability of a relationship between the two entities. R-GCN [34] applied GCN to relational networks, especially relational prediction and entity classification, and introduced the method of weight sharing and coefficient constraint so that it could be applied to networks with numerous relationships. IRN [71] uses memory matrix and recursive neural networks as the structure of the control unit to simulate the multi-step reasoning process.

The ability to recognize that certain facts exist purely because of other existing relationships is the goal of learning models of common sense reasoning. NTN [33] (Figure 8) aims to find and certainly predict the relationship R between entities $\langle e_1, e_2 \rangle$. For example, $(e_1, R, e_2) = (\text{Bengal tiger}, \text{has part}, \text{tail})$ is a positive and deterministic relationship. Neural tensor networks (NTN) replace a standard linear neural network layer with a bilinear tensor layer, which directly relates two entity vectors on multiple dimensions and calculates the probability fraction of two entities in a specific relationship.

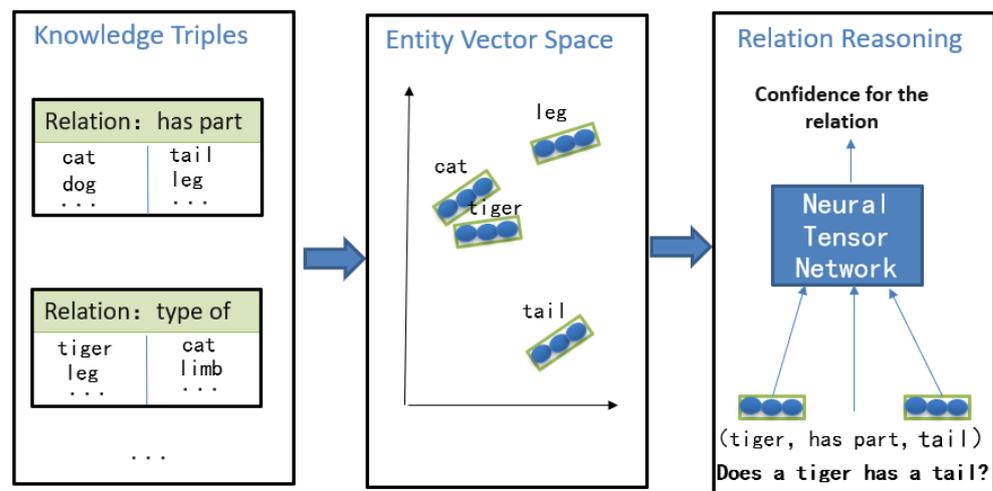


Figure 8. A simple example that uses the NTN method to predict new relationships.

R-GCN [34] is different from conventional GCNs in that it introduces a relational transformation determined by the type and direction of the edge (Figure 9), and the last term of + represents the self-connection of the node. The red part represents the entity. Matrix operation is performed with the blue neighbor node, and then the edge type of each relationship is converted to obtain the normalized result and sum of the green part. After summation, the result is transmitted through an activation function, and the node parameters of the model are updated.

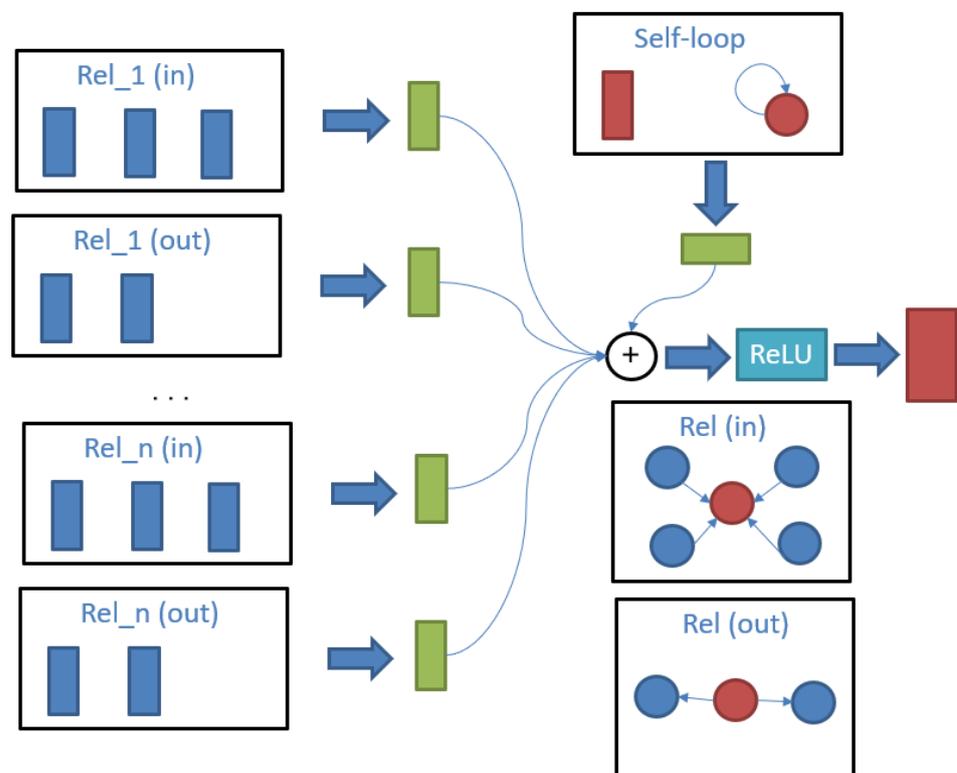


Figure 9. Diagram representing the computation of the update of a single graph node/entity (red) in the R-GCN model.

In order to imitate the ability of the human brain to store knowledge, Shen et al. proposed the IRN (Implicit ReasonNets) model [71] in 2017. In this model, a shared memory component is designed to store knowledge base information implicitly, which mimics

the storage of knowledge in the human brain. Different from commonly used reasoning methods that explicitly manipulate the observed triplet by manually designing the reasoning process, this method can implicitly learn the multi-step reasoning process by reading the shared memory module without human intervention, which simulates the process of reading memory in the human brain. In the prediction process, the model needs to form multiple intermediate representations successively. For each generated intermediate representation, an RNN controller is used to judge whether the intermediate representation has encoded enough information to generate a prediction output. If the controller allows, the current forecast is output as the final result. Otherwise, the controller takes the current intermediate representation and reads the shared memory component, merges the two information into a set of context vectors to produce a new intermediate representation, and then repeats the above judgment process until the controller allows the process to stop, at which point the prediction results are outputted. The schematic diagram of the IRN model framework is shown in Figure 10. The input module takes a query and converts the query into a vector representation q . The output module is a function f_o , which converts the hidden state received from the search controller(s) into an output O .

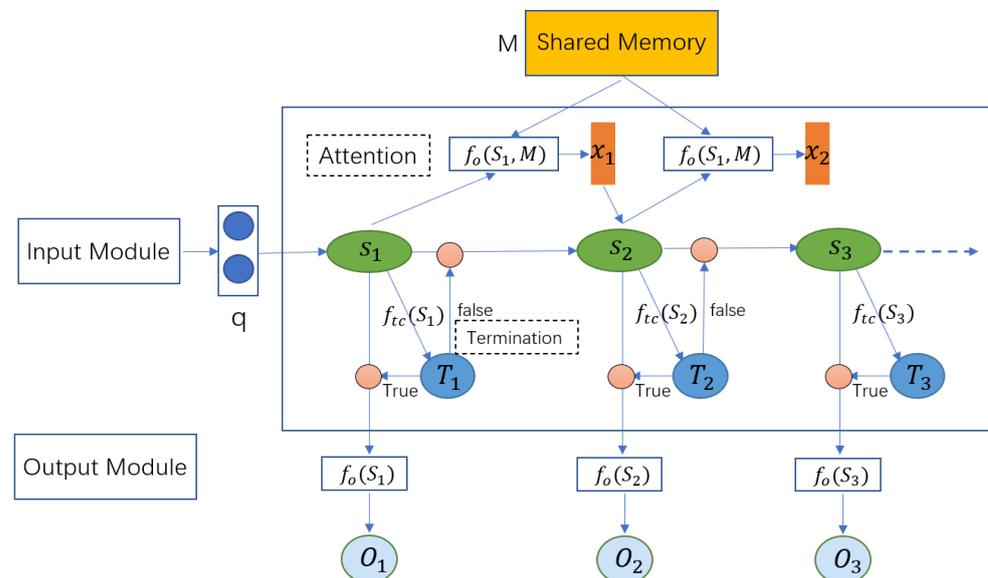


Figure 10. The structure of IRN model. The input module takes a query and converts the query into a vector representation q . The output module is a function f_o , which converts the hidden state received from the search controller(s) into an output O .

3.2.4. Mixed Reasoning

Mixed reasoning can combine the abilities of symbolic-based reasoning, embedding-based reasoning, and neural network-based reasoning to achieve complementary advantages and improve the accuracy and interpretability of reasoning results at the same time. Therefore, it has become the mainstream method of knowledge graph reasoning. For example, neural logic programming (Neural-LP) [35] is a differentiable knowledge graph reasoning method, which integrates relational representation learning, rule learning, and recurrent neural networks. LSTM generates implicit variables in multi-step reasoning, and it generates attention to each relationship in multi-step reasoning through implicit variables. DeepPath [36] and MINERVA [37] used reinforcement learning to learn a path selection strategy for the multi-step reasoning process of knowledge graphs. RUGE [38] inputs the existing reasoning rules into a knowledge graph to represent the learning process, and constraint and influence represent the learning results and are used to achieve a better reasoning effect.

In addition, in order to deal with unknown new entities outside the knowledge graph, Shi et al. [72] further defined “open-world knowledge graph completion”, which can

receive entities outside the knowledge base and link them to the knowledge graph. Based on the above ideas, the ConMask model was proposed, which is mainly divided into three parts (Figure 11): (1) Relationship-dependent content masking: text information is screened, irrelevant information is deleted, and only task-related content is left. The model adopts an attention mechanism to obtain the weight matrix of words in context and words with a given relationship based on similarity. It is found that target entities are sometimes near indicator words with high weight through observation. An MCRW weight solution method considering context was proposed. (2) Target fusion: a full convolutional neural network is used to extract target entity embedding from relevant text (FCN is the full convolutional neural network method). The partial input is a masked content matrix, where each layer has two 1D convolution operations, followed by sigmoid activation functions, followed by batch normalization and maximum pooling. The last layer of the FCN is connected to mean pooling rather than maximum pooling to ensure that the output of the target fusion layer always returns a single K-dimensional embed. (3) Target entity resolution: a similarity ranking between the candidate entity and extracted entity embedding is generated. By calculating the similarity of the candidate entity and an extracted entity embedding in KG, it is possible to obtain a ranked list combining other text features. The ranking with the highest rank is considered to be the best result. A loss function, list-wise ranking, was designed to replace the head and tail by 50% to generate negative samples to enhance model robustness.

