This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3505433

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.Doi Number

Leveraging Non-Parametric Reasoning with Large Language Models for Enhanced Knowledge Graph Completion

Ying Zhang[1], YangPeng Shen[2], Gang Xiao[1], JingHui Peng[3]

[1] Institute of Systems Engineering, Academy of Military Sciences (AMS), Beijing 100107, China.
[2] CITIC Bank, Beijing 100123, China.
[3] Anhui Polytechnic University, Anhui Province, 241000, China.

Corresponding author: Ying Zhang (e-mail: yingzhang199608@foxmail.com)

This work was supported by the National Key Laboratory for Complex Systems Simulation Foundation (NO.6142006190301).

ABSTRACT In recent years, the increasing complexity of research tasks, the expansion of application scenarios, the iterative enhancement of user demands, and the intensification of data specialization have led to the continuous expansion of existing models, with model parameters growing exponentially. Although the increased model parameters have significantly improved accuracy under the current computing environment, this has also resulted in excessive consumption of computational resources and challenges in model scalability, presenting substantial obstacles for practical applications. The issue of balancing model parameter scale with practical application requirements has thus become a primary concern for researchers. This paper introduces a framework that integrates large language models with case-based reasoning for knowledge graph completion. The proposed framework aims to address the issue of knowledge graph completeness by combining non-parametric reasoning with the advanced semantic understanding capabilities of large language models, while also addressing challenges such as new entity information, parameter explosion, and limited generalization ability.

INDEX TERMS Case-based reasoning, large language model, information entropy, knowledge graph completion.

I. INTRODUCTION

Currently, knowledge graphs serve as a foundational function of artificial intelligence by interrelating entities through triples and presenting them systematically to users, thereby helping users efficiently obtain accurate information. To keep pace with technological advancements, various industries have introduced related knowledge graphs aimed at providing targeted services such as information retrieval, intelligent Q&A, and smart recommendations. However, due to the use of diverse data sources and construction techniques in building knowledge graphs, the inconsistency in data source quality and technical imperfections inevitably result in quality issues. These issues include compromised accuracy, lack of completeness, data redundancy, and loss of timeliness. Consequently, specific quality dimensions are defined, and corresponding evaluation schemes and enhancement methods are designed to monitor and improve the quality of knowledge graphs.

Completeness, as a critical quality dimension, measures the extent to which a knowledge graph covers knowledge in a specific domain or meets user needs. A lack of completeness significantly impacts the usability of a knowledge graph. In the context of search queries, inadequate feedback diminishes user experience and erodes trust in the knowledge graph. Enhancing completeness ensures more comprehensive search feedback, thereby improving user experience. Consequently, addressing completeness issues is essential for quality improvement. Currently, completeness can be enhanced using knowledge inference techniques to deduce unknown information from known data [1]. For example, the relationship between a pair of entities can be inferred, or the corresponding tail entity can be deduced from a known entity and relationship [2]. These inference scenarios rely on the internal semantic and structural information within the knowledge graph, supplemented by external data resources.

At present, knowledge inference methods focus on three primary areas. (1) Inference Based on Logical Rules: Early knowledge inference tasks involved applying mined rules to the knowledge graph to discover unexpressed facts. Common techniques include AMIE [3], AMIE+ [4], RDF2Rules [5], NELLs [6], and ProPPR [7]. The PRA model [8], which uses path rules, has introduced new approaches for large-scale knowledge graph inference. Models based on PRA include those utilizing subgraphs [9] and those combining global and subgraph information [10,11]. (2) Inference Based on Distributed Representation: The rise of embedding technology has made rule-based methods less flexible and straightforward for various knowledge graphs. Researchers now focus on using entity and relationship embeddings for inference tasks. Common models include the tensor decomposition-based RESCAL [12-14], translation-based models such as TransE [15], TransH [16], and TransR [17], and extended models like t-TransE [18], TAE-TransE [19], and TranSparse [20]. To enhance model accuracy, integrating various types of information, such as entity descriptions, entity types, relationship descriptions, related knowledge graphs, and textual information, is effective. Notable models include TEKE [21], TKRL [22], and MKRL [23]. (3) Inference Based on Neural Networks: Neural network-based inference techniques are as significant as those based on distributed representation. Neural networks' powerful capabilities enable their application to various natural language processing tasks. Knowledge inference leverages neural networks to encode entities, relationships, and attributes with related textual descriptions and external data, or to achieve better results through complex models. Well-known models include DKRL [24], ConMask [25], ConvE [26], DSKG [27], DeepPath [28], and MINERVA [29].

Overall, an increasing number of knowledge inference tasks tend to enhance model accuracy from the following two aspects. First, external data resources are sought. However, in practice, suitable external resources are sometimes difficult to obtain or require substantial human and computational costs. Thus, integrating external text into completion models has certain limitations in practical tasks. This approach heavily relies on the effectiveness of the text to improve the precision and scope of completion, making it challenging to achieve effective expansion. Second, model improvements are pursued to enhance the final inference results. This is mainly achieved by experimenting with different training algorithms, increasing model parameters, designing relevant negative sampling methods, and stacking models to improve performance. However, these techniques often validate model inference accuracy under ideal dataset conditions, where commonly used experimental datasets are relatively orderly and lack the exceptional cases found in the real world. These datasets also assume that there are no new entities and relationships added, making the models trained on them difficult to apply directly to real-world scenarios. Additionally, models trained in experimental settings tend to overlook computational resource consumption and the difficulty of model expansion when new entities and relationships are encountered in practical applications. In short, most existing models struggle to balance prediction accuracy and resource consumption when applied in real-world scenarios. Therefore, to effectively address these issues and implement experimental models practically, researchers are increasingly turning to case-based reasoning methods.

Case-based reasoning (CBR) has been applied to knowledge inference tasks. Das et al. [54,55] were among the first researchers to extend CBR to the problem of knowledge graph completion. By utilizing the four steps of retrieval, reuse, revision, and retention [54], and considering the importance of different relationship paths, they significantly improved completion accuracy. Although CBR methods effectively address issues such as parameter explosion in parametric inference models, difficulty in generalization, catastrophic forgetting with new data, and the lack of interpretability of results, they sometimes fail to provide answers for certain queries. This is due to the inability to reuse paths stored in the case memory for the current query entity. Additionally, while existing CBR methods can generate numerous candidate entities, they lack refined methods for selecting the most accurate answers. Thus, improving the accuracy of completion tasks remains a pressing issue. Furthermore, current methods typically rank paths based on their frequency, assuming that entities confirmed by multiple paths are more accurate. However, more rational and effective path ranking methods need to be explored. To address these issues, we propose the CBR-LLM model, which combines large language models (LLMs) with CBR for knowledge graph completion. In the CBR module, we match paths with similar semantics by considering path semantics, thereby expanding the range of reusable paths for query entities. Additionally, leveraging the strengths of graph neural networks, we filter and obtain precise, relevant entities while effectively reducing irrelevant interference, thus laying a solid foundation for subsequent completion tasks. By introducing information entropy, we select low-entropy paths as effective paths for inferring answer entities. Furthermore, we efficiently utilize the powerful automatic generation capabilities of LLMs to identify a series of potential tail entity candidates for the completion task. By integrating these candidate answers from the LLM module with the output of the CBR module, we conduct a comprehensive analysis to derive the final, accurate completion results, ensuring high efficiency and precision in information completion.

To further validate the performance and effectiveness of the CBR-LLM model, we introduced a new movies knowledge graph and conducted comprehensive experiments with detailed designs. Throughout the experiments, we not only compared the performance of CBR-LLM with other mainstream models on relevant standard metrics but also

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3505433

focused particularly on its ability to handle new entities. The experimental results demonstrated significant advantages of CBR-LLM in addressing the aforementioned challenges.

II. RELATED WORKS

Case-Based Reasoning (CBR) was first introduced by Schank in 1983 [30]. It involves a process of solving new tasks by learning from previously resolved similar cases and making appropriate modifications [30,31]. CBR emphasizes extracting solutions from historical experiences and adjusting these solutions in new contexts to fit current problems.

Non-parametric methods refer to statistical methods that do not require pre-set fixed forms of probability distributions or model structures [32-34]. The characteristic of these methods is that they do not make strong assumptions about the form of the data, and as the sample size increases, they can become more complex to adapt to the characteristics of the data [33,34]. Therefore, non-parametric methods can more flexibly adapt to the characteristics of the data itself.

Since CBR is considered a non-parametric model, it effectively infers without significant adjustments as new data becomes available, unlike traditional models that require fixed structures. Given this characteristic, CBR has gradually been applied to static and dynamic knowledge graphs of varying scales and domains, achieving favorable inference results. Furthermore, in recent years, CBR methods have gained widespread application not only in the field of artificial intelligence [30,31,35-37] but also in natural language processing, where they have demonstrated unique advantages, leading to a growing body of related research [38-44].

Currently, in knowledge graph reasoning tasks, case-based reasoning methods have gradually gained the attention of researchers because they are more adapted to the characteristics of practical reasoning tasks. Das et al [44] utilized the feature of graph structure similarity of local subgraphs to answer knowledge-base-based questions and proposed the CBR-SUBG semiparametric model. Unlike other case-based reasoning approaches, Das et al [44] argued that the representations of the answer nodes in the query subgraphs should be the closest to each other, and thus the final answer node of a query can be efficiently found by approximating the representations of the answer nodes in different subgraphs in the vector space during the training phase. Similar to general case-based reasoning approaches, the CBR-SUBG model realizes Q&A mainly through the following three stages, firstly, the query is encoded by the pre-training model, and the query with a similar relationship is obtained by entity masking retrieval. Next, the similar queries obtained in the previous stage are utilized to generate a minimal subgraph including inference patterns and answers. Specifically, the smallest subgraph about the query and similar queries is obtained by collecting the relationship paths between the query entities connected to the answer entities. Finally, the correct target answer node is obtained by a graph neural network on the query subgraph based on the relational reasoning paths obtained in the previous stage. Similarly, Ye et al [45] applied a case-based reasoning approach to a multimodal dialog system by finding similar case responses, so that the final responses produced are traceable and more in line with the laws of natural language. Similar to the general case-based reasoning approach, the RERG model proposed by Ye et al. mainly accomplishes the retrieval of similar neighbors and the generation of new dialogues through two components the retrieval module and the reuse module. To obtain accurate similar cases in a specific context, RERG utilizes comparative learning in the retrieval module to obtain valid textual and image information in the conversation context and then is trained by Triplet Ranking Loss to obtain a higher level of similarity in the conversation context, and then obtains the similarity ranking of similar cases. In the reuse module, since the dialog system needs to generate smooth and natural response texts in the context of the dialog, the text generation part generates the corresponding response texts by focusing on the current dialog context as well as the dialog contexts and responses of the similar cases at the same time. The image reuse part searches for the image with the highest similarity in the training set by utilizing the generated response and integrating the image representations from similar cases as a query and co-submits it with the text generated above to get the final response.

Moreover, besides utilizing the case-based reasoning approach in the above scenarios, Wiseman et al [46] have also successfully applied it to text-generation techniques. By finding sentences that are as similar as possible to the target sentence, and then combining them so that they can be modified with as few as possible to get the final desired text. Atzeni et al [47] solve the problem of text-based games by combining a case-based reasoning approach with reinforcement learning to enable better generalization, especially when it can be achieved efficiently in non-distributed environments. Facing the intractable problem of semantic drift in multi-hop natural language reasoning tasks, Valentino et al [48] based on the case-based reasoning methodology to obtain a reasonable explanation of natural language hypotheses through three steps that are retrieval, reuse, and refinement. In addition, for the subtask of relationship prediction in scientific text Swarup et al [49] performed well in the practical task context of small data volume and unavailability of labels for similar neighbors by reasonably combining the local instance-level approach and the global parameterization approach based on case-based reasoning. In recent years, with the emergence of knowledge graph quality enhancement tasks, researchers have also gradually applied the case-based reasoning approach to the knowledge graph complementation task to improve the completeness of the knowledge graph.

In recent years, particularly between 2022 and 2024, significant advancements in large language models have not only propelled the development of natural language processing but also found widespread applications in various domains, including multimodal semantic communication, 6G communication systems, automated edge computing, and autonomous drone planning systems. For example, Jiang et al. [50] propose a multimodal semantic communication framework based on large AI models, which addresses semantic consistency issues during the conversion of multimodal data to unimodal text data through the integration of LLMs. This framework also utilizes conditional generative adversarial networks (CGANs) to estimate wireless channel state information, alleviating signal distortion challenges in semantic communication. Additionally, Jiang et al. [51] further explore the application of LLMs in 6G communication systems, proposing an enhanced multi-agent system. This system optimizes communication task processing in a 6G environment by leveraging the natural language understanding and generation capabilities of LLMs, supported by multi-agent data retrieval, collaborative planning, and evaluation reflection mechanisms. In the field of automated edge computing, research by Shen et al. [52] introduces LLMs to empower edge computing nodes with autonomous processing and intelligent decision-making capabilities, enabling devices to efficiently support smart connectivity at the network edge. Furthermore, Zhong et al. [53] present a vision-based autonomous planning system that integrates dynamic obstacle trajectory prediction technology, utilizing LLMs to create a safer vision-based autonomous planning system, providing efficient navigation and obstacle avoidance solutions for quadcopters.

It is noteworthy that significant advancements have also been made in the field of knowledge graph completion using large language models, with several representative completion models being proposed in succession. For instance, Lv, Lin et al. proposed PKGC [56], which combines traditional triple data with additional contextual information. Through meticulously designed human-assisted prompt templates, PKGC aims to transform structured triple information into coherent and understandable text, facilitating processing and parsing by large language models. Leveraging the powerful semantic understanding and generation capabilities of LLMs, PKGC executes binary classification tasks. Saxena, Kochsiek et al. introduced the KGT5 model [57], which presents a novel solution. Built on the lightweight T5 architecture, KGT5 is trained from scratch using random initialization, providing a new research approach for knowledge graph completion tasks. This approach not only demonstrates the potential to achieve high-performance knowledge completion with smaller model scales but also significantly enhances task scalability and generalization capabilities. Chen, Chen et al. proposed the KG-S2S model [58], which innovatively uses quads to restate traditional triple information, introducing a higher-level semantic expression for knowledge graph completion tasks. This method enriches the depth of context input for the model and significantly enhances its applicability and performance in handling time-sensitive and small-sample scenario knowledge graphs. By employing quads, KG-S2S enables the model to capture dynamic changes between entities and subtle relationships in sparse data, thereby demonstrating superior generalization ability and completion accuracy in complex knowledge graph completion tasks. Li, Yi et al. presented the LP-BERT model [59], which serves knowledge graph completion tasks by implementing a multi-task pre-training strategy. Specifically, LP-BERT designs tasks for masked entity modeling and masked relation modeling, aiming to enhance the model's understanding and generation capabilities of entity and relation embeddings by randomly masking entities or relations in the knowledge graph and predicting the missing parts. Furthermore, LP-BERT utilizes contrastive learning as an optimization strategy to maximize the difference between positive and negative samples, promoting the model's ability in complex relationship inference and pattern recognition. Choi, Youngjoong et al. proposed OpenWorld KGC [60], which systematically addresses open-world knowledge graph completion problems by constructing a comprehensive pipeline framework. This framework integrates tasks for entity description prediction and incomplete triple prediction, surpassing traditional closed-world assumptions in KGC methods. Building upon MEM-KGC, OpenWorld KGC incorporates a masked language module to predict potential entity description information, enhancing the model's depth of entity context understanding while predicting missing or incomplete elements in the knowledge graph. This mechanism not only enriches the model's ability to capture complex associations between entities but also significantly enhances its inference and generation capabilities when dealing with partially observed information. These cutting-edge completion models exhibit unique characteristics in their methods, collectively driving the development of knowledge graph completion technology facilitated by large language models.

At present, various studies are focused on enhancing the training and inference efficiency of large models through diverse technical approaches, enabling them to perform better under limited resources and across different application scenarios and domains. For instance, Tuan, Moore, et al. [61] proposed a method using LoRA and DeepSpeed to address the resource constraints faced by generative AI technologies. By reducing training costs and improving model efficiency, this approach facilitates the adoption and customization of large language models by small and medium-sized enterprises. Pham, Moore, et al. [62] reduced computational and human resource costs by combining hedge algebra with multilingual large language models to extract hidden rules from large datasets, thereby improving applications such as ChatGPT and generative AI, and enabling more efficient use of large language models in practical scenarios. Yang, Zhu, et al. [63] introduced the CP-KGC framework to support large language models in data augmentation. This framework employs prompts tailored to specific domain datasets to enhance semantic richness, thereby improving model efficiency and reducing resource consumption. Guo, Yang, et al. [64] proposed a framework named KnowledgeNavigator, which enhances the performance and accuracy of large language models in complex reasoning tasks by incorporating external knowledge and optimizing reasoning paths, while also improving resource efficiency. These methods predominantly leverage external data resources to enhance the performance of large language models in completing natural language tasks. The external data resources typically include synthetic datasets, knowledge graphs, expert-generated prompts, or other related data obtained through various means. In summary, much like traditional models, most approaches utilizing large language models rely on high-quality, highly relevant external data, raising concerns about the labor and computational costs, as well as the complexity involved in processing large datasets. Additionally, it is crucial to consider how external knowledge can be effectively integrated into models to support robust generalization across different domains.

III. METHOD

The knowledge graph is a collection of triples $(head\ entity, relation, tail\ entity)$, where the head entity and tail entity belong to set E, and the relation belongs to set R. E and R represent sets of entities and relations between entities, respectively. In this paper, P represents the set of paths, where p_i denotes the i-th path in set P, $len(p)$ denotes the length of path p, and p represents a relationship chain used to connect the head and tail entities, i.e., p_i= (r1, r2,…,rn), where p does not include entities. Our completion method aims to enhance the completeness of the knowledge graph by completing queries such as $(Jennifer\ Connelly, Nationality, ?)$, where the head entity and relation are provided, to infer the missing tail entity.

A. CBR MODULE

The method we propose differs from conventional case-based reasoning approaches by primarily enhancing the inference scheme and thereby improving the completeness of the knowledge graph through three aspects: expanding optional paths, strengthening entity representation, and updating path sorting methods. This section aims to elaborate on the case-based reasoning module, CBR-ADV, within our proposed completion framework. An overview of its architecture is presented in Figure 1, highlighting the functionality and working mechanism of this module within the overall framework.

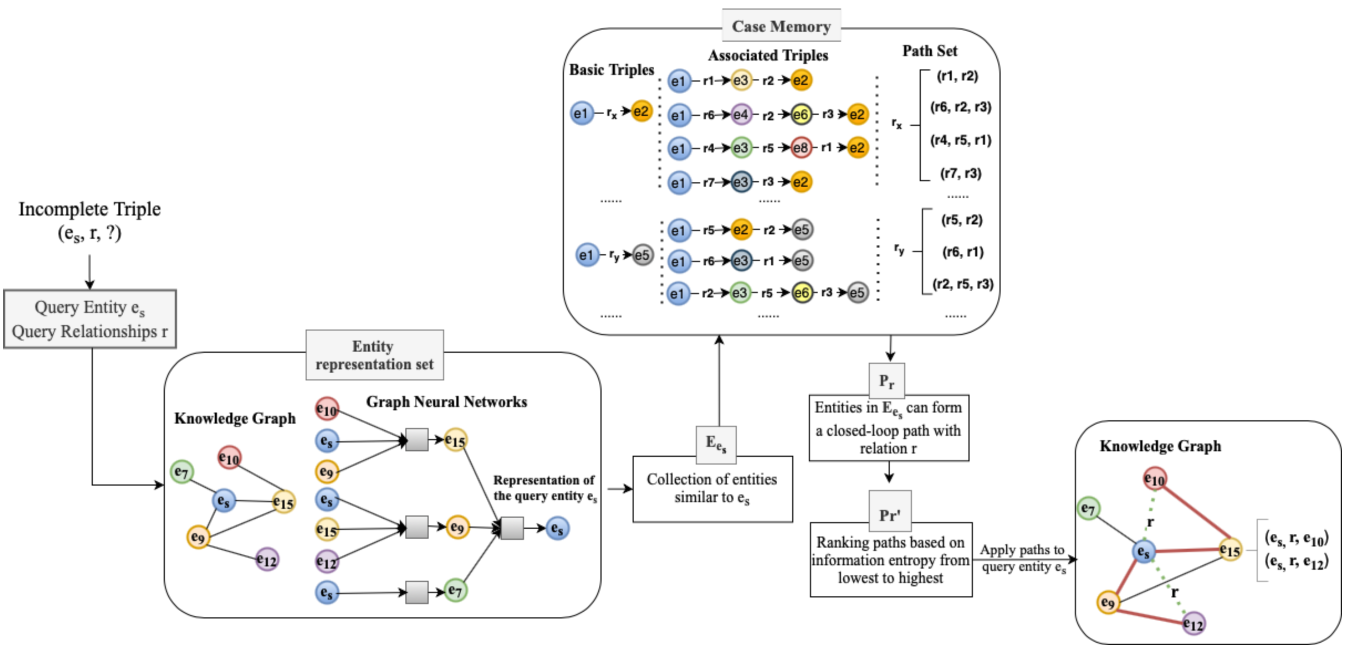


FIGURE 1. Overview of the CBR-ADV model. For a given head entity relationship pair (e_s, r), the entities are first represented by a graph neural network and then the set of entities most similar to the head entity E_{e_s} is retrieved as well as the set of related paths P_r is collected. Next, the collected paths are sorted in order of information entropy from low to high and these paths are used in turn on the head entity e_s until the answer entity e_p or answer entity collection E_{e_p} are reasoned.

1) RETRIEVAL STEP

The retrieval step aims to retrieve entity sets $E_{e_s} = \{e_1, e_2, \ldots e_n\}$, from the knowledge graph for a given entity relationship pair (e_s, r) that is most similar to entity e_s, where n is the maximum number of similar entities that can be found. It is also necessary to retrieve a set of paths $P_r = \{p_1, p_2, \ldots p_m\}$ that can be used to reason about the answer entity based on the relation r, where m is the maximum

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3505433

number of paths that can be found for reasoning about the answer entity. For more detail, the paths p_i in P_r are paths that can be collected from the knowledge graph G about similar entities e_i that can be used to form a closed loop with the relation r. For example, for the entity-relationship pair $(Jennifer\ Connelly, nationality)$, we first retrieve the similarity entity Celine Dion through the entity representation search and find the paths that can form a closed loop based on the relationship nationality, and obtain the triples (Celine Dion, place of birth, Quebec), (Quebec, affiliation, Canada), and (Celine Dion, husband, René Angélique), (René Angélique, Nationality, Canada). Ultimately, we store them as relational nationalities, and the corresponding collectible paths (birthplace, affiliation) and (husband, nationality). We maintain a case library as depicted in Figure 2, which stores the collection of paths obtained by finding closed paths for each entity and relation.

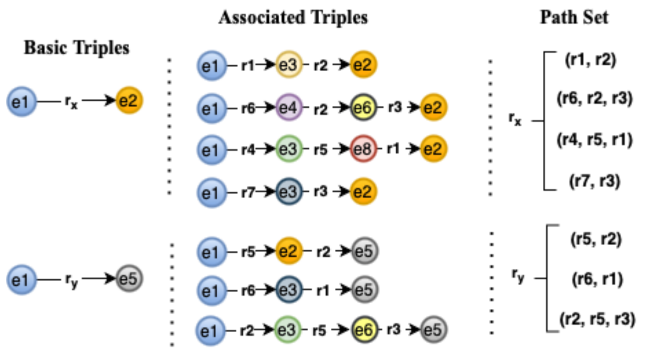


FIGURE 2. Case library, we perform an exhaustive search for each entity and its existent relations, for the triple (e_1, r_x, e_2), we store the paths of relations around the entity e_1 that can form a closed loop with e_2, that is (r_1, r_2)、(r_6, r_2, r_3)、(r_4, r_5, r_1)、(r_7, r_3).

In particular, for retrieving similar sets of entities, we consider using Relational Graph Convolutional Networks (R-GCN) to represent entities. R-GCN excels in learning entity and relation embeddings by fully considering the diverse types of relationships within the knowledge graph, thus effectively capturing and expressing its complex relational structures [58]. In R-GCN, information propagation follows a convolutional process, but the unique aspect lies in considering different relationship types. At each layer l, for each entity i, its embedding update is computed using (1).

$$\boldsymbol{h}_i^{(l+1)} = \sigma\left(\sum_{r\in R_i}\sum_{j\in N_i^r}\frac{1}{c_{i,r}}\boldsymbol{W}_r^{(l)}\boldsymbol{h}_j^{(l)} + \boldsymbol{W}_{self}^{(l)}\boldsymbol{h}_i^{(l)}\right) \tag{1}$$

Where R_i is the set of all relationships that entity i is involved in, N_i^r is the set of all neighboring entities connected to entity i through relationship r, $c_{i,r}$ is the normalization coefficient, σ is the activation function, $\boldsymbol{W}_r^{(l)}$ is the weight matrix corresponding to relationship r at the l-th layer, and $\boldsymbol{W}_{self}^{(l)}$ is the self-connection weight matrix used to retain node's own information. Relationship embeddings in R-GCN are implicitly represented by the weight matrix $\boldsymbol{W}_r^{(l)}$, which is optimized along with other parameters during training through backpropagation and gradient descent to minimize the overall model's loss function. The loss function is defined as in (2), where (h, r, t) represents a correct triple, P is the set of all positive triple samples, (h', r', t') is a negative sample triple obtained through random replacement of the head entity or tail entity, belonging to the negative sample set N. $f(h, r, t)$ is a scoring function used to compute the matching degree of a given triple, and γ is a margin value used to ensure that the score of a correct triple is higher than that of an incorrect triple. Ultimately, the embedding of each relationship r is actually reflected by its corresponding weight matrix $\boldsymbol{W}_r^{(l)}$, which controls the transformation of information specific to the relationship during propagation, thereby learning the characteristics of the relationship.

$$L = \sum_{(h,r,t)\in P}\sum_{(h',r',t')\in N}\max\left(0, f(h,r,t) + \gamma - f(h',r',t')\right) \tag{2}$$

Furthermore, on this basis, cosine similarity is utilized to measure the similarity between two entity vectors, as shown in (3), where $\boldsymbol{h_i}$ and $\boldsymbol{h_j}$ represent the embedding vectors of two entities, $\boldsymbol{h_i}\cdot\boldsymbol{h_j}$ denotes the dot product of the vectors, and $\|\boldsymbol{h_i}\|$ and $\|\boldsymbol{h_j}\|$ respectively represent the norms of the vectors, i.e., their lengths. By computing the cosine similarity between the target entity and all neighboring entities, and sorting them in descending order of similarity, a concise and precise set of similar neighbors can be obtained. This strategy aims to rigorously control the accuracy of the neighbor set from the initial stage, filtering out potential noisy paths, effectively suppressing the generation of inaccurate completion results, and enhancing the overall accuracy and reliability of the model.

$$Sim(\boldsymbol{h_i}, \boldsymbol{h_j}) = \frac{\boldsymbol{h_i}\cdot\boldsymbol{h_j}}{\|\boldsymbol{h_i}\|\|\boldsymbol{h_j}\|} \tag{3}$$

2) REUSE STEP

The reuse step aims at obtaining paths that can be used to reason about answer entities. We have retrieved some of the paths in the previous step but consider that some paths are semantically similar even though the composition of the relations between them is not the same. Therefore, we consider path semantics, where the paths obtained in the vicinity of similar entities are semantically matched with the existing paths of the query entity to obtain more valid inference paths. Through semantic matching of paths, we can extend the existing available paths by setting a threshold while trying not to be limited by the length of the paths. The method of semantic matching of paths can also be considered complementary to the existing method.

Firstly, we represent each path as a vector, as shown in (4).

$$\boldsymbol{P_i} = \sum_{k=1}^{n_i}\boldsymbol{r_k} \tag{4}$$

where $\boldsymbol{P_i}$ is the vector representation of path i, $\boldsymbol{r_k}$ is the embedding vector of the k-th relationship in the path, and n_i is the number of relationships in path i. Next, we can also compute the semantic similarity between two paths using vector-based cosine similarity, as shown in (5).

$$Sim(\boldsymbol{P_i}, \boldsymbol{P_j}) = \frac{\boldsymbol{P_i}\cdot\boldsymbol{P_j}}{\|\boldsymbol{P_i}\|\|\boldsymbol{P_j}\|} \tag{5}$$

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3505433

Finally, based on the computed semantic similarity, we set a threshold θ to decide which paths are similar enough and should be included in the corresponding case base, that is, if $Sim(P_i, P_j) \geq \theta$, then P_j is a candidate for reuse. By employing the above method, we can compare the paths around the queried entity with existing paths, filtering out more potentially useful paths and expanding the set of inference paths. By augmenting the original path set, we enhance the accuracy of the final answer inference.

In addition, since only the number of occurrences of a path is currently used as a single path ranking indicator, other uncertainties have not yet been considered. Therefore, the paths obtained by such methods and the answer entities obtained by reasoning also have some uncertainties. These uncertainties lead to the existence of noisy entities. Therefore, an ideal path that can accurately reason out the correct answer should be one that can exist stably and can better eliminate uncertainties [65]. Currently, information theory has been widely used to quantify the uncertainty of information [65-75]. Among them, information entropy has been recognized as a method that can effectively assess uncertainty [65, 76-80]. Therefore, for the ranking of paths consider the use of information entropy to further obtain quality paths by evaluating the uncertainty of the paths. According to [81-85] it can be concluded that the higher the uncertainty of the information, the higher its entropy and at the same time the lower the confidence level. More specifically, uncertainty implies that there are more factors affecting the stability of information, especially when the length of the path is longer, the more various information it carries with it, and the lower the usability of its path. In the context of filtering and sorting the paths, the paths with low entropy are first selected, and the answer entities obtained from the low entropy paths are taken as valid entities with a high accuracy rate. For the different paths obtained, their entropy is calculated using (6).

$$S_{P_i} = -\sum_{c=1}^{len(P_i)} f_{r_{P_{i_c}}} \log_2 f_{r_{P_{i_c}}} \tag{6}$$

where S_{P_i} is the entropy of the ith path, $len(P_i)$ is the length of the ith path, $r_{P_{i_c}}$ is the cth relation existing in the i-th path, and $f_{r_{P_{i_c}}}$ is the probability that relation $r_{P_{i_c}}$ exists in the knowledge graph g, that is, $f_{r_{P_{i_c}}} = \frac{n_{r_{P_{i_c}}}}{N}$ where $n_{r_{P_{i_c}}}$ is the total number of times that relation $r_{P_{i_c}}$ occurs in knowledge graph g, and N is the number of times that all the relations occur in knowledge graph g. Ultimately, after calculating the entropy of the obtained paths, the paths are ranked from low to high based on the entropy. The answer entities derived from paths with low entropy are considered as more secure entities.

3) COMPLETION STEP

In the completion step, we utilize the paths obtained above to match and reason along the paths near the queried entity e_s to derive the final potential answer entities.

Compared to traditional case-based reasoning (CBR) completion techniques, our CBR-ADV model introduces notable improvements in several key areas. First, entity representation has been strengthened. Traditional CBR methods often rely on similarity measures for case retrieval, which can be susceptible to noise and irrelevant entities. In contrast, our approach leverages the capabilities of graph neural networks, particularly the sensitivity of R-GCN (Relational Graph Convolutional Networks) to different relationships, enabling the effective selection of entities genuinely relevant to the query. This results in a more precise and concise set of similar entities, reducing interference from irrelevant information and providing a more accurate basis for subsequent completion tasks. Second, our approach expands the range of selectable paths. Traditional CBR methods are often limited by the finite paths in the case memory. We overcome this limitation by considering path semantics to match paths with similar meanings, significantly broadening the range of paths that can be reused for query entities. This method not only increases the potential for completion but also improves the model's adaptability to new entities. Finally, we have refined the path ranking method. Traditional CBR methods typically rely on simple statistical techniques for path ranking. We introduce the concept of information entropy, a well-established measure in information theory, to more accurately assess the uncertainty and information content of paths. By prioritizing paths with lower entropy, our approach more effectively filters out noisy paths and extracts key information-rich patterns, leading to more accurate completion results. These enhancements aim to improve the model's completion accuracy and generalization ability, making it more robust and adaptable when handling new entities and complex cases. Consequently, our approach advances the effectiveness of case-based reasoning in knowledge graph completion.

B. LLM MODULE

1) GENERAL STEP

In the process of completing knowledge graphs using large language models (LLMs), a series of steps is adopted to ensure the accuracy and effectiveness of the completion, as illustrated in Figure 3. First, the triples to be completed, represented as <subject, relation, ?>, are converted into natural language questions that LLMs can understand. For example, questions such as "Who directed the movie 'Oppenheimer'?", "Who stars in the movie 'Dune: Part Two'?", or "Who is the author of the original novel for the movie 'Wonka'?" are formulated. This conversion process aims to leverage the semantic understanding and generation capabilities of LLMs, enabling them to infer the missing information in the triples.

In the specific implementation, each element of the triples undergoes a targeted mapping process. The subject is mapped to a specific entity name in natural language, while the relation is transformed into an inquiry format regarding that entity. Each triple <subject, relation, ?> is formatted into the structure "[who/what] + [relation description] + [subject name]?", facilitating the LLM's quick recognition of the question's core. For instance, the triple <Oppenheimer, director, ?> is converted to "Who directed the movie 'Oppenheimer'?", and the triple <Dune: Part Two, starring, ?> is transformed into "Who stars in the movie 'Dune: Part Two'?". This structured conversion not only enhances the LLM's comprehension abilities but also allows for efficient generation of expected answers.

Next, the constructed questions are encoded using the encoder module of the LLM to capture their contextual dependencies and semantic features. Subsequently, the decoder generates representations of candidate answers based on the encoded information. These answers may include movie entities from the knowledge graph, relevant relations, or other information related to the film domain. For example, the model may produce answers such as "Christopher Nolan directed 'Oppenheimer'," "Timothée Chalamet stars in 'Dune: Part Two'," or "Roald Dahl wrote the original novel for 'Wonka.'"

After generating candidate answers, particular attention is given to the impact of sentence structure similarity on the model's outputs. Despite the potential for natural language questions to share similar structural forms, the LLM can rely on its inherent attention mechanism and contextual capturing abilities to identify and extract the key elements of the questions, ensuring accuracy and consistency. For instance, for structurally similar inquiries such as "Who stars in the movie 'Dune: Part Two'?" and "Who stars in the movie 'Dune'?", the model generates correct answers based on differing contextual features. In this process, the attention mechanism can not only measure the semantic relevance between candidate answers and the input question context, but also dynamically adjust the information focus during the model generation process. Therefore, the interference of sentence similarity on the model output results is significantly reduced, further ensuring the reliability and pertinence of the answers. Therefore, in the scoring stage, we combine the attention weights of the large language model to quantify the relevance between candidate answers and the context of the question, in order to ensure that the selected answers are highly semantically aligned with the subject and relationship of the input question. This mechanism effectively reduces the interference of sentence similarity on the output results, further improving the accuracy and consistency of generated answers.

In summary, through the natural language conversion of triplets, effective utilization of attention mechanisms, and semantic relevance assessment during the scoring phase, our completion method not only enhances its semantic understanding ability, but also improves its robustness in dealing with sentence structure similarity interference, providing solid technical support for high-quality completion of knowledge graphs. This strategy ensures that the generated answers are grammatically and semantically reasonable, thereby improving the overall accuracy and reliability of knowledge graph completion.

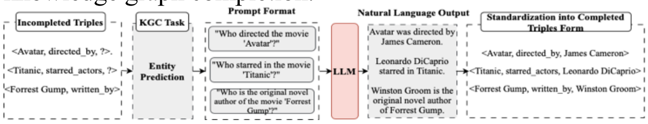


FIGURE 3. Illustration of Movie Knowledge Graph Completion Based on Large Language Models. For example, given the head entity "Avatar" and the relation "directed_by", the predicted tail entity is "James Cameron". Finally, the output natural language answer is "Avatar was directed by James Cameron", which is then normalized into the desired triple format as <Avatar, directed_by, James Cameron>.

2) UPDATE STEP

To address the inherent limitations of large language models in processing information about newly released films, while significantly reducing update costs, we have implemented a fine-tuning strategy. Specifically, when completing newly generated film entities, data injection is necessary. This involves transforming new data—such as detailed information about newly released films, including reviews, news reports, web pages, and posters—into a format suitable for model learning and integrating it as additional training data into the existing language model. By incorporating this new data, we provide the model with up-to-date knowledge on newly released films, filling gaps that may exist in its original training dataset. To ensure the quality and accuracy of this new data, all information obtained from external sources is rigorously preprocessed and individually verified. This thorough data validation process maximizes the accuracy and reliability of the input, minimizing the risk of misleading outputs caused by low-quality data. Subsequently, we apply fine-tuning techniques to iteratively update portions of the model's parameters based on the injected data. This approach enables our model to retain its deep understanding of previously known film knowledge while gradually acquiring the ability to process new film information. The detailed process is depicted in Figure 4.

After fine-tuning, the model is able to continue answering questions about known films while also acquiring the ability to understand and generate information related to newly released films due to targeted parameter adjustments. This allows for stable completion of historical film knowledge and quick responses to new film information during subsequent knowledge graph completion tasks. Consequently, the model significantly enhances the timeliness, generalization, and adaptability of film knowledge graph completion tasks, while avoiding the substantial time costs and computational resource demands associated with retraining the entire model.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3505433

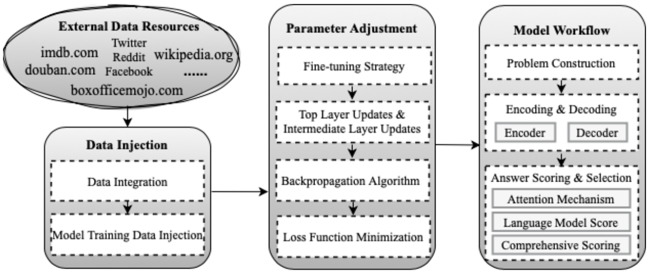


FIGURE 4. Overview of the method. We start by identifying suitable movie data sources, which are then injected as additional training data into the existing large language model. Subsequently, we iteratively update certain parameters of the model through fine-tuning techniques. Following fine-tuning, the model gains enhanced understanding and generation capabilities regarding new movie information. Our approach enhances timeliness and adaptability while avoiding the need for full retraining.

To provide a clearer illustration of the overall architecture of the proposed CBR-LLM model, Figure 5 presents the framework, which consists of three main components: the Case-Based Reasoning (CBR) module, the Large Language Model (LLM) module, and the output integration module. The CBR module retrieves the most similar entity set to the given entity using a graph neural network, while also collecting semantically relevant closed paths and ranking them by information entropy to identify the most valuable paths for completion. The LLM module converts the triples to be completed into natural language questions and leverages the large language model's advanced semantic understanding and generation capabilities to produce candidate answers. Additionally, a fine-tuning strategy is incorporated to ensure the model quickly adapts to new entity information. The output integration module combines the completion abilities of the large language model with the case-based reasoning approach to improve the accuracy and comprehensiveness of the final results. To optimize the synergy between these two methods, a confidence-weighting approach is used to balance the contributions of both methods, as demonstrated in Equation (7).

$$F_{conf} = \alpha \cdot R_{conf} + \beta \cdot M_{conf} \quad (7)$$

In this framework, the confidence derived from case-based reasoning is denoted as R_{conf}, while the confidence predicted by the large language model is denoted as M_{conf}. The weights α and β (where $\alpha,\ \beta \in [0,1]\ and\ \alpha + \beta = 1$) determine the contribution of each method to the final result. Adjusting α and β allows for the control of each method's influence on the final outcome.

C. INTERPRETABILITY

The interpretability of the CBR-LLM model is particularly important in practical applications, especially when the model is used for knowledge graph completion tasks. A transparent reasoning process and clear decision-making basis can help users understand the model's behavior and increase their trust in the model's predictions. The following sections discuss the interpretability of the CBR module, the LLM module, and their integration.

1) INTERPRETABILITY OF THE CBR MODULE

The CBR module completes missing information in the knowledge graph by retrieving similar cases from the knowledge base and reasoning based on these cases. This process exhibits high interpretability because it relies on known historical cases, clearly illustrating the reasoning pathways and logical chains. The interpretability of the CBR module can be elaborated upon from three aspects: similarity retrieval, path reasoning, and result reuse.

First, consider the interpretability of similarity retrieval. The CBR module encodes the representation of the query entity using a graph neural network and retrieves similar entities from the knowledge graph. The interpretability of this retrieval process stems from the similarity measures, where the model calculates similarity based on specific features of the entities. Users can clearly understand why the model selects these entities as references for reasoning by examining the similar entities and their attributes. This transparency allows users to trace each step of the reasoning process and validate the model's rationale.

Next, the interpretability of path reasoning is considered. After retrieving similar entities, the CBR module performs reasoning based on the historical paths associated with these entities. The logic behind each path is clearly visible, as they demonstrate the relationships between entities and target relations. To enhance the interpretability of path selection, the CBR module employs information entropy calculations to quantify the uncertainty of each path. A lower entropy value indicates reduced uncertainty and higher confidence in the path. Consequently, users can understand why the model prioritizes a particular path as the basis for final reasoning by examining the ranking of entropy values. This mechanism provides clearer explanatory grounds for the reasoning process.

Finally, the interpretability of result reuse is addressed. The CBR module reuses derived paths to infer predictions for the query entities. In this process, the model relies not only on historical paths but also incorporates the confidence levels of these paths to make final inferences. This approach allows users to clearly understand each step of the reasoning chain, including the selection of similar entities, the utilization of paths, and the logical derivation of the final results. The mechanism of history-based reasoning in the CBR module ensures transparency in its reasoning process, enabling users to trace and validate each decision made by the model.

2) INTERPRETABILITY OF THE LLM MODULE

The interpretability of the LLM module primarily manifests in the prompt construction and semantic understanding processes. This module converts structured data into natural language questions and utilizes large language models to generate candidate answers. During this process, the design of the prompts plays a crucial role in determining the quality of the model's output. Although the internal reasoning mechanisms of LLMs are complex, the answers generated can be explained

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3505433

through prompt design. By adjusting prompts or modifying the structure of questions, users can gain insights into the LLM's reasoning process to some extent. Therefore, the interpretability of the LLM module focuses on how to design and optimize prompts to influence the accuracy and relevance of the model's outputs. Although this process is somewhat abstract, results can be validated by adjusting input conditions.

3) OVERALL INTERPRETABILITY OF THE MODEL

The overall interpretability of the CBR-LLM model is enhanced through the integration of the two modules. Specifically, the CBR module provides a transparent reasoning framework that allows users to clearly understand each step of the reasoning process, including similarity retrieval, path reasoning, and result reuse. The LLM module further complements the CBR module's reasoning by generating natural language outputs, providing semantic support for the final predictions.

To further enhance the model's interpretability, the CBR-LLM model employs a confidence-weighting method to integrate the predictions of both modules. Specifically, the CBR module calculates the confidence of each path based on the retrieved paths and ranks these paths using information entropy, while the LLM module generates candidate answers based on the prompts. Ultimately, the model combines the confidences from both CBR and LLM, resulting in a comprehensive decision-making approach. This confidence-weighting method not only allows for effective integration of predictions from both modules but also provides a clear explanation of the generation process of the final predictions. Users can evaluate the accuracy of predictions by examining the confidence levels of each candidate answer, thereby understanding why the model selects a particular result.

In summary, the CBR-LLM model ensures high interpretability in knowledge graph completion tasks through the transparent reasoning process of the CBR module, the semantic generation capabilities of the LLM module, and the integration mechanism of the confidence-weighting method. This enables users to gain a deeper understanding of the model's reasoning process and the key factors influencing the prediction results.

To illustrate the completion process of the CBR-LLM model, we provide a specific example. Given a knowledge graph that needs completion, consider a missing triple such as <Jennifer Connelly, Nationality, ?>, where Jennifer Connelly is the query entity and Nationality is the query relationship.

The CBR module retrieves a set of entities similar to Jennifer Connelly using a graph neural network. In this study, we use R-GCN to represent entities, identifying a set of similar entities, including Céline Dion and Sean Penn. We then search for paths that form a closed loop with the relationship Nationality for these entities. For Céline Dion, we find closed-loop paths such as <Celine Dion, place of birth, Quebec>, <Quebec, affiliation, Canada> and <Celine Dion, husband, René Angélique>, <René Angélique, Nationality, Canada>. We evaluate the uncertainty of each path using information entropy. Based on the resulting triples, we obtain the closed paths <birthplace, affiliation> and <husband, nationality>, and rank them by entropy from low to high. The path <birthplace, affiliation> has lower entropy and higher confidence, indicating a higher probability that the inferred tail entity from this path is the correct answer. We reuse this path for the query entity Jennifer Connelly. By prioritizing the path with lower entropy <birthplace, affiliation>, we derive the CBR module's inferred completion result <Jennifer Connelly, Nationality, American>. Similarly, using the path with the second lowest entropy <husband, nationality>, we obtain another completion result <Jennifer Connelly, Nationality, British>.

The LLM module begins by formulating a natural language question from the triple <Jennifer Connelly, Nationality, ?>, specifically, "What is Jennifer Connelly's nationality?" This question is then input into the large language model, which encodes it to capture contextual dependencies and semantic features. The model generates the candidate answer statement: "Jennifer Connelly's nationality is American," which, after normalization, yields the triple <Jennifer Connelly, Nationality, American>.

The output integration module combines the candidate answers from the CBR module, which are "American" with a confidence of 0.87 and "British" with a confidence of 0.13. The LLM module provides a single answer "American" based on the current prompt, resulting in $CBR_{\text{American}} = 0.87$, $CBR_{British} = 0.13$, and $LLM_{American} = 1$. The probabilities of each candidate answer being correct are computed by integrating the answers from both modules, with balancing coefficients α and β set at 0.33 and 0.67, respectively. This results in $P_{\text{American}} = 0.33 * 0.87 + 0.67 * 1 = 0.9571$, $P_{British} = 0.33 * 0.13 + 0.67 * 0 = 0.0429$. Given that the Nationality relationship expects a unique tail entity, the entity with the highest integrated probability, American, is selected as the final answer. Consequently, the normalized triple is <Jennifer Connelly, Nationality, American>.

The above steps demonstrate how the CBR-LLM framework integrates case-based reasoning path information with the semantic understanding capabilities of the large language model to complete missing information in the knowledge graph. In this example, we accurately determined that Jennifer Connelly's nationality is American and combined the predictions from both modules using confidence weighting, leading to the final completion answer.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3505433

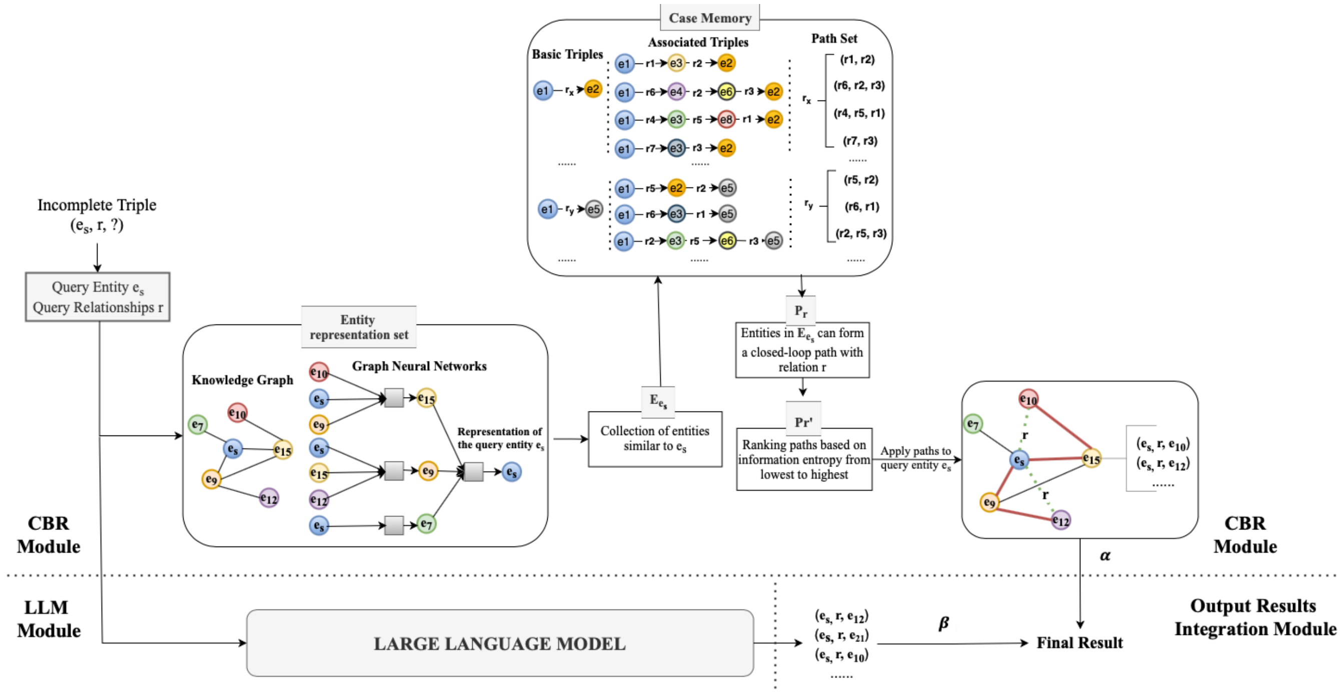


FIGURE 5. **CBR-LLM Framework Diagram. It illustrates the overall architecture of the CBR-LLM framework, including the CBR module, the LLM module, and the output result integration module. By assigning appropriate weights to the candidate answer entities obtained from the CBR and LLM modules, the predictions from both modules are integrated to derive a more accurate final answer entity.**

IV. EXPERIMENTS

In this section, we evaluate the performance of the proposed completion framework. We first introduce the datasets and baseline models used for evaluation, and then analyze and discuss the experimental results.

A. DATASETS

Our research utilizes the MetaQA movie knowledge graph [86], with necessary data cleaning to remove redundant triples. We also integrated triples for the latest movies (MetaQA-Latest Additions) to assess the effectiveness of the CBR-LLM model in reasoning about new entities. We conducted an in-depth analysis of the CBR-LLM model's performance on both the MetaQA benchmark dataset and its latest extension (MetaQA-Latest Additions), providing a thorough evaluation through detailed results. Additionally, we validated our completion framework's effectiveness by performing the same experiments on two widely used datasets: FB15k-237N [87] and NELL-995 [88]. FB15k-237N is an extended knowledge graph dataset based on the original FB15k-237, designed to test the model's capability to handle ambiguous entities and to enhance the dataset's diversity and complexity by adding additional entities and relationships. NELL-995, derived from the Never-Ending Language Learning (NELL) project, is meticulously curated and includes 995 categories of relational instances. Its high quality and comprehensive entity-relation information make it a widely used benchmark in academic research.

B. SETTINGS

1) BASELINE METHODS

In our experimental design, we systematically compared the performance of the proposed method with a carefully selected set of baseline models to comprehensively validate its effectiveness in knowledge graph completion tasks. The choice of baseline models covered core dimensions, aiming to explore the innovative value and practicality of our method through multidimensional comparisons. We included classic knowledge graph embedding algorithms such as RESCAL [12], DistMult [89], and ComplEx [90], which focus on representing entities and relations in low-dimensional vectors and mapping the structure of the knowledge graph to continuous space, laying the foundation for completion tasks. To contrast with large language models, we directly employed information-guided large models to perform completion tasks on movie knowledge graphs as a comparison baseline. Through these multidimensional comparisons, our aim was to comprehensively analyze the unique features of the proposed method and its performance advantages in practical tasks.

2) DETAIL SETTINGS

In developing the completion framework, we employed R-GCN embeddings with a dimensionality of 512 and set the θ value to 0.85. During the integration of the CBR and LLM modules, we performed a grid search to meticulously adjust the parameter settings for optimal performance. We tested various weight combinations to determine which configuration provided the best overall results. We determined that the optimal weight ratio for α and β was 0.33 and 0.67, respectively, based on experimental performance. For the large language model, we selected the powerful LLaMA-13B as the base model and performed efficient fine-tuning using the LoRA method with a rank of 64. This approach aimed to harness the LLM's strong generative capabilities alongside the CBR module's case-based reasoning strengths, ensuring effective collaboration between the two in managing new entities and long-tail data.

C. Result and Discussion

Figure 6 displays the hits@1 results for various models in the knowledge graph completion task on the MetaQA dataset. The experiments demonstrate that the CBR-LLM model significantly outperforms other models under both 50% and 30% data missingness conditions. Notably, under the 50% data missingness scenario, the hits@1 score of the CBR-LLM model is markedly higher, reflecting its robustness and generalization capabilities in large-scale knowledge graph completion tasks.

In comparison to traditional knowledge graph embedding models, RESCAL typically performs well in relatively complete data environments. However, its adaptability to complex relationships and high data missingness rates is limited, resulting in a significant decline in performance under high missingness. This limitation arises because RESCAL relies on a fixed tensor decomposition method, which only functions effectively with structurally complete data and struggles to maintain a accurate predictions of entity’s relationships in high missingness situations. In contrast, the CBR-LLM model can flexibly integrate historical cases and generate completion information through a large language model, thereby better addressing the issue of incomplete information due to data missingness. DistMult and ComplEx exhibit some stability under missingness conditions. However, their symmetrical and fixed model structures hinder their ability to complete for missing information as the data missingness increases. These models are less effective than CBR-LLM in relationship modeling because they rely on a single representation vector for each pair of entity relationships, lacking the ability to capture and reason about contextual information. By combining structured reasoning from the CBR module with semantic generation from the LLM module, the CBR-LLM model better captures and completes missing data, demonstrating significant advantages under high missingness conditions.

For the CBR-ADV model, although its overall performance is inferior to that of CBR-LLM, it achieves higher hits@1 scores than traditional embedding models in the presence of data missingness. CBR-ADV leverages case-based reasoning mechanisms during inference, allowing it to partially complete for missing data based on available information, thus exhibiting robustness under these conditions. However, due to the absence of semantic support from the LLM module, CBR-ADV is unable to extract additional information from complex semantics, limiting its ability to complete intricate relationships. Consequently, the CBR-LLM model surpasses CBR-ADV in completing complex relationships and uncovering deeper semantic information, showcasing stronger reasoning capabilities.

While the performance of the LLM model under 50% and 30% data missingness is slightly lower than that of CBR-LLM, its hits@1 scores still exceed those of traditional embedding models. This indicates that large language models assist in capturing semantic and contextual information in completion tasks, particularly within environments characterized by data missingness in large-scale knowledge graphs. Notably, the LLM's strong performance in completion tasks partly arises from its pre-training on extensive datasets, which cultivates rich contextual understanding and language generation abilities. This pre-training not only enhances the LLM's capacity to capture semantics but also improves its generalization under conditions of knowledge graph data missingness, providing robust semantic support for the advantages of the CBR-LLM model. However, the drawback of solely utilizing the LLM lies in its lack of structured reasoning, making it challenging to deeply explore the graph structure of knowledge graphs, especially when completing complex relationships that require structured reasoning. In such cases, the performance of the LLM model falls short compared to that of CBR-LLM. This further emphasizes that CBR-LLM has a clear advantage by integrating structured reasoning from CBR with semantic understanding from LLM in scenarios of data missingness.

In summary, the CBR-LLM model demonstrates significant advantages in knowledge graph completion tasks, particularly under high data missingness conditions. This advantage stems from its combination of structured reasoning from the CBR module and semantic generation capabilities from the LLM module, effectively integrating predictions from both modules through a confidence-weighting method, thus achieving complementary semantic and structural information. This integration enables CBR-LLM to exhibit exceptional robustness and generalization across various data missingness scenarios.

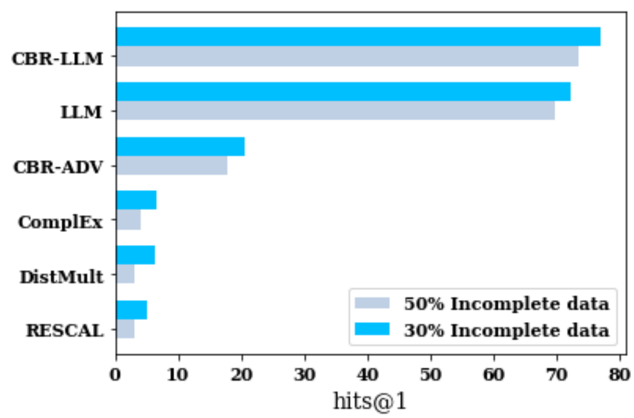


FIGURE 6. The hits@1 outcomes for knowledge graph completion across various models, as assessed on the MetaQA dataset.

TABLE I

THE MAIN EXPERIMENTAL RESULTS OF THE COMPLETION TASK.

Dataset / Model	MetaQA		MetaQA-Latest additions		NELL-995		FB15K-237N	
	Hits@1	*Hits*@5	*Hits*@1	*Hits*@5	*Hits*@1	*Hits*@5	*Hits*@1	*Hits*@5
LLM	72.3	77.1	65.2	67.8	50.1	54.3	60.1	63.2
Our (Fine-tuned LLM)	74.2	78.9	70.9	73.4	55.8	59.7	65.7	68.7
Our (CBR + Fine-tuned LLM)	77.1	80.7	72.4	75.7	59.4	63.5	67.8	70.3

Table 1 displays the main experimental results of various models on the knowledge graph completion task across the MetaQA, MetaQA-Latest Additions datasets, and the commonly used NELL-995 and FB15K-237N datasets. The results highlight the significant advantages of our approach. On the MetaQA dataset, the unfinetuned LLM achieves Hits@1 and Hits@5 scores of 72.3% and 77.1%, respectively. After fine-tuning, these scores increase to 74.2% and 78.9%, demonstrating that fine-tuning enhances the LLM's ability to learn implicit patterns in the dataset and improves its performance on knowledge graph completion tasks. Integrating CBR with the fine-tuned LLM yields Hits@1 and Hits@5 scores of 77.1% and 80.7%, respectively, which are improvements of 2.9% and 1.8% over the fine-tuned LLM alone. This improvement indicates that CBR effectively addresses the limitations of large language models in managing new entities and complex reasoning paths, enhancing overall model performance. For the MetaQA-Latest Additions dataset, the unfinetuned LLM has Hits@1 and Hits@5 scores of 65.2% and 67.8%, respectively. Post-fine-tuning, these scores rise to 70.9% and 73.4%, further validating the benefits of fine-tuning. Our method achieves Hits@1 and Hits@5 scores of 72.4% and 75.7%, respectively, surpassing the fine-tuned LLM by 1.5% and 2.3%. These gains underscore the effectiveness of our approach in handling new entities, especially in tasks requiring extensive reasoning and completion, where the CBR module significantly improves model accuracy.

For the NELL-995 dataset, the unfinetuned LLM achieves Hits@1 and Hits@5 scores of 50.1% and 54.3%, respectively. After fine-tuning, these scores improve to 55.8% and 59.7%. Our method further enhances these scores to 59.4% and 63.5%, demonstrating superior performance in capturing hidden relationships and performing effective completion in knowledge graphs with complex relationships and substantial data noise. On the FB15K-237N dataset, the unfinetuned LLM scores 60.1% for Hits@1 and 63.2% for Hits@5. After fine-tuning, these scores rise to 65.7% and 68.7%. By incorporating the CBR module, our method achieves Hits@1 and Hits@5 scores of 67.8% and 70.3%, respectively, showing a clear advantage. This indicates that our approach not only excels on standard datasets but also delivers improved results on more challenging datasets.

In conclusion, our method—combining CBR with a fine-tuned LLM—leverages the strong generative capabilities of large language models and the classical CBR approach to achieve a synergistic effect in handling complex reasoning and long-tail entities. As a result, our method delivers the best performance across all tested datasets, confirming its effectiveness and wide applicability in knowledge graph completion tasks.

TABLE I I
THE RESULTS OF THE COMPLETION EXPERIMENTS OBTAINED BY REPLACING THE THREE PARTS OF THE CBR MODEL, NAMELY THE PATH EXPANSION, ENTITY REPRESENTATION ENHANCEMENT, PATH RANKING UPDATES, IN TURN. THE PART THAT IS NOT REPLACED IS THE PART THAT WAS INITIALLY BASED ON CASE-BASED REASONING.

Dataset / **Model**	MetaQA		MetaQA-Latest additions	
	Hits@1	*Hits*@5	*Hits*@1	*Hits*@5
LLM	72.3	77.1	65.2	67.8
Path Expansion + LLM	72.7	77.8	66.5	68.4
Entity Representation Enhancement + LLM	73.3	78.4	67.8	68.9
Path Ranking Updates + LLM	74.3	78.9	68.3	69.7
Our (Fine-tuned LLM)	74.2	78.9	70.9	73.4
Our (CBR + Fine-tuned LLM)	77.1	80.7	72.4	75.7

Additionally, to investigate the reasons behind the effectiveness of the CBR-LLM framework and the necessity of each component within the CBR module, we performed ablation studies on the movie knowledge graph. The results of these experiments are detailed in Table 2. We introduced improvements such as the expansion of optional paths, the enhancement of entity representation, and the update of the path ordering method in turn. Among them, "Path Expansion + LLM", "Entity Representation Enhancement + LLM" and "Path Ranking Updates + LLM" represent the experimental results of the large language model with the introduction of a single improvement measure, respectively. For further comparison, we list Our (Fine-tuned LLM) and Our (CBR + Fine-tuned LLM). As can be seen from the experimental results in Table 2, the overall completion effect improves with the introduction of each improvement. Specifically, on the MetaQA dataset, the base LLM model achieves Hits@1 and Hits@5 scores of 72.3% and 77.1%, respectively. Upon incorporating Path Expansion, slight enhancements are observed in both Hits@1 and Hits@5, which ascend to 72.7% and 77.8%, respectively. This outcome signifies that Path Expansion can contribute to a modest improvement in the model's proficiency. However, the improvements imparted by Entity Representation Enhancement are more pronounced, elevating Hits@1 and Hits@5 to 73.3% and 78.4%, respectively, suggesting that the enriched entity representations more effectively capture the most pertinent neighboring entities. Further, Path Ranking Updates propel model performance even higher, lifting Hits@1 to 74.3% and Hits@5 to 78.9%, illustrating how a refined path ranking methodology more efficiently filters out noisy paths and extracts informative ones, thereby markedly enhancing response accuracy. The LLM model, when subjected to fine-tuning (Fine-tuned LLM), attains comparable results with Hits@1 and Hits@5 scores of 74.2% and 78.9%, respectively. Ultimately, the CBR combined with the Fine-tuned LLM

(CBR + Fine-tuned LLM) achieves the pinnacle of performance, registering Hits@1 and Hits@5 scores of 77.1% and 80.7%. These findings underscore that the synergistic application of optional path expansion, strengthened entity representation, and refined path ranking strategies collectively leads to significantly enhanced knowledge graph completion outcomes.

In the MetaQA-Latest Additions dataset, the baseline LLM model achieves Hits@1 and Hits@5 rates of 65.2% and 67.8%, respectively. With the introduction of various enhancement techniques, the model's performance improves progressively. Path extension techniques increase Hits@1 and Hits@5 to 66.5% and 68.4%, respectively; entity representation enhancement further elevates these metrics to 67.8% and 68.9%. The introduction of path ranking updates results in Hits@1 and Hits@5 rising to 68.3% and 69.7%. Similarly, the LLM model, synergistically fine-tuned with CBR and various techniques, achieves the best performance on this dataset, with Hits@1 and Hits@5 rates reaching 72.4% and 75.7%, respectively.

Overall, the enhancement techniques demonstrate a consistent trend in improving model performance across both datasets, though the improvements are relatively smaller on the MetaQA-Latest Additions dataset. This is primarily due to the inclusion of more new movie entities, which increases the difficulty for the model. In summary, the synergistic use of our proposed methods, coupled with a fine-tuned LLM model, significantly outperforms other methods on both datasets, indicating that our combined approach better addresses the completion tasks. In summary, the improvement in CBR-LLM completion ability primarily relies on three aspects. First, the CBR module expands the optional paths, removing the limitation on path length and providing a greater number of potential paths, thereby increasing the possibility of identifying the correct answer entities. Second, the approach retrieves the most similar context for each entity and relation pair within the knowledge graph, maximizing the consideration of contextual information and ensuring that the final answer entities are accurate, reliable, and verifiable. Finally, the use of established information entropy for path ranking further enhances the accuracy of the final answer entities.

To better understand the advantages of our approach, it is necessary to conduct a detailed comparison with other traditional models in terms of complexity. As one of the earliest proposed tensor decomposition models, RESCAL's parameter count primarily depends on the embedding dimension and the size of the relation matrices. DistMult, a simplified version of RESCAL, significantly reduces the number of parameters by eliminating bilinear operations, thereby decreasing computational complexity. For the same embedding dimension, DistMult requires fewer parameters compared to RESCAL. ComplEx extends DistMult by using complex embeddings, which leads to a similar parameter count. However, due to the inclusion of both real and imaginary components, ComplEx's total parameter count is approximately double that of DistMult. In contrast, our model, LLaMA-13B, contains a large number of parameters, reaching tens of billions or even hundreds of billions. After fine-tuning, LLaMA-13B maintains a similar parameter count, but its adaptability to specific tasks is improved. Finally, our proposed CBR+LLM method adds a CBR module on top of the LLM, with parameters primarily derived from embeddings and similarity calculations in the case base. Although this adds some computational overhead, the overall parameter scale remains comparable to the LLM alone, keeping the parameter count within the range of billions.

Additionally, traditional models typically rely on matrix factorization and multiplication operations during inference, making inference time proportional to the embedding dimension and dataset size. For example, on our knowledge graph dataset, DistMult achieves millisecond-level inference speed, demonstrating high computational efficiency. However, LLM models, due to their large number of parameters and more complex reasoning paths, take longer for inference. Under the same experimental conditions, inferring a single triple may take several seconds. Although fine-tuning can slightly accelerate this process, the overall inference speed remains slow. To address this issue, our proposed CBR+LLM method introduces the CBR module, which leverages fast retrieval of similar cases and experience-based reasoning to assist the LLM, thereby reducing inference time. In our experiments, the CBR+LLM method achieved stable inference times between a few hundred milliseconds to one second, significantly outperforming the unoptimized LLM model. This indicates that while the CBR module introduces additional computational overhead, the resulting improvement in inference efficiency leads to a more balanced performance in terms of both parameter count and inference time.

In summary, the CBR+LLM method outperforms traditional models and standalone LLM models in both parameter count and inference time, demonstrating superior performance balance. This result further validates the effectiveness and practicality of the CBR+LLM framework in knowledge graph completion tasks.

V. CONCLUSION

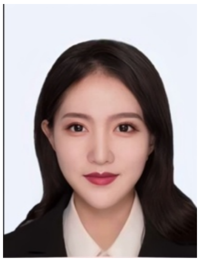
This paper tackles the challenges in knowledge graph completion tasks, including excessive reliance on manual effort, high computational costs, and poor scalability, by innovatively proposing a method that combines non-parametric case-based reasoning (CBR) with large language models (LLMs), termed CBR-LLM. This approach aligns with current technological trends, leveraging the strengths of LLMs in natural language understanding and generation. By integrating the practicality of case-based reasoning with the advanced capabilities of LLMs, the final results are not only accurate but also exhibit high semantic consistency. The method effectively enhances the model's ability to handle new entities and demonstrates exceptional performance in experimental evaluations, reflecting high competitiveness and robustness. Unlike traditional CBR methods, our approach improves scalability and flexibility through enhanced entity representation, expanded optional paths, and updated path sorting methods. This collaborative approach ensures that every step, from initial reasoning to final validation, is completed efficiently and accurately. Ultimately, the effectiveness of the CBR-LLM framework in practical knowledge graph completion tasks has been validated. The experimental results demonstrate the method's significant value in practical applications. Both theoretical support and practical outcomes highlight the important prospects of the CBR-LLM framework in the field of knowledge graph completion. Future research will continue to explore strategies for knowledge graph integrity completion, optimizing the balance between resource utilization, model scalability, and completion accuracy. Specifically, the focus will be on the implementation of the CBR-LLM model in large-scale domain knowledge graphs and further developing information integration mechanisms to enhance the model's cross-domain adaptability and improve completion precision.

REFERENCES

[1] Tari, L. (2013). Knowledge inference. Encyclopedia of Systems Biology, 1074–1078.
[2] Chen, X., Jia, S., & Xiang, Y. (2020). A review: Knowledge reasoning over knowledge graph. Expert Systems with Applications, 141, 112948.
[3] Galárraga, L. A., Teflioudi, C., Hose, K., & Suchanek, F. (2013). Amie: Association rule mining under incomplete evidence in ontological knowledge bases. In Proceedings of the 22nd international conference on world wide web (pp. 413–422). ACM.
[4] Galárraga, L., Teflioudi, C., Hose, K., & Suchanek, F. M. (2015). Fast rule mining in ontological knowledge bases with AMIE+. The VLDB Journal-The International Journal on Very Large Data Bases, 24(6), 707–730.
[5] Wang, Z., & Li, J. (2015). Rdf2rules: learning rules from rdf knowledge bases by mining frequent predicate cycles. arXiv:1512.07734
[6] Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Betteridge, J., Carlson, A., et al. (2015). Never-ending learning. In Proceedings of the twenty-ninth aaai conference on artificial intelligence (pp. 2302–2310). AAAI Press.
[7] Wang, W. Y., Mazaitis, K., Lao, N., & Cohen, W. W. (2015). Efficient inference and learning in a large knowledge base. Machine Learning, 100(1), 101–126.
[8] Lao, N., & Cohen, W. W. (2010). Relational retrieval using a combination of path–constrained random walks. Machine Learning, 81(1), 53–67.
[9] Gardner, M., & Mitchell, T. (2015). Efficient and expressive knowledge base completion using subgraph feature extraction. In Proceedings of the 2015 conference on empirical methods in natural language processing (pp. 1488–1498).
[10] Gardner, M., Talukdar, P., Krishnamurthy, J., & Mitchell, T. (2014). Incorporating vector space similarity in random walk inference over knowledge bases. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 397–406).
[11] Liu, Q., Han, M., Jiang, L., Liu, Y., & Geng, J. (2017b). Two-tier random walk based relational inference algorithm. Chineses Journal of Computers, 40(6), 1275–1290.
[12] Nickel, M., Tresp, V., & Kriegel, H.-P. (2011). A three-way model for collective learn- ing on multi-relational data.. In Icml: 11 (pp. 809–816).
[13] Nickel, M., Tresp, V., & Kriegel, H.-P. (2012). Factorizing yago: scalable machine learning for linked data. In Proceedings of the 21st international conference on world wide web (pp. 271–280). ACM.
[14] Nickel, M., & Tresp, V. (2013). Logistic tensor factorization for multi-relational data. In Proceedings of the 30th international conference on machine learning-volume 28. JMLR. org.
[15] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In Advances in neural in- formation processing systems (pp. 2787–2795).
[16] Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014c). Knowledge graph embedding by translating on hyperplanes. Twenty-eighth aaai conference on artificial intelligence.
[17] Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015b). Learning entity and relation embeddings for knowledge graph completion. Twenty-ninth AAAI conference on artificial intelligence.
[18] Jiang, T., Liu, T., Ge, T., Sha, L., Chang, B., Li, S., et al. (2016a). Towards time-aware knowledge graph completion. In Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers (pp. 1715–1724).
[19] Jiang, T., Liu, T., Ge, T., Sha, L., Li, S., Chang, B., et al. (2016b). Encoding temporal information for time-aware link prediction. In Proceedings of the 2016 conference on empirical methods in natural language processing (pp. 2350–2354).
[20] Ji, G., Liu, K., He, S., & Zhao, J. (2016). Knowledge graph completion with adaptive sparse transfer matrix. Thirtieth AAAI conference on artificial intelligence.
[21] Wang, Z., & Li, J.-Z. (2016). Text-enhanced representation learning for knowledge graph.. In Ijcai (pp. 1293–1299).
[22] Xie, R., Liu, Z., Jia, J., Luan, H., & Sun, M. (2016). Representation learning of knowledge graphs with entity descriptions. Thirtieth aaai conference on artificial intelligence.
[23] Tang, X., Chen, L., Cui, J., & Wei, B. (2019). Knowledge representation learning with entity descriptions, hierarchical types, and textual relations. Information Processing & Management, 56(3), 809–822.
[24] Xie, R., Liu, Z., Jia, J., Luan, H., & Sun, M. (2016). Representation learning of knowledge graphs with entity descriptions. Thirtieth aaai conference on artificial intelligence.
[25] Shi, B., & Weninger, T. (2017a). Open-world knowledge graph completion. arXiv:1711. 03438
[26] Dettmers, T., Minervini, P., Stenetorp, P., & Riedel, S. (2018). Convolutional 2d knowl- edge graph embeddings. Thirty-second AAAI conference on artificial intelligence.
[27] Guo, L., Zhang, Q., Ge, W., Hu, W., & Qu, Y. (2018). DSKG: A deep sequential model for knowledge graph completion. In China conference on knowledge graph and semantic computing (pp. 65–77). Springer.
[28] Xiong, W., Hoang, T., & Wang, W. Y. (2017). Deeppath: A reinforcement learning method for knowledge graph reasoning. In Proceedings of the 2017 conference on empirical methods in natural language processing (pp. 564–573).
[29] Das, R., Dhuliawala, S., Zaheer, M., Vilnis, L., Durugkar, I., Krishnamurthy, A., et al. (2018). Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. ICLR.
[30] Schank, R. C. (1983). Dynamic memory: A theory of reminding and learning in computers and people. cambridge university press.
[31] Kolodner, J. L. (1983). Maintaining organization in a dynamic long-term memory. Cognitive science, 7(4), 243-280.
[32] Siegel, S. (1957). Nonparametric statistics. The American Statistician, 11(3), 13-19.
[33] Russell, S. J., & Norvig, P. (2016). Artificial intelligence: a modern approach. Pearson.
[34] Nejati, M. (Ed.). (2013). Frontiers of business, management and economics: An interdisciplinary collection of managerial research findings and breakthroughs. Universal-Publishers.
[35] Rissland, E. L. (1983, August). Examples in Legal Reasoning: Legal Hypotheticals. In IJCAI (pp. 90-93).
[36] Ross, B. H. (1984). Remindings and their effects in learning a cognitive skill. Cognitive psychology, 16(3), 371-416.

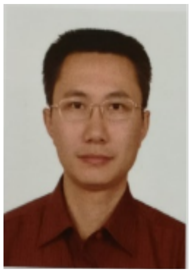
[37] Schmidt, H. G., Norman, G. R., & Boshuizen, H. P. (1990). A cognitive perspective on medical expertise: theory and implication [published erratum appears in Acad Med 1992 Apr; 67 (4): 287]. Academic medicine, 65(10), 611-21.
[38] Wiseman, S., Backurs, A., & Stratos, K. (2021). Data-to-text generation by splicing together nearest neighbors. arXiv preprint arXiv:2101.08248.
[39] Atzeni, M., Dhuliawala, S. Z., Murugesan, K., & Sachan, M. (2022). Case-based reasoning for better generalization in textual reinforcement learning. In International Conference on Learning Representations.
[40] Valentino, M., Thayaparan, M., & Freitas, A. (2022, October). Case-based abductive natural language inference. In Proceedings of the 29th International Conference on Computational Linguistics (pp. 1556-1568).
[41] Swarup, D., Bajaj, A., Mysore, S., O'Gorman, T., Das, R., & McCallum, A. (2020, November). An instance level approach for shallow semantic parsing in scientific procedural text. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 3010-3017).
[42] Das, R. (2022). Nonparametric Contextual Reasoning for Question Answering over Large Knowledge Bases.
[43] Bichindaritz, I. (2007, August). Prototypical cases for knowledge maintenance in biomedical CBR. In International Conference on Case-Based Reasoning (pp. 492-506). Berlin, Heidelberg: Springer Berlin Heidelberg.
[44] Das, R., Godbole, A., Naik, A., Tower, E., Zaheer, M., Hajishirzi, H., ... & McCallum, A. (2022, June). Knowledge base question answering by case-based reasoning over subgraphs. In International Conference on Machine Learning(pp. 4777-4793). PMLR.
[45] Ye, C., Liao, L., Liu, S., & Chua, T. S. (2022, October). Reflecting on experiences for response generation. In Proceedings of the 30th ACM International Conference on Multimedia (pp. 5265-5273).
[46] Wiseman, S., Backurs, A., & Stratos, K. (2021). Data-to-text generation by splicing together nearest neighbors. arXiv preprint arXiv:2101.08248.
[47] Atzeni, M., Dhuliawala, S. Z., Murugesan, K., & Sachan, M. (2022). Case-based reasoning for better generalization in textual reinforcement learning. In International Conference on Learning Representations.
[48] Valentino, M., Thayaparan, M., & Freitas, A. (2022, October). Case-based abductive natural language inference. In Proceedings of the 29th International Conference on Computational Linguistics (pp. 1556-1568).
[49] Swarup, D., Bajaj, A., Mysore, S., O'Gorman, T., Das, R., & McCallum, A. (2020, November). An instance level approach for shallow semantic parsing in scientific procedural text. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 3010-3017).
[50] Jiang, F., Dong, L., Peng, Y., Wang, K., Yang, K., Pan, C., & You, X. (2024). Large AI model empowered multimodal semantic communications. IEEE Communications Magazine.
[51] Jiang, F., Peng, Y., Dong, L., Wang, K., Yang, K., Pan, C., ... & Dobre, O. A. (2024). Large language model enhanced multi-agent systems for 6G communications. IEEE Wireless Communications.
[52] Shen, Y., Shao, J., Zhang, X., Lin, Z., Pan, H., Li, D., ... & Letaief, K. B. (2024). Large language models empowered autonomous edge AI for connected intelligence. IEEE Communications Magazine.
[53] Zhong, J., Li, M., Chen, Y., Wei, Z., Yang, F., & Shen, H. (2024). A safer vision-based autonomous planning system for quadrotor uavs with dynamic obstacle trajectory prediction and its application with llms. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 920-929).
[54] Das, R., Godbole, A., Dhuliawala, S., Zaheer, M., & McCallum, A. (2020). A simple approach to case-based reasoning in knowledge bases. arXiv preprint arXiv:2006.14198.
[55] Das, R. (2022). Nonparametric Contextual Reasoning for Question Answering over Large Knowledge Bases.
[56] Lv, Xin, et al. "Do pre-trained models benefit knowledge graph completion? a reliable evaluation and a reasonable approach." Association for Computational Linguistics, 2022.
[57] Saxena, A., Kochsiek, A., & Gemulla, R. (2022). Sequence-to-sequence knowledge graph completion and question answering. arXiv preprint arXiv:2203.10321.
[58] Chen, C., Wang, Y., Li, B., & Lam, K. Y. (2022). Knowledge is flat: A seq2seq generative framework for various knowledge graph completion. arXiv preprint arXiv:2209.07299.
[59] Li, D., Yi, M., & He, Y. (2022). Lp-bert: Multi-task pre-training knowledge graph bert for link prediction. arXiv preprint arXiv:2201.04843.
[60] Choi, B., & Ko, Y. (2023). Knowledge graph extension with a pre-trained language model via unified learning method. Knowledge-Based Systems, 262, 110245.
[61] Tuan, N. T., Moore, P., Thanh, D. H. V., & Pham, H. V. (2024). A Generative Artificial Intelligence Using Multilingual Large Language Models for ChatGPT Applications. Applied Sciences, 14(7), 3036.
[62] Yang, R., Zhu, J., Man, J., Fang, L., & Zhou, Y. (2024). Enhancing text-based knowledge graph completion with zero-shot large language models: A focus on semantic enhancement. Knowledge-Based Systems, 300, 112155.
[63] Pham, H. V., & Moore, P. (2024). Applied Hedge Algebra Approach with Multilingual Large Language Models to Extract Hidden Rules in Datasets for Improvement of Generative AI Applications. Information, 15(7), 381.
[64] Guo, T., Yang, Q., Wang, C., Liu, Y., Li, P., Tang, J., ... & Wen, Y. (2024). Knowledgenavigator: Leveraging large language models for enhanced reasoning over knowledge graph. Complex & Intelligent Systems, 1-14.
[65] Lei, W., Zhang, G., He, X., Miao, Y., Wang, X., Chen, L., & Chua, T. S. (2020, August). Interactive path reasoning on graph for conversational recommendation. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 2073-2083).
[66] Adami, C. (2004). Information theory in molecular biology. Physics of Life Reviews, 1(1), 3-22.
[67] Majda, A. J., & Gershgorin, B. (2010). Quantifying uncertainty in climate change science through empirical information theory. Proceedings of the National Academy of Sciences, 107(34), 14958-14963.
[68] Quian Quiroga, R., & Panzeri, S. (2009). Extracting information from neuronal populations: information theory and decoding approaches. Nature Reviews Neuroscience, 10(3), 173-185.
[69] Gallager, R. G. (1968). Information theory and reliable communication (Vol. 588). New York: Wiley.
[70] Wang, C., & Shen, H. W. (2011). Information theory in scientific visualization. Entropy, 13(1), 254-273.
[71] Chen, N. (2023). Introduction to Information Theory. In Stochastic Methods for Modeling and Predicting Complex Dynamical Systems: Uncertainty Quantification, State Estimation, and Reduced-Order Models (pp. 23-37). Cham: Springer International Publishing.
[72] Salichos, L., Stamatakis, A., & Rokas, A. (2014). Novel information theory-based measures for quantifying incongruence among phylogenetic trees. Molecular biology and evolution, 31(5), 1261-1271.
[73] Berlinger, E. (1980, May). An information theory based complexity measure. In Proceedings of the May 19-22, 1980, national computer conference (pp. 773-779).
[74] Brillouin, L. (2013). Science and information theory. Courier Corporation.
[75] Mourtzis, D., Fotia, S., Boli, N., & Vlachou, E. (2019). Modelling and quantification of industry 4.0 manufacturing complexity based on information theory: a robotics case study. International Journal of Production Research, 57(22), 6908-6921.
[76] Maassen, H., & Uffink, J. B. (1988). Generalized entropic uncertainty relations. Physical review letters, 60(12), 1103.
[77] Zhou, M., Zhu, S. S., Chen, Y. W., Wu, J., & Herrera-Viedma, E. (2021). A generalized belief entropy with nonspecificity and structural conflict. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52(9), 5532-5545.
[78] Son, J., Lee, J., Larsen, K. R., & Woo, J. (2020). Understanding the uncertainty of disaster tweets and its effect on retweeting: The perspectives of uncertainty reduction theory and information

entropy. Journal of the Association for Information Science and Technology, 71(10), 1145-1161.
[79] Xiao, F. (2019). Multi-sensor data fusion based on the belief divergence measure of evidences and the belief entropy. Information Fusion, 46, 23-32.
[80] Zhang, Y., Huang, F., Deng, X., & Jiang, W. (2021). A New Total Uncertainty Measure from A Perspective of Maximum Entropy Requirement. Entropy, 23(8), 1061.
[81] Zhang, H., Zhong, S., Deng, Y., & Cheong, K. H. (2021). LFIC: Identifying influential nodes in complex networks by local fuzzy information centrality. IEEE Transactions on Fuzzy Systems, 30(8), 3284-3296.
[82] Lv, Z., Lu, Z., & Wang, P. (2015). A new learning function for Kriging and its applications to solve reliability problems in engineering. Computers & Mathematics with Applications, 70(5), 1182-1197.
[83] Cui, H., Liu, Q., Zhang, J., & Kang, B. (2019). An improved deng entropy and its application in pattern recognition. IEEE Access, 7, 18284-18292.
[84] Fresneda, J. E., & Gefen, D. (2020). Gazing at the stars is not enough, look at the specific word entropy, too!. Information & Management, 57(8), 103388.
[85] Tilei, G., Tong, L., Ming, Y., & Rong, J. (2019). Research on a trustworthiness measurement method of cloud service construction processes based on information entropy. Entropy, 21(5), 462.
[86] Zhang, Y., Dai, H., Kozareva, Z., Smola, A., & Song, L. (2018, April). Variational reasoning for question answering with knowledge graph. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1)
[87] Lv, X., Lin, Y., Cao, Y., Hou, L., Li, J., Liu, Z., ... & Zhou, J. (2022). Do pre-trained models benefit knowledge graph completion? a reliable evaluation and a reasonable approach. Association for Computational Linguistics.
[88] Xiong, W., Hoang, T., & Wang, W. Y. (2017). Deeppath: A reinforcement learning method for knowledge graph reasoning. arXiv preprint arXiv:1707.06690.
[89] Yang, Bishan, et al. "Embedding entities and relations for learning and inference in knowledge bases." arXiv preprint arXiv:1412.6575 (2014)
[90] Trouillon, Théo, et al. "Complex embeddings for simple link prediction." International conference on machine learning. PMLR, 2016.



YING ZHANG was born in 1996. She received the M.S. degree in data science from King's College of London (KCL), British, in 2020. She is working toward the PH.D. degree in the Institute of Systems Engineering, Academy of Military Sciences. Her research interests include knowledge graph refinement and knowledge graph evaluation.

YANGPENG SHEN was born in 1994. He received his B.S. degree in mathematics from University of Alberta, Canada, his M.S. degree in data science from the University of Sydney, Australia, in 2020. Currently, he is serving as a researcher in the CITIC Bank. His research focuses data mining, natural language processing and deep learning.



GANG XIAO received the B.S. degree and Ph.D. degree in computer science and technology from National University of Defense Technology. He is currently a Professor of Institute of Systems Engineering, Academy of Military Sciences. His research interests include complex system theory, system design, knowledge graph and deep learning.

JINGHUI PENG received his B.S. degree from Shenyang Aerospace University, his M.B. degree from the Naval University of Engineering, and his Ph.D. degree from the Institute of Systems Engineering at the Military Academy of Sciences. Currently, he is serving as a lecturer in the School of Artificial Intelligence at Anhui Polytechnic University. His research focuses encompass multimodal knowledge graphs and swarm intelligence technologies.