electronics

MDPI

Article

Construction of Journal Knowledge Graph Based on Deep Learning and LLM

Jiankun Zuo and Jiaojiao Niu *

School of Computer Science, Yangtze University, Jinzhou 434023, China; zuojiankunn@163.com
* Correspondence: njjiao_92@163.com

Abstract: Knowledge graphs are powerful tools for representing the relationships between concepts and entities in the real world through triples. Due to their superior knowledge representation and efficient reasoning abilities, knowledge graphs have gained widespread attention across various fields, leading to their development in multiple domains. However, research on the construction of journal knowledge graphs remains relatively limited, posing challenges for the integration and utilization of knowledge in the journal domain. To address this gap, this study explores effective methods for constructing journal knowledge graphs and develops a journal knowledge graph-based question answering system. Specifically, journal datasets were collected from multiple sources using the Scrapy framework, encompassing structured, semi-structured, and unstructured data. A BERT-BiLSTM-CRF framework was then employed to extract entities, attributes, and relationships from semi-structured and unstructured data. In addition, the constructed journal knowledge graph was integrated with large language models (LLMs) to build a journal-related question answering system, facilitating efficient querying and utilization. Finally, Neo4j was used for storing the constructed journal knowledge graph.

Keywords: knowledge graph construction; deep learning; large language model; question answering systems

check for updates

Academic Editor: Arkaitz Zubiaga

Received: 25 March 2025
Revised: 19 April 2025
Accepted: 22 April 2025
Published: 24 April 2025

Citation: Zuo, J.; Niu, J. Construction of Journal Knowledge Graph Based on Deep Learning and LLM. *Electronics* **2025**, *14*, 1728. https://doi.org/10.3390/electronics14091728

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A knowledge graph is a relational dataset where nodes represent entities or concepts in the real world, and directed edges represent relationships between entities. Due to their rich knowledge representation and excellent knowledge reasoning capabilities, researchers have developed various general-purpose knowledge graphs [1–4] to meet different needs. Although these general-purpose knowledge graphs have been successfully applied in relevant fields, they often fail to fully meet professional requirements in specialized domains. As a result, the construction of domain-specific knowledge graphs has become a key research topic. Considering the importance of journal data integration and utilization in scientific research, this paper aims to construct a journal knowledge graph and build a journal-related question answering system to fill the existing gap in this area.

The existing knowledge graph construction methods can be categorized into three types [5,6]: bottom-up, top-down, and hybrid construction methods. The bottom-up approach focuses on leveraging raw data by applying a stepwise data analysis strategy to extract relationships between entities, thereby organizing the structure of the knowledge graph. As a result, knowledge graphs constructed by this method are more detailed. The top-down approach involves constructing a knowledge graph based on predefined entity types and relationships. This method requires the development of an ontology and clear

definitions of relationships between entities. Due to the accuracy of the information sources, the top-down approach typically results in knowledge graphs with lower redundancy. The hybrid approach combines the strengths of both bottom-up and top-down methods. In this approach, a large volume of entities and relationships is initially extracted using the bottom-up method, and after which the top-down approach is employed to filter, refine, and supplement the information with domain expert knowledge and rules, ensuring the accuracy and completeness of the knowledge graph. However, the hybrid method still faces challenges such as dependency on domain experts, potential inconsistencies in knowledge, and coordination difficulties. The quality of a knowledge graph is closely tied to the knowledge extraction process, which relies heavily on the named entity recognition (NER) techniques [7] applied.

Currently, there are two main types of NER methods: one is the machine learning-based approaches, such as the Markov-based models [8] and Conditional Random Fields (CRF) [9], and the other one is the deep learning-based approaches, such as the BiLSTM [10] model. Due to their better ability to handle complex language structures and textual information, deep learning-based methods have become widely used. In the literature, Collobert et al. [11] first systematically demonstrated how convolutional layers can be used to extract local features from sentences. Based on this, Strubell et al. [12] introduced Iterative Convolutional Neural Networks (ID-CNNs) for large-scale text and structured data prediction. To better utilize contextual information within sequences, Qiu et al. [13] explored the use of the LSTM-CRF model for entity recognition in geological reports by capturing contextual sequence information. Žukov-Gregorič et al. [14] employed multiple independent Bi-LSTM distributions for entity recognition, significantly reducing the number of parameters. To further make full use of textual information, researchers have incorporated the BERT [15] pre-trained language model into deep-learning models. For instance, Yang et al. [16] built the BERT-BiLSTM-IDCNN-CRF model for tracking the polysemy of words; Chen et al. [17] proposed a BERT-BiLSTM-CRF-based entity extraction method to extract feature information from high-frequency speech, while Zhu et al. [18] used the BERT-BiLSTM-CRF model to extract remanufacturing process entities and constructed a remanufacturing process knowledge graph. In addition to BERT, other commonly used pre-trained models include Word2Vec [19], ELMo [20], and GPT [21]. Notably, BERT effectively integrates the advantages of other pre-trained models through masked language modeling (MLM) and next-sentence prediction during pre-training, thereby becoming exceptionally popular in the field.

As a core subfield of natural language processing (NLP), intelligent question-answering (QA) aims to comprehend and respond to user queries formulated in natural language.