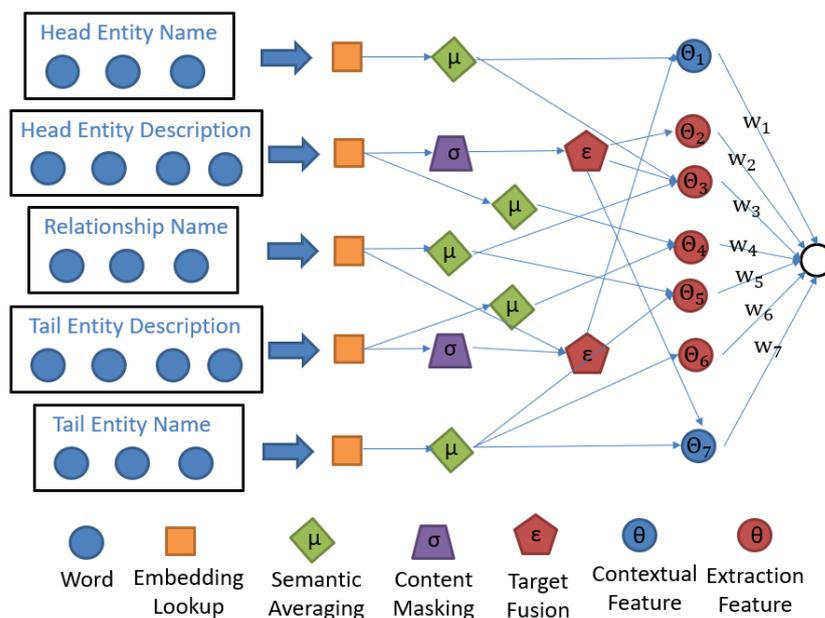


Figure 11. Illustration of the ConMask model for open-world knowledge graph completion.

4. Comparisons and Analysis

Approaches to KGR can be broadly classified into four main categories: *embedding-based reasoning*, *symbolic-based reasoning*, *neural network-based reasoning*, and *mixed reasoning*. Table 1 shows a comparison and analysis of the above methods.

In embedding-based reasoning methods, TransE series construct a canonical model to interpret relationships as translations operations on the low-dimensional embeddings of the entities. They are inspired by the translation invariance of a word vector and propose a hypothesis: if a certain triplet relation is established in a knowledge graph, then its entity and relation vector need to satisfy the relation head + relation \approx tail. These methods are simple, fast, and effective, but they are only suitable for one-to-one relationships and not one-to-many/multi-to-one relationships.

In symbolic-based reasoning methods, association rule mining under incomplete evidence (AMIE) is used, which predicts the rules of each relationship by learning in

sequence. For each relationship, the rule body is expanded by three operations, and the candidate (closed) rules whose support is greater than the threshold are reserved. AMIE and PRA provide interpretable methods and the ability to discover rules automatically. However, they often face the problem of inefficiency caused by large search space. PRA is also affected by graph sparsity.

Neural network-based reasoning and hybrid reasoning are relatively good reasoning methods and need to be further studied. Reasoning based on neural networks makes use of the powerful learning ability of neural networks for modeling reasoning. There has been some research work on simulating computer or human knowledge storage and reasoning. With the continuous development of neural networks, further research remains to be carried out. It is also difficult to further enhance the interpretability of neural networks for reasoning. Hybrid reasoning attempts to use the advantages of various reasoning methods to obtain a better reasoning performance. In general, the effect of integrated learning is better than that of a single model. However, mixed reasoning is currently limited to the mixing of two methods, and mixing of a variety of complementary methods to further improve reasoning ability needs further research. At the same time, most of the current mixing modes are relatively shallow, and more and deeper mixing modes need to be explored in the future. Among them, the hybrid rule-based reasoning method is limited to simple rules and transitive constraints, and more effective rules need to be introduced.

In recent years, modern knowledge graph reasoning technology has been developing rapidly. Many models have very good results in their own fields and are widely used, but they are far from perfect. Challenges and opportunities coexist.

- Knowledge graph embedding is usually embedded in Euclidean space. In recent years, MuRP, ATTH, and other models have explored the case of embedding in hyperbolic space and achieved very good results. However, in general, there are few studies on embedding knowledge graphs into hyperbolic space. Some models show that hyperbolic space and other non-Euclidean Spaces can better express knowledge graphs. The representation and reasoning of knowledge graph in non-Euclidean space is worth further exploration.
- Graph neural network natural matching knowledge graphs such as r-GCN and RGHAT models introduced in this paper are still early attempts and are far from perfect. The design of more sophisticated graph network structures to realize knowledge graph reasoning is a hot and promising direction.
- Transformer networks excel because of their strong expression ability and efficient parallel training ability in the field of natural language processing, and can be quickly migrated to computer vision, image processing, and speech recognition, in which the results are equally outstanding. It is believed that the converter network can also perform well in knowledge graphs and knowledge graph reasoning.
- Transfer learning based on the pre-training of models is widely used in natural language processing, image processing, and computer vision, but is rarely used in knowledge graph reasoning. It is worth exploring pre-training models in knowledge graphs and knowledge graph reasoning.
- Modern knowledge graph reasoning techniques also have great opportunities in data sets and corresponding contests and evaluations, especially in Chinese knowledge graph data sets.

Table 1. Comparison and analysis of knowledge graph reasoning methods. Bold indicates that the method is recommended.

Method	Advantages	Disadvantages
TransE series (embedding-based reasoning)	Simple, fast speed.	Only suitable for one-to-one relationships.
AMIE (symbolic-based reasoning)	Interpretable; automatic discovery of rules.	With large search space and low coverage of generated rules, the prediction effect of the final model is also poor.
NTN (neural network-based reasoning)	More resilient against the sparsity problem.	High complexity; requires a large number of triples to be fully learned.
R-GCN (neural network-based reasoning)	The graph product network is introduced into knowledge reasoning domain for the first time.	Unstable; as the number of relationships increases, the number of parameters explodes, introducing too many relationship matrices.
IRN (neural network-based reasoning)	Stores knowledge through shared memory components. Can simulate the human brain to learn multi-step reasoning process.	Has difficulty with unstructured data and self-heating language query.
ConMask (mixed reasoning)	Can add unknown new entities from outside the knowledge graph and link them to internal entity nodes.	When no text pair that can accurately describe entities or relations appears, the model cannot obtain enough reasoning basis, resulting in poor reasoning effect.

5. Applications

In this section, we detail the application of knowledge graph reasoning to different tasks, including **wireless communication networks (WCN)**, **question answering (QA) systems**, **recommendation systems**, and **personalized search**.

5.1. Wireless Communication Networks (WCN)

For the wireless communication networks (WCN), the cellular cell (also known as a cell sector) is the basic unit of wireless network coverage area division. At present, the information management and maintenance of wireless networks and wireless network equipment (that is, base station equipment) and the optimization of wireless network are mainly carried out around the base station information table. The base station information table (BSA) contains the core basic data needed by telecommunications operators for network operation and maintenance. It describes the basic parameters of all base stations and cells under a network, such as the type of base station to which the cell belongs, the longitude and latitude of the site, and the direction angle of each sector of the common station. It is an important data asset and strategic resource for operators. In the process of network optimization related to the optimization and adjustment of base station and cell parameters, it is necessary to update the base station information table in time to ensure the accuracy, integrity, and timeliness of basic data.

There are many problems and challenges [7] in the use and maintenance of the base station information table in network operation and maintenance. The base station information table is usually stored, managed, and presented in the form of a simple relational database, which needs manual summary input and update and is of a quasi-static data form. With the continuous construction, expansion, and optimization of the network, it is constantly being deployed in new residential areas and removed or relocated in old residential areas. In daily wireless optimization operations, antenna angle and dip angle are often adjusted to improve coverage. Therefore, the base station information table is dynamically changing. In addition, its information carrying capacity is limited, and it can not effectively represent the complex relationship between the base station and the cell.

The base station and the cell have strong spatial attributes, and the base station information table cannot visually express such attribute information. In addition, the existing base station information table is targeted at the network of the operator, and generally cannot obtain or represent the base station data of other operators, so it is impossible to carry out targeted network construction and network quality benchmarking. Therefore, as the upgrade and replacement of the base station information table management occurs, a more comprehensive, timely, dynamic, visual base station information storage management and presentation method has a great market demand.

Based on massive wireless network perception data, the automatic extraction of relevant information and the construction of wireless network knowledge graphs [7] is a more comprehensive, timely, dynamic, and visual method of base station information storage management and presentation, which is conducive to improve the intelligence level of mobile network operations and maintenance and improve the efficiency of operation and maintenance work. If the data source contains the network sampling data of other operators, the wireless network information can be effectively extracted and appropriately presented in the wireless network knowledge graph so as to facilitate cross-network benchmarking and network operation and maintenance based on the comparative advantage. Figure 12 shows the application architecture of a wireless communication network knowledge graph.

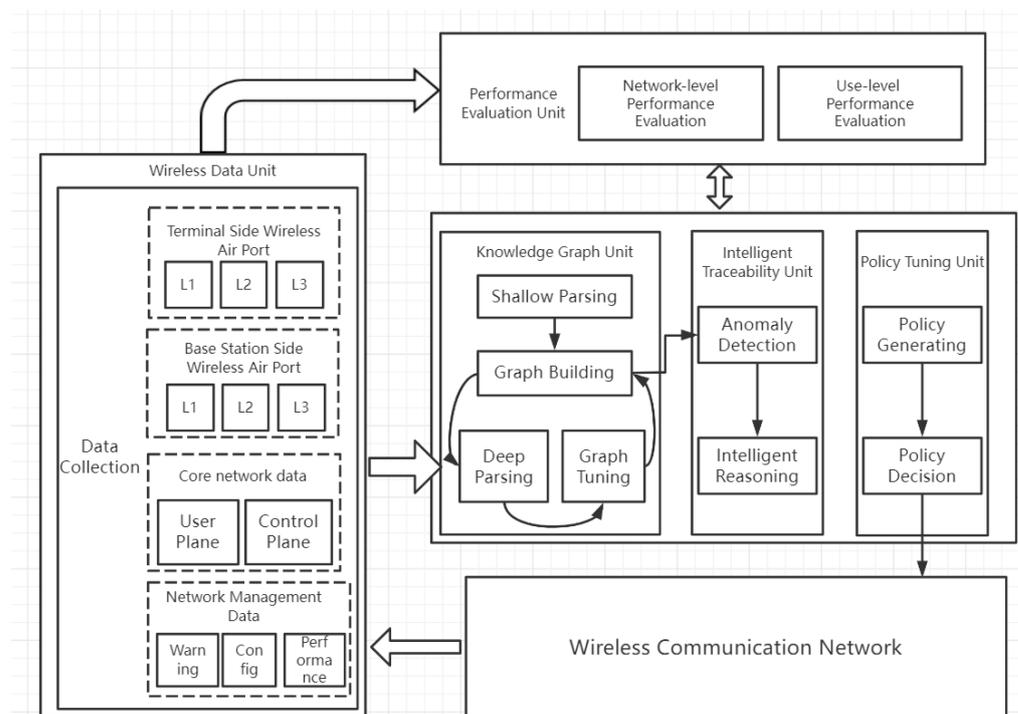


Figure 12. The application architecture of wireless communication network knowledge graph.

5.2. Question Answering (QA) Systems

The QA system is an important development direction in the field of natural language processing, which aims to acquire knowledge through natural language. In many real application scenarios, knowledge graph-based question answering systems can provide great convenience to knowledge acquisition. However, in addition to simply retrieving entities from knowledge graphs, more questions require the ability of knowledge reasoning. The knowledge reasoning methods introduced in this paper can be used in knowledge question answering systems based on knowledge graphs to answer complex questions and improve the accuracy and completeness of results.

In addition to embedding knowledge graphs into general knowledge question answering systems, knowledge reasoning technology can also be directly applied to question answering systems to solve specific problems. For example, R-GCN [34] is used to model

the dialogue structure and background knowledge of multi-round dialogue question answering systems. Question answering systems based on knowledge graph embedding are emerging. For example, the TransE [28] vector space has been combined with search technology to realize a question answering system based on knowledge graph embedding. With the deepening of modern knowledge graph reasoning methods, the effect of knowledge question answering has been continuously improved. Meanwhile, the methods that combine the above knowledge graph reasoning techniques and question answering techniques to solve specific problems are also emerging.

5.3. Recommendation Systems

Recommendation systems are a rising technology with the development of the Internet and the great abundance of information, aiming to understand users and actively recommend information to users. With the development of knowledge graph technology, knowledge graphs and recommendation systems are comprehensively integrated so as to better understand users, better match users' needs, and provide stronger explanations.

In terms of understanding users, a network of relationships between knowledge graphs and users that buy the same item or watch the same video values the same information, such as user behavior, map construction, and using knowledge graph reasoning technology. To deduce the relationship between the user and the depth of the mining potential demand, the network cooperates with the collaborative filtering recommendation technology such as to improve the effect of the recommendation. The literature uses knowledge graphs to enrich user information and R-GCN [34] to model and provide reasoning about relationships and structures to better understand users. The recommended content (such as goods, information, knowledge, people, etc.) can also be constructed through various relationships to construct knowledge graphs, and reasoning technology can be used to mine potential features and infer potential relationships so as to enhance the understanding of the recommended content and improve the recommendation effect. The literature attempts to use TransR [60] to model structured recommendation content to improve recommendation effectiveness. On this basis, a recommendation itself can be regarded as a complex network relationship between user and the recommended content, so knowledge graph reasoning technology can be applied to directly implement recommendations. KGAT [73] combines TransR [60] and deep learning to realize the recommendation method with knowledge graph attention networks.

With the further development and maturity of knowledge graphs and knowledge reasoning technology, various modern knowledge reasoning methods will be applied more and more in the field of recommendation, and will gradually become the mainstream methods.

5.4. Personalized Search

Personalized search makes full use of the historical behavior records such as searching, clicking, and the user's own information to achieve a personalized search and return more matching user results. When searching for "Apple", for example, fruit growers and gadget lovers have different expectations. When searching for "weather forecast", people in Shanghai and Beijing have different expectations of the search results. Personalized search is a technology aimed at solving such problems. The method of user modeling using knowledge graph and reasoning technology in recommendation systems can be used for personalized search.

Directly using knowledge reasoning technology to realize personalized search is also being attempted. In the literature, users, documents, and the interactive relations between users and documents, such as searching and clicking, are constructed into knowledge graphs, and the TransE [28] reasoning method is used to realize personalized search.

6. Future Directions

In this section, we discuss the future directions of knowledge graph reasoning.

(1) **Knowledge reasoning for multivariate relations.** Compared with unitary and binary relations, multivariate relations are diverse in structure, complex in semantics, and difficult to deal with. Therefore, the existing knowledge processing work mainly focuses on binary relations. There are few studies on multivariate relations, and the existing studies usually treat it as a binary relation, which loses a lot of semantic information. However, multivariate relationships are not a minority in knowledge graphs; in Freebase, for example, more than one-third of entities are involved in multivariate relationships. In the future, it is necessary to study how to implement representation and representation learning of multivariate relations and how to combine representation and reasoning ability.

(2) **Knowledge reasoning based on the fusion of multi-source information and multiple methods.** Knowledge reasoning based on fusion of multi-source information can reduce the disconnectedness and sparsity of knowledge graphs by combining a text corpus or other knowledge graph with additional information. Knowledge reasoning based on the fusion of multiple methods can complement each other's advantages and improve reasoning performance by mixing different methods at a deeper level. For example, reasoning of co-modeling rules and knowledge graphs [74] is a current method with outstanding performance. In addition, due to the outstanding performance of neural networks in various fields including knowledge graphing, the fusion of neural networks and other complementary methods will become a major focus of future research. For example, Neural LP [35] obtained excellent reasoning results by combining the strong learning and generalization ability of a neural network with the high accuracy and interpretability of the rule method. In two commonly used data sets, WN18 (a subset of the knowledge graph WordNet) and FB15k (a subset of the knowledge graph Freebase [9]), the proportion of valid entities ranked in the top 10 for entity prediction tasks reached 99.8% and 91.6%, respectively. The integration of multi-source information and multiple methods simultaneously to further improve reasoning performance will also become a major research direction in the future. Between them, the fusion mode (that is, the merging of the two concepts) is a major difficulty.

(3) **Knowledge reasoning based on few-shot learning.** Existing knowledge reasoning models often need a large number of high-quality samples for training and learning, which requires a large cost to acquire samples. In practice, it is even difficult to obtain a large number of training samples, which greatly limits the application scope of existing knowledge reasoning models. On the other hand, people can learn reasoning quickly with only a few samples of relevant prior knowledge. In this process, the brain senses the external environment, pays attention to the information that it is interested in or needs to learn, and quickly establishes new knowledge by combining it with existing prior knowledge, and then forms long-term memory that is difficult to be forgotten after processing and arrangement by neurons. As a result, people learn to deal with increasingly complex tasks by building and integrating knowledge from life experiences. In the process of continuous learning, people can quickly learn new tasks with little training by retrieving and utilizing previous knowledge.

(4) **Dynamic knowledge reasoning.** For example, Tay et al. [75] proposed puTransE, which realized block learning and integrated reasoning of knowledge graph through a divide-and-conquer strategy, and effectively dealt with dynamic addition and deletion of knowledge graphs. For a new triplet, it can learn by creating a new parallel space; for deleted triples, this is done by invalidating the corresponding representation space during prediction. However, puTransE cannot directly process the dynamic modification of a knowledge graph (of course, a modification operation can be handled indirectly by simply treating a modification operation as a deletion and then an addition), and the deletion operation does not usually actually delete fact tuples, but exceeds the effective time of fact tuples. Fact tuples become invalid, and within the effective time, the fact tuple still holds.

7. Summary

In the paper, we attempted to establish the basics for future research through a thorough literature review of knowledge graph reasoning approaches. More specifically, firstly, we reviewed the basic concepts of knowledge graphs and classified and summarized the methods related to knowledge graphs from three aspects: knowledge representation, knowledge extraction, and knowledge fusion. Then, we started from the definition of knowledge graph reasoning and explored its applications in industry. Then, we classified knowledge graph reasoning methods and introduced and compared the various types of knowledge graph reasoning algorithms from classical models to the latest developments.

Author Contributions: Conceptualization, resources and writing—original draft preparation, Y.C.; writing—review and editing, H.L. (Hao Li); validation, H.L. (Han Li); formal analysis, W.L.; investigation, data curation, project administration, Y.W.; supervision, funding acquisition, validation, Q.H.; visualization, validation, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by a grant from the National Key R&D Program of China under Grant No. 2021YFB3900601, the Fundamental Research Funds for the Central Universities under Grant B220202074, the National Natural Science Foundation of China under Grant No. 62172438, the Fundamental Research Funds for the Central Universities, JLU, and the Joint Foundation of the Ministry of Education (No. 8091B022123).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Ren, H.; Dai, H.; Dai, B.; Chen, X.; Zhou, D.; Leskovec, J.; Schuurmans, D. SMORE: Knowledge Graph Completion and Multi-hop Reasoning in Massive Knowledge Graphs. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022.
- Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Yu, P.S. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 494–514. [[CrossRef](#)] [[PubMed](#)]
- Wu, Y.; Ma, Y.; Wan, S. Multi-scale relation reasoning for multi-modal Visual Question Answering. *Signal Process. Image Commun.* **2021**, *96*, 116319. [[CrossRef](#)]
- Wu, Y.; Ji, X.; Ji, W.; Tian, Y.; Zhou, H. CASR: A context-aware residual network for single-image super-resolution. *Neural Comput. Appl.* **2020**, *32*, 14533–14548. [[CrossRef](#)]
- Li, G.; He, B.; Wang, Z.; Zhou, Y. A Data Agent Inspired by Interpersonal Interaction Behaviors for Wireless Sensor Networks. *IEEE Internet Things J.* **2022**, *9*, 8397–8411. [[CrossRef](#)]
- Xie, K.; Jia, Q.; Jing, M.; Yu, Q.; Yang, T.; Fan, R. Data Analysis Based on Knowledge Graph. In Proceedings of the 15th International Conference on Broad-Band and Wireless Computing, Communication and Applications, Yonago, Japan, 28–30 October 2020.
- Liu, W.; Yin, L.; Wang, C.; Liu, F.; Ni, Z. Medical Knowledge Graph in Chinese Using Deep Semantic Mobile Computation Based on IoT and WoT. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5590754. [[CrossRef](#)]
- Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z.G. DBpedia: A Nucleus for a Web of Open Data. In Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference, Busan, Korea, 11–15 November 2007; pp. 722–735.
- Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the Special Interest Group on Management Of Data, Vancouver, BC, Canada, 9–12 June 2008; pp. 1247–1250.
- Hoffart, J.; Suchanek, F.M.; Berberich, K.; Weikum, G. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artif. Intell.* **2013**, *194*, 28–61. [[CrossRef](#)]
- Dong, B.X.; Gabilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmman, T.; Sun, S.; Zhang, W. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 601–610.
- Wu, Q.; Zhang, H.; Gao, X.; He, P.; Weng, P.; Gao, H.; Chen, G. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In Proceedings of the International World Wide Web Conference (WWW), San Francisco, CA, USA, 13–17 May 2019.
- Wang, H.; Zhang, F.; Xie, X.; Guo, M. Dkn: Deep knowledge-aware network for news recommendation. In Proceedings of the International World Wide Web Conference (WWW), Lyon, France, 23–27 April 2018.

14. Wang, H.; Zhang, F.; Zhang, M.; Leskovec, J.; Zhao, M.; Li, W.; Wang, Z. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Anchorage, AK, USA, 4–8 August 2019.
15. Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; Leskovec, J. Graph convolutional neural networks for web-scale recommender systems. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), London, UK, 19–23 August 2018.
16. Sun, H.; Dhingra, B.; Zaheer, M.; Mazaitis, K.; Salakhutdinov, R.; Cohen, W.W. Open domain question answering using early fusion of knowledge bases and text. In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP), Brussels, Belgium, 31 October–4 November 2018.
17. Sun, H.; Bedrax-Weiss, T.; Cohen, W.W. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP), Hong Kong, China, 3–7 November 2019.
18. Berant, J.; Chou, A.; Frostig, R.; Liang, P. Semantic parsing on freebase from question-answer pairs. In Proceedings of Empirical Methods in Natural Language Processing the (EMNLP), Grand Hyatt Seattle, Seattle, DC, USA, 18–21 October 2013.
19. Saxena, A.; Tripathi, A.; Talukdar, P. Improving multihop question answering over knowledge graphs using knowledge base embeddings. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Seattle, WA, USA, 5–10 July 2020.
20. Ren, H.; Dai, H.; Dai, B.; Chen, X.; Yasunaga, M.; Sun, H.; Schuurmans, D.; Leskovec, J.; Zhou, D. Lego: Latent execution-guided reasoning for multi-hop question answering on knowledge graphs. In Proceedings of the International Conference on Machine Learning (ICML), Vienna, Austria, 18–24 July 2021.
21. Lin, B. Y.; Chen, X.; Chen, J.; Ren, X. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP), Hong Kong, China, 3–7 November 2019.
22. Feng, Y.; Chen, X.; Lin, B. Y.; Wang, P.; Yan, J.; Ren, X. Scalable multi-hop relational reasoning for knowledgeable question answering. In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP), Hong Kong, China, 3–7 November 2020.
23. Yasunaga, M.; Ren, H.; Bosselut, A.; Liang, P.; Leskovec, J. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), Online, 6–11 June 2021.
24. Lv, S.; Guo, D.; Xu, J.; Tang, D.; Duan, N.; Gong, M.; Shou, L.; Jiang, D.; Cao, G.; Hu, S. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), New York, NY, USA, 7–12 February 2020.
25. Ruiz, C.; Zitnik, M.; Leskovec, J. Identification of disease treatment mechanisms through the multiscale interactome. *Nat. Commun.* **2021**, *12*, 1796 [CrossRef] [PubMed]
26. Zitnik, M.; Agrawal, M.; Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* **2018**, *34*, i457–i466. [CrossRef] [PubMed]
27. Ioannidis, V. N.; Song, X.; Manchanda, S.; Li, M.; Pan, X.; Zheng, D.; Ning, X.; Zeng, X.; Karypis, G. Drkg—Drug Repurposing Knowledge Graph for COVID-19. 2020. Available online: <https://github.com/gnn4dr/DRKG/> (accessed on 27 July 2022).
28. Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In Proceedings of the Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–8 December 2013;
29. Nickel, M.; Tresp, V.; Krieger, H.-P. A Three-Way Model for Collective Learning on Multi-Relational Data. In Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, WA, USA, 28 June–2 July 2011; pp. 809–816.
30. Yang, B.; Yih, W.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and reasoning in Knowledge Bases. In Proceedings of the International Conference on Machine Learning (ICLR) (Poster), San Diego, CA, USA, 7–9 May 2015.
31. Galárraga, L.A.; Teflioudi, C.; Hose, K.; Suchanek, F.M. AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In Proceedings of the 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 413–422.
32. Muggleton, S.H. Inductive Logic Programming. *New Gener. Comput.* **1991**, *8*, 295–318. [CrossRef]
33. Socher, R.; Chen, D.; Manning, C.D.; Ng, A.Y. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In Proceedings of the Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–8 December 2013;
34. Schlichtkrull, M.S.; Kipf, T.N.; Bloem, P.; Berg, R.v.; Titov, I.; Welling, M. Modeling Relational Data with Graph Convolutional Networks. In Proceedings of the Extended Semantic Web Conference (ESWC), Heraklion, Crete, Greece, 3–7 June 2018; pp. 593–607.
35. Wang, P.-W.; Stepanova, D.; Domokos, C.; Kolter, J.Z. Differentiable learning of numerical rules in knowledge graphs. In Proceedings of the 8th International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.
36. Xiong, W.; Hoang, T.; Wang, W.Y. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP), Copenhagen, Denmark, 9–11 September 2017; pp. 564–573.
37. Das, R.; Dhuliawala, S.; Zaheer, M.; Vilnis, L.; Durugkar, I.; Krishnamurthy, A.; Smola, A.; McCallum, A. Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning. In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP), Copenhagen, Denmark, 9–11 September 2018.

38. Guo, S.; Wang, Q.; Wang, L.; Wang, B.; Guo, L. Knowledge Graph Embedding With Iterative Guidance From Soft Rules. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 4816–4823.
39. Vrandečić, D.; Krötzsch, M. Wikidata: A free collaborative knowledgebase. *Commun. ACM* **2022**, *57*, 78–85. [[CrossRef](#)]
40. Speer, R.; Havasi, C. ConceptNet5: A Large Semantic Network for Relational Knowledge. In *The People's Web Meets NLP, Collaboratively Constructed Language Resources*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 161–176.
41. Lin, Y.; Liu, Z.; Luan, H.; Sun, M.; Rao, S.; Liu, S. Modeling Relation Paths for Representation Learning of Knowledge Bases. In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal, 17–21 September 2015; pp. 705–714.
42. Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; Xu, B. Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1: Long Papers, pp. 1227–1236.
43. Feng, J.; Huang, M.; Zhao, L.; Yang, Y.; Zhu, X. Reinforcement Learning for Relation Classification From Noisy Data. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 5779–5786.
44. Morwal, S.; Jahan, N.; Chopra, D. Named Entity Recognition Using Hidden Markov Model(HMM). *Int. J. Nat. Lang. Comput. (IJNLC)* **2012**, *1*, 15–23. [[CrossRef](#)]
45. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv* **2015**, arXiv:1508.01991.
46. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
47. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Tian, H.; Wu, H.; Wang, H. Ernie 2.0: A continual pre-training framework for language understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 8968–8975.
48. Zheng, S.; Hao, Y.; Lu, D.; Bao, H.; Xu, J.; Hao, H.; Xu, B. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing* **2017**, *257*, 59–66. [[CrossRef](#)]
49. Soares, L.B.; FitzGerald, N.; Ling, J.; Kwiatkowski, T. Matching the Blanks: Distributional Similarity for Relation Learning. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019; Volume 1: Long Papers, pp. 2895–2905.
50. Cai, R.; Zhang, X.; Wang, H. Bidirectional Recurrent Convolutional Neural Network for Relation Classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL, 2016, Berlin, Germany, 7–12 August 2016; Volume 1: Long Papers.
51. Miwa, M.; Bansal, M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL, Berlin, Germany, 2016, 7–12 August 2016; Volume 1: Long Papers.
52. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* **2019**, *38*, 146:1–146:12. [[CrossRef](#)]
53. Zeng, D.; Liu, K.; Chen, Y.; Zhao, J. Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, 17–21 September 2015; pp. 1753–1762.
54. Agichtein, E.; Gravano, L. Snowball: Extracting relations from large plain-text collections. In Proceedings of the Fifth ACM Conference on Digital Libraries, San Antonio, TX, USA, 2–7 June 2000; pp. 85–94.
55. Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka, E.R., Jr.; Mitchell, T.M. Toward an Architecture for Never-Ending Language Learning. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 11–15 July 2010.
56. Wu, L.; Petroni, F.; Josifoski, M.; Riedel, S.; Zettlemoyer, L. Scalable Zero-shot Entity Linking with Dense Entity Retrieval. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 6397–6407.
57. Li, C.; Cao, Y.; Hou, L.; Shi, J.; Li, J.; Chua, T.-S. Semi-supervised Entity Alignment via Joint Knowledge Embedding Model and Cross-graph Model. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019; pp. 2723–2732.
58. Zeng, W.; Zhao, X.; Tang, J.; Li, X.; Luo, M.; Zheng, Q. Towards Entity Alignment in the Open World: An Unsupervised Approach. In Proceedings of the Database Systems for Advanced Applications (DASFAA), Taipei, Taiwan, 11–14 April 2021; pp. 272–289.
59. Zhu, H.; Xie, R.; Liu, Z.; Sun, M. Iterative Entity Alignment via Joint Knowledge Embeddings. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, 19–25 August 2017; pp. 4258–4264.
60. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2181–2187.

61. Nguyen, D.Q.; Vu, T.; Nguyen, T.D.; Nguyen, D.Q.; Phung, D.Q. A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization. In Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Minneapolis, MN, USA, 2–7 June 2019; pp. 2180–2189.
62. Zhang, Z.; Zhuang, F.; Zhu, H.; Shi, Zh.; Xiong, H.; He, Q. Relational Graph Neural Network with Hierarchical Attention for Knowledge Graph Completion. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 9612–9619.
63. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the 27th Annual Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–8 December 2013; pp. 3111–3119.
64. Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. Knowledge Graph Embedding via Dynamic Mapping Matrix. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL), Beijing, China, 26–31 July 2015; pp. 687–696.
65. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge Graph Embedding by Translating on Hyperplanes. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014.
66. Cohen, W.W. TensorLog: A Differentiable Deductive Database. *arXiv* **2016**, arXiv:1605.06523.
67. Wang, W.Y.; Mazaitis, K.; Cohen, W.W. Programming with personalized pagerank: A locally groundable first-order probabilistic logic. In Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, 27 October–1 November 2013; pp. 2129–2138.
68. Wang, W.Y.; Mazaitis, K.; Lao, N.; Cohen, W.W. Efficient reasoning and learning in a large knowledge base—Reasoning with extracted information using a locally groundable first-order probabilistic logic. *Mach. Learn.* **2015**, *100*, 101–126. [[CrossRef](#)]
69. Paulheim, H.; Bizer, C. Improving the Quality of Linked Data Using Statistical Distributions. *Int. J. Semant. Web Inf. Syst* **2014**, *10*, 63–86. [[CrossRef](#)]
70. Jang, S.; Megawati; Choi, J.; Yi, M.Y. Semi-Automatic Quality Assessment of Linked Data without Requiring Ontology. In Proceedings of the 14th International Semantic Web Conference (ISWC), Bethlehem, PA, USA, 11 October 2015; pp. 45–55.
71. Shen, Y.; Huang, P.-S.; Chang, Mi.; Gao, J. Implicit ReasonNet: Modeling Large-Scale Structured Relationships with Shared Memory. *arXiv* **2016**, arXiv:1611.04642v1.
72. Shi, B.; Wenginger, T. Open-World Knowledge Graph Completion. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 1957–1964.
73. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.-S. KGAT: Knowledge Graph Attention Network for Recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, 4–8 August 2019.
74. Guo, S.; Wang, Q.; Wang, L.; Wang, B.; Guo, L. Jointly embedding knowledge graphs and logical rules. In Proceedings of the Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–4 November 2016; pp. 1488–1498.
75. Tay, Y.; Luu, A.T.; Hui, S.C. Non-Parametric estimation of multiple embeddings for link prediction on dynamic knowledge graphs. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 1243–1249.