In recent years, several open-source large language models such as Zephyr [22], Mistral [23], Phi [24], Flan-T5 [25], ChatGPT [26], ChatGLM [27], and LLaMa3 [28] have demonstrated strong contextual understanding and language generation capabilities through training on large-scale datasets. Leveraging the significant advantages of LLMs in terms of accuracy and efficiency [29], question-answering systems based on these models have become mainstream approaches. In specialized domains such as finance and law, domain-specific large models like FinGPT [30] and ChatLaw [31] have driven real-world applications of these technologies, providing strong support for the intelligent processing of professional knowledge. However, despite the effectiveness of LLM-based QA systems, challenges such as hallucinations can impact their accuracy [32]. To address this, using knowledge graphs to enhance the accuracy of LLM-based QA systems has proven effective. For instance, Zhou et al. [33] enhanced general large language models (GLMs) for CPM-QA using a multimodal CMP knowledge graph (CPM-KG). Sun et al. [34] intro-

duced the Think-on-Graph (ToG) framework, combining LLMs with knowledge graphs to improve reasoning capabilities in domain-specific tasks. Jiang et al. [35] proposed the StructGPT framework, which constructs an iterative reading and reasoning mechanism to access and filter structured data, such as knowledge graphs and databases, enabling better reasoning. Feng et al. [36] developed the Knowledge Solver (KSL) approach, which leverages the general capabilities of LLMs to retrieve domain knowledge from external knowledge graphs. Furthermore, the Retrieval-Augmented Generation (RAG) [37] technique, which combines retrieval and generation, has been widely adopted to enhance the question-answering performance of large language models. For instance, Hou et al. [38] introduced the RAG technique into medical question-answering systems, improving the accuracy of answers by retrieving structured information from medical knowledge graphs. Hang et al. [39] developed a novel framework for generating personalized multiple-choice questions (MCQs) by integrating retrieval-augmented generation with prompt engineering techniques, utilizing large language models. Additionally, Aigerim et al. [40] alleviated the "hallucination" problem by retrieving relevant information from external knowledge sources, thereby supplementing the contextual knowledge of LLMs.

Based on the above discussion, this paper adopts a bottom-up approach to construct a journal knowledge graph. For structured data, the information is stored as a triple (E, R, F), where E represents entities, R denotes relationships, and F refers to facts or attribute values. For unstructured and semi-structured data, the BERT-BiLSTM-CRF framework was employed to extract entities, attributes, and relationships, which are then further integrated and aligned with the structured data through consistency checks. Considering the powerful understanding and expression capabilities of LLMs, this paper further proposes an intelligent question answering system (JKG-LLM) that integrates a journal knowledge graph with LLMs. In summary, the main contributions of this paper are as follows:

1. We construct the journal knowledge graph by integrating structured, semi-structured, and unstructured data with the BERT-BiLSTM-CRF model for entity and relation extraction.
2. We propose JKG-LLM, a journal knowledge graph-based question-answering system that enhances response accuracy through retrieval-augmented generation.

The structure of the paper is as follows: Section 2 describes the construction process of the journal knowledge graph; Section 3 further introduces the usage of the journal knowledge graph and the question-answering system; and Section 4 presents the experimental results. Section 5 concludes the paper.

2. The Construction of Journal Knowledge Graph

The construction of the journal knowledge graph consists of five key components: (1) dataset acquisition; (2) knowledge extraction; (3) knowledge fusion; (4) ontology construction; and (5) knowledge storage and visualization. Figure 1 illustrates the overall construction process of the journal knowledge graph.

2.1. Dataset Acquisition

Acquiring high-quality, multi-source journal datasets is crucial for constructing a journal knowledge graph. To ensure the comprehensiveness and accuracy of entity and relationship information, this paper collects journal datasets from various sources, including China National Knowledge Infrastructure (CNKI), Wanfang Data, Web of Science, official journal websites, libraries, and others.

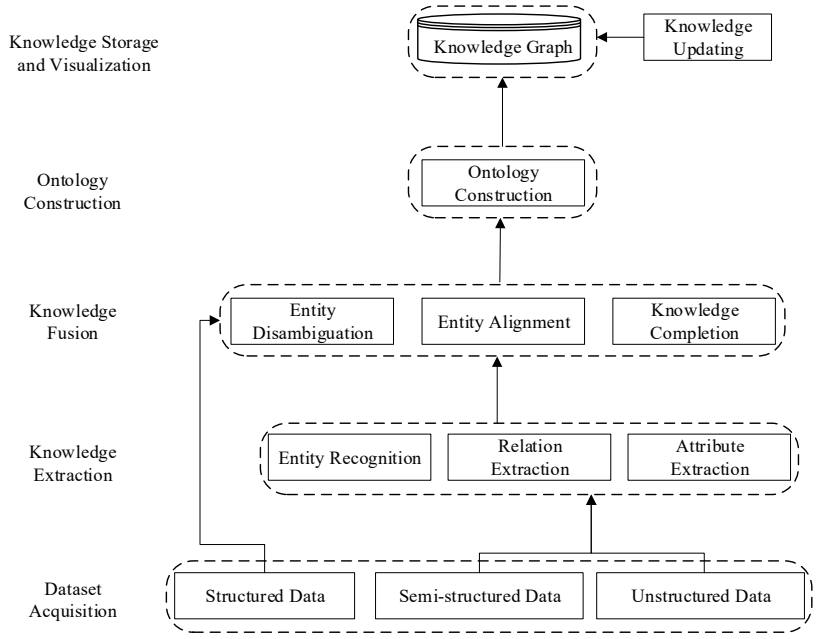


Figure 1. The overall construction process of the journal knowledge graph.

According to the degree of structure, the datasets can be divided into three categories: structured data, semi-structured data, and unstructured data. Structured data generated from databases and third-party journal websites have a clear format and standardized structure, making it easy for direct analysis and processing. Semi-structured data, while partially structured, do not fully adhere to a standardized format. Unstructured data lack a fixed format and usually exist in the form of natural language text. For semi-structured and unstructured data, this paper employed the Scrapy framework [41] for web crawling, with the data presented in formats such as HTML and XML. During the crawling process, the web crawler automatically accessed target websites, grabbed relevant pages, and extracted the required information based on predefined rules and strategies, converting it into structured data for subsequent analysis and use. The crawling process for unstructured and semi-structured data is shown in Figure 2.

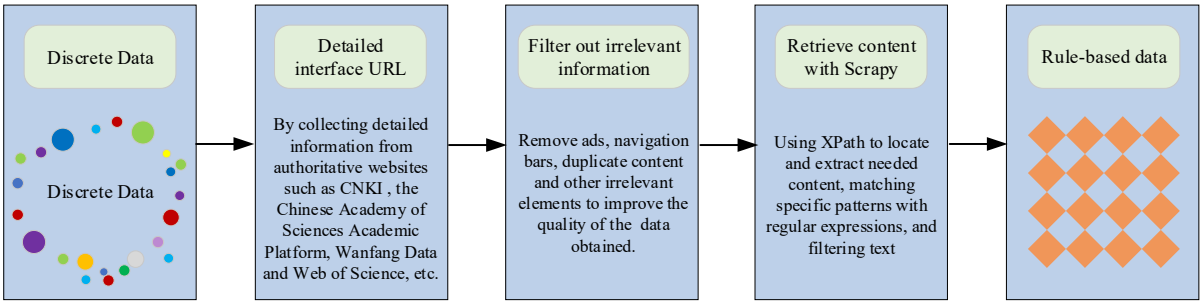


Figure 2. Process for obtaining unstructured and semi-structured journal datasets.

2.2. Knowledge Extraction

Knowledge extraction is one of the key steps in constructing a knowledge graph. This step aims to automatically or semi-automatically extract knowledge units such as entities, relationships, and attributes from text. Examples of extractions can be seen in Table 1.

Table 1. Knowledge extraction example for International Journal of Computer Vision.

Entity	Relation	Attribute
International Journal of Computer Vision	Acceptance Difficulty	Fairly Difficult
International Journal of Computer Vision	Press	Springer US
International Journal of Computer Vision	WOS partition	Q 1
International Journal of Computer Vision	Review Cycle	About 4.5 months
International Journal of Computer Vision	Major Category	Computer Science

Since deep learning-based relationship extraction methods and LLM-based models offer superior accuracy compared to traditional machine learning methods, this paper adopted the BERT-BiLSTM-CRF model for relationship extraction. In the knowledge extraction process, the BERT pre-trained model first converts the raw input text into corresponding word vectors. A BiLSTM model was then used to capture contextual features from the input text. Finally, due to the strong dependency between output labels, CRF was employed to decode the output from the BiLSTM and generate the label sequence with the highest probability.

For relationship extraction, the CRF module performs global optimal sequence labeling to optimize the output from the BiLSTM. The score $s(x, y)$ of the input sequence x for the label y is calculated by:

$$s(x, y) = \sum_{i=0}^{n} W_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i} \tag{1}$$

where $W_{i,j}$ represents the transition weight from label y_i to y_j. P_{i,y_i} denotes the probability of the i word corresponding to label y_i. Furthermore, the probability distribution of the input sequence can be obtained by:

$$P(y|x) = \frac{e^{s(x,y)}}{\sum_{y \in Y} e^{s(x,y)}} \tag{2}$$

where Y is the set of all possible label sequences. During the prediction process, the label sequence with the highest probability is selected, as described by the following equation:

$$y^* = \arg\max_{y \in Y} s(x, y) \tag{3}$$

Figure 3 illustrates the knowledge extraction process for the input sequence "Chinese Journal of Computers was first established in 1978, and is published monthly". The BERT-BiLSTM-CRF model identifies and labels entities using the BIO tagging method. After tagging, the entities in the input sequence are identified as "Journal of Computer Science" (tagged as: B-ORG I-ORG I-ORG I-ORG I-ORG), "1978" (tagged as: B-DATA), and "monthly" (tagged as: B-TIME). The relationship corresponding to the entity "Journal of Computer Science" and the attribute "1978" is "founded in" and the relationship corresponding to the attribute "monthly" is "established". Ultimately, two triplets are extracted: <Chinese Journal of Computers, established in, 1978> and <Chinese Journal of Computers, published, monthly>.

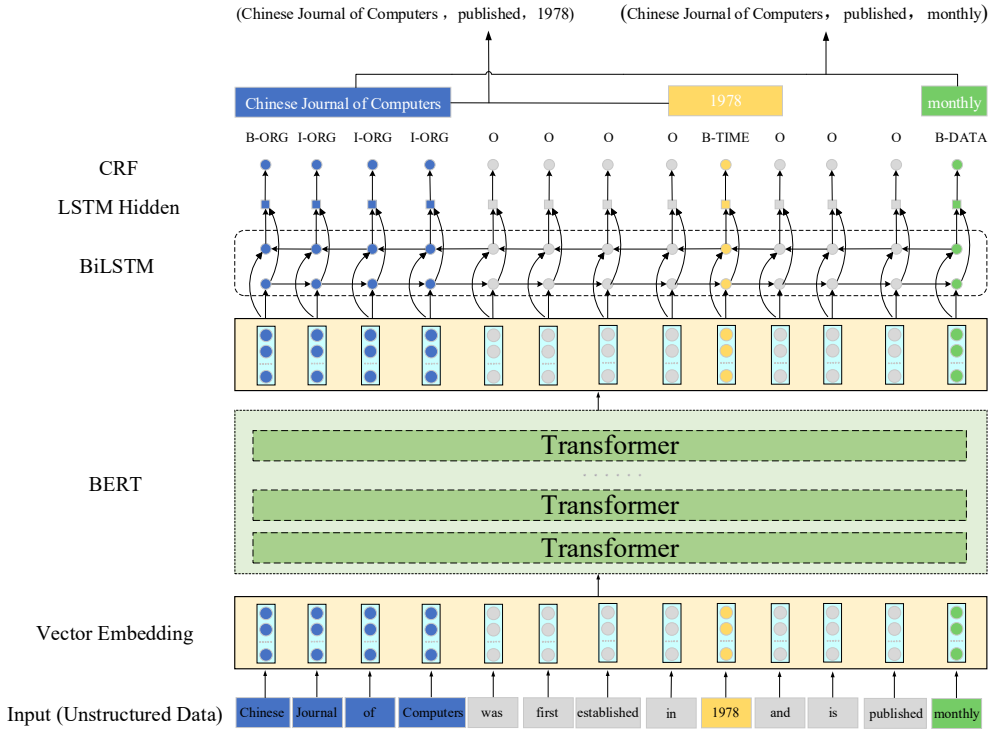


Figure 3. Example of knowledge extraction with the BERT-BLSTM-CRF model.

2.3. Knowledge Fusion

Knowledge fusion aims to integrate information about the same entity from different knowledge bases and multi-source heterogeneous external knowledge into a unified framework. The main tasks in this process include entity alignment, entity disambiguation, and knowledge completion.

For entity alignment tasks, this study employs a context-based BERT model to enhance accuracy. By leveraging the bidirectional encoding mechanism, BERT effectively captures the surrounding contextual semantics of entities, reducing ambiguities that commonly arise in dictionary-based matching methods. Furthermore, BERT encodes entity contexts into vector representations, facilitating more precise entity matching and alignment across diverse data sources.

Conflicts may arise during information extraction from diverse data sources, including value conflicts, structural conflicts, and semantic conflicts. For value and structural conflicts, data priority is determined as follows: the journal's official website is prioritized over CNKI and Web of Science, which in turn have higher priority than third-party sources (e.g., LetPub, Academic Home, etc.). If no relevant information is found, the next-level data source will be queried sequentially. For semantic conflicts, this study employs entity disambiguation [42] as the solution strategy. The key is to distinguish entities with the same name but different meanings. For example, "Machine Learning" can refer to a journal or a keyword. To address this, the BERT model encodes contextual semantics into vector representations. "Machine Learning" will have different vector representations in different contexts, allowing its specific meaning to be inferred based on contextual similarity.

Knowledge completion [43] refers to filling in missing values or gaps in a knowledge graph, which is often achieved through various methods. This paper employs a rule-based inference model. For example, in the case of the journal IET Computer Vision, which is published by Wiley in England, if the publication location of IET Intelligent Transport Systems is unknown, it can be inferred that the publication location is also England, as both journals are published by Wiley.

2.4. Ontology Construction

Ontology construction defines the concepts, attributes, and relationships of entities, providing a clear semantic hierarchy and constraint rules for the knowledge graph. Existing ontology construction methods can be divided into manual, semi-automatic, and automatic construction methods. Manual construction methods ensure high-quality entities but rely heavily on domain experts, leading to high costs and low efficiency Semi-automated construction methods combine human expertise with machine assistance. Although this method can improve construction efficiency, its reliance on human involvement limits its applicability to large-scale ontology construction. Automatic construction methods extract entities directly from data sources, enabling efficient and rapid ontology construction. However, due to technological and algorithmic limitations, the accuracy and completeness of automatically constructed ontologies may not always meet the standards of the first two methods. To balance entity quality and construction time, this paper adopts the semi-automatic ontology construction approach.

During ontology construction processes, we designed a well-defined semantic hierarchy by defining the concepts, attributes, and relationships of entities, see details in Tables 2 and 3. To address the complex issues of cross-domain journals, evolving indexing statuses, and interdisciplinary classifications, we implemented the following designs and optimizations: For cross-domain journals, the Research fields attribute in the Journal class supports multi-valued input, allowing a journal to be associated with multiple research fields. For example, IEEE Transactions on Knowledge and Data Engineering can be labeled as both “Computer Science” and “Information Systems”. Second, to track the evolving indexing statuses, the Indexing and WOS partition attributes in the Journal class record the journal’s current state. To capture the dynamic nature of indexing statuses, we added time information to the WOS partition attribute (e.g., “2023: Q1, 2024: Q2”). Finally, to support interdisciplinary classifications, we extended the Contains relationship in the Domain class, allowing a Major to belong to multiple Domains. For instance, “Deep Learning” can be classified under both “Computer Science” and “Neuroscience”.

Table 2. Data attributes of journal ontology.

Class	Data Property	Description
Domain	name	Name of the academic discipline, e.g., ”Computer Science”
Major	name	Specific research direction within a discipline, e.g., “Deep Learning”
Journal	Keywords, Cycle, Research fields, CAS partition, Acceptance difficulty, Indexing, Journal metrics, Journal count, ISSN, EISSN, Impact factor, H-index, Abstract, Official website, Contact information, Publisher, Research directions, Publication region, Annual article count, Gold OA article percentage, WOS partition, Warning list, CAS major partition, CAS minor partition, TOP journal, Review journals, SCI indexed, PMC link	Various attributes of journals
Citation Score	SNIP	Source Normalized Impact per Paper (Scopus SNIP)
Impact Factor	Last five-year impact factor, Annual impact factor	Annual impact factor

Table 2. *Cont.*

Class	Data Property	Description
CAS partition	CAS major Partition, CAS minor Partition	Classification of the journal in the CAS (Chinese Academy of Sciences) system
Website	Journal website, Submission website	Websites related to the journal and its submission process
WOS partition	Discipline category	Discipline category in the Web of Science (WOS)
Publisher	Publication language, Publication year, Publication cycle, Publication region	Details about the publisher, including language, year, cycle, and region
Citation score	SNIP, SJR	Source Normalized Impact per Paper (SNIP), SCImago Journal Rank (SJR)

Table 3. Object attributes of journal ontology.

Object Property	Domain	Range
Contains	Domain	Major
Owns	Major	Journal
Owns	Journal	Keywords, Cycle, Research fields, CAS partition, Acceptance difficulty, Indexing, Journal metrics, Journal count, ISSN, EISSN, H-index, Abstract, Official website, Contact information, Publisher, Research directions, Publication region, Annual article count, Gold OA article percentage, WOS partition, Warning list, TOP journal, Review journals, SCI indexed, PMC link
Owns	CAS partition	CAS major Partition, CAS minor Partition
Owns	CAS major Partition	CAS Major Partition Rank
Owns	CAS minor Partition	CAS Minor Partition Rank
Submits to	website	Submission website, Journal website
Partition	WOS partition	Discipline category
Publication language	Publisher	Publication language
Publication year	Publisher	Year
Publication cycle	Publisher	Publication cycle
Publication region	Publisher	Region
SNIP metric	Citation score	SNIP
SJR metric	Citation score	SJR
Last five-year impact factor	Impact factor	Impact factor (last five years)
Annual impact factor	Impact factor	Impact factor (annual)

To ensure the quality and consistency of the ontology, a rigorous validation mechanism that included both a manual review and an automated consistency check was implemented. The manual review process assessed the accuracy of entity definitions and relationships to ensure that the ontology's knowledge adheres to the standards and requirements of the domain. The automated tools were utilized to analyze relationships between entities, ensuring the absence of conflicts or inconsistencies in entity definitions.

2.5. Knowledge Storage and Visualization

Knowledge storage and visualization are the final steps in constructing a journal knowledge graph. These steps focus on storing the extracted triplets and ontologies while enabling effective visualization.

After the construction was completed, this paper successfully built a journal knowledge graph containing over 200,000 triplets. For storage, the Neo4j graph database was selected as the storage engine for the journal knowledge graph. In Neo4j, entities are represented as circular nodes, and relationships are denoted by labeled edges. Figure 4 illustrates the knowledge structure obtained from querying Acta Automatica Sinica using Cypher (see Section 3.1). The constructed Chinese knowledge graph, as depicted in Figure 4, includes various details such as the publisher, publication cycle, WOS partition, communication method, and OA status of the journal.

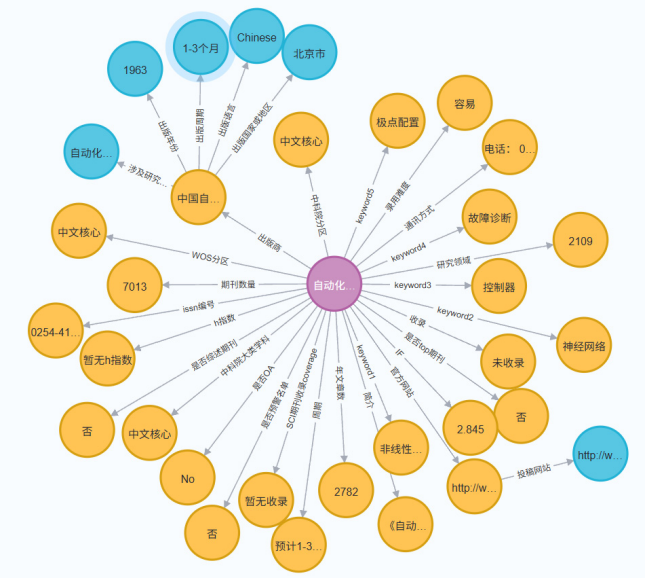


Figure 4. Knowledge graph of the Acta Automatica Sinica.

Considering the timeliness of the journal information, incorporating a knowledge update process is crucial in constructing a journal knowledge graph. The existing update process consists of schema-level updates and data-level updates. The former involves modifying the data structure of the knowledge graph, while the latter focuses on adding new entities or updating relationships between existing entities. This study focuses solely on the data-level updates of the journal knowledge graph. Specifically, this study introduces a version control mechanism. Administrators regularly update the data layer to ensure that the ontology accurately reflects changes in journal information. Each journal entry includes a timestamp and version number to track updates to journal information. Through this update mechanism, the system can regularly update journal information based on the latest data sources and incorporate changes into the ontology in a timely manner, ensuring the ontology's timeliness and completeness.

3. The Question Answering System Based on Journal Knowledge Graph

This Section primarily introduces how to query the constructed knowledge graph and develop a question-answering system based on journal knowledge graphs.

3.1. Knowledge Query

Querying the journal knowledge graph can be achieved with Cypher, the graph query language provided by the Neo4j database. A Cypher query typically consists of four

parts: expressions, graph patterns, clauses, and queries. An expression usually refers to operations performed on values within the query, such as numerical operations, string operations, relationship operations, list operations, map operations, and path operations. The graph pattern defines the structure of nodes and relationships in the query, typically defined within the MATCH clause. The MATCH clause is the core element in Cypher, which is used to query the database. These components provide flexible operations for complex journal information retrieval. For example, the following query retrieves detailed information about the Proceedings of the IEEE journal:

```
MATCH (n)
WHERE n.Source= 'Proceedings of the IEEE'
RETURN n
```

Due to the fact that we are constructing a Chinese knowledge graph, some Chinese journals and relationships are represented in Chinese. The query results are shown in Figure 5. The Proceedings of the IEEE journal is divided into Zone 1 by the Chinese Academy of Sciences, with 'Artificial Intelligence' as one of its keywords. Its impact factor is 23.2, and the five-year average impact factor is 18.4. The review cycle is 3–8 weeks, and the publisher is IEEE. Additionally, the results include information on the country of publication, acceptance difficulty, and other relevant journal details.

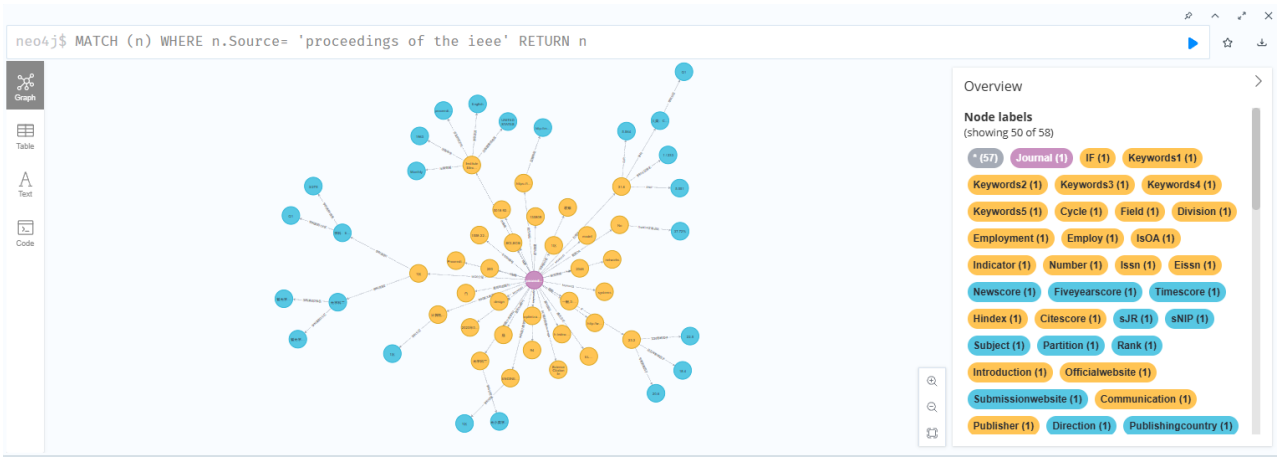


Figure 5. Query results for Proceedings of the IEEE.

3.2. Question Answering

With the continuous expansion of data scale, the traditional knowledge graph-based single-domain question-answering systems face significant limitations. They rely on pre-existing nodes in the graph structure for intelligent queries, making it difficult to associate contextual queries from the same user effectively. Consequently, these systems struggle to meet users' needs for efficient and diverse knowledge retrieval. At the same time, relying solely on retraining or fine-tuning large models to improve the performance of question answering systems is significantly inefficient. Therefore, this study integrates LLMs, KGs within a retrieval-augmented generation technique to construct a hybrid QA system. Specifically, the JKG-LLM system is built on the open-source Python framework LangChain [44] and integrates the Qwen-7B model through Ollama as the generative module, leveraging its 7 billion parameters to efficiently handle complex queries. The Retrieval-Augmented Generation framework in the system operates through a collaborative process between the retriever and the generator: the retriever extracts relevant information from the knowledge graph and vector database, while the generator produces contextually consistent answers based on the retrieved results. In the retrieval process, the knowledge graph is used to extract structured contextual knowledge, while journal metadata are

embedded into dense vectors stored in the vector database via a pre-trained language model. To handle long documents (e.g., journal descriptions), we employ an overlapping segmentation strategy (with each segment consisting of 256 tokens and a 20% overlap) and utilize the BART model to generate summary embeddings, reducing information loss. Retrieval from the knowledge graph is implemented using Cypher queries. For example, when a user queries "SCI Zone 1 journals in the field of Databases", the system identifies key entities in the query (e.g., "Databases" maps to the Major entity, and "SCI Zone 1" corresponds to the WOS partition attribute), then performs the following Cypher query to retrieve relevant journals:

```
MATCH (j:Journal)-[:OWNS]->(m:Major)
WHERE m.name = "Database" AND j.WOS_partition = "Q1"
RETURN j.name AS journal_name
```

The knowledge graph is stored in the Neo4j graph database, leveraging its efficient graph traversal capabilities to accelerate query processing. For the scoring mechanism, we use cosine similarity to compute the degree of match between the retrieved candidate entities and the query vector. During answer generation, the generator ensures logical consistency with the knowledge graph through prompt optimization, and performs semantic reasoning based on the retrieved knowledge, thereby generating contextually appropriate responses.

In JKG-LLM, the knowledge graph has unique advantages in processing structured knowledge and logical reasoning, which can clearly display the relationships and reasoning paths between knowledge and continuously update over time to ensure the real-time performance of the knowledge base. Through the visualization of graph structures, the system can intuitively present a large amount of knowledge and its interrelationships, helping users better understand complex knowledge networks. Meanwhile, Qwen-7B is deployed locally through Ollama, efficiently handling complex queries with its 7 billion parameters, while ensuring logical consistency with the journal knowledge graph through prompt optimization. Based on semantic reasoning from the knowledge graph, it generates accurate answers that are contextually appropriate. Unlike traditional large language models that rely solely on textual data, the JKG-LLM combines the knowledge graph to fill in gaps in logical reasoning, while leveraging the strengths of large language models in addressing open-ended questions, thereby providing more comprehensive and dynamic responses.

Figure 6 shows the overall framework of JKG-LLM, where users interact in multiple rounds through natural language. Specifically, when a user asks a question, JKG-LLM first processes the query using an LLM to understand the user's intent and generate the preliminary query semantics. This information is then passed on to an intelligent agent (Agent), which determines the subsequent actions based on the type of problem. If relevant information exists in the knowledge base, the Agent retrieves the corresponding content and refines the LLM-generated response to provide a more accurate answer. If no relevant information is available, the system defaults to generating answers using the LLM to avoid an unanswered query. Finally, JKG-LLM comprehensively sorts the retrieved information and generated content, and outputs the optimal solution. Meanwhile, to ensure the timeliness of system responses, administrators regularly perform data maintenance, updating journal information in the knowledge base to accurately reflect the latest journal developments.

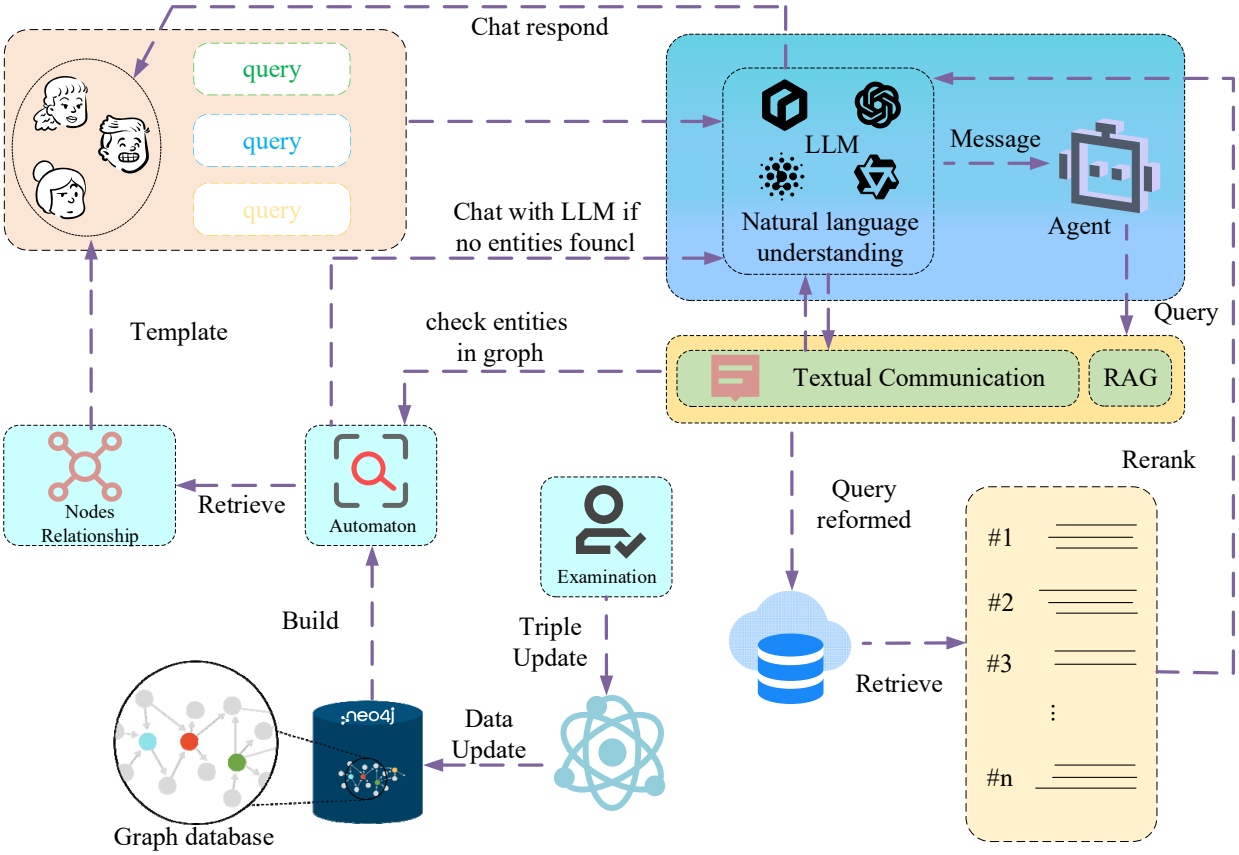


Figure 6. JKG-LLM design framework of Q&A system.

After the user raises a question, JKG-LLM retrieves relevant multidimensional information based on the knowledge graph and provides fast and accurate answers. For example, if a user asks, "I have an article in the database field and would like to submit it to a SCI Zone 1 journal. Can you recommend two journals for me?" JKG-LLM first performs semantic analysis on the natural language query to understand the user's intent and specific requirements, including the topic of "database" and the request for recommendations of two SCI Zone 1 journals. After parsing the query, the LLM generates an initial query statement, which is then executed by an intelligent agent to search the knowledge graph for journals that meet the criteria—belonging to SCI Zone 1 and relevant to the database field. Finally, the LLM formulates a response based on the retrieved results, providing precise recommendations.

The comparison of answers generated by other large models is shown in Table 4. As shown in Table 4, two common issues are observed with the response of large models for comparison: one is the recommendation of incorrect journals, and the other is the suggestion of articles instead of journals. These inconsistencies primarily arise because these LLM models rely on knowledge bases that are updated up to around 2023, which limits their access to the most current journal ranking information. In contrast, JKG-LLM accurately associates JCR journal rankings with their corresponding years, enabling users to retrieve not only the latest ranking information but also the specific year of each ranking. This ensures both accuracy and timeliness in the information provided.

Table 4. Example of large language models for journal recommendation.

LLM	Question: I Have an Article in the Database Field and Would Like to Submit It to a SCI Zone 1 Journal. Can You Recommend Two Journals for Me?	Real Partition	Q&A Results	Applicable Scenarios
	Generate Reply			
ChatGPT	The International Journal on Very Large Data Bases, ACM Transactions on Database Systems	Q2 Q2	×	General Q&A
JKG-LLM	Advanced Engineering Informatics, Information Processing and Management	Q1 Q1	√	Domain-specific journal recommendations
ChatGLM	ACM Transactions on Database Systems, IEEE Transactions on Knowledge and Data Engineering	Q2 Q1	×	General Q&A
Qwen	IEEE Transactions on Knowledge and Data Engineering, ACM Transactions on Database Systems	Q1 Q2	×	General Q&A
01.AI	Effectiveness of Journals Funded by Category D of the China Science and Technology Journal International Influence Enhancement Program—Inclusion in SCIE/SSCI, Impact Analysis of Chinese Scientific and Technical Journals Indexed by Scopus	Two articles	×	General Q&A
Ernie Bot	ACM Transactions on Database Systems, The VLDB Journal	Q2 Q2	×	General Q&A
Deepseek	IEEE Transactions on Knowledge and Data Engineering, ACM Transactions on Database Systems	Q1 Q2	×	General Q&A
OpenAlex	PubMed, PubMed Central	Error	×	Domain Q&A

Unlike traditional knowledge graph-based question-answering systems that rely on SPARQL queries, the JKG-LLM system does not require users to write structured queries. Instead, it directly parses the query content using natural language understanding techniques and returns relevant answers. This approach enables users to interact with the system more easily, without needing to understand complex query syntax.

4. Experiment

4.1. Datasets

Currently, there is no standardized corpus of journal entities in the field of knowledge graphs. To validate the model, this study collected 13,400 web page data entries and constructed an NER dataset covering journal-related domains. During data processing, semi-structured and unstructured data were manually annotated using the BIO tagging method to ensure accuracy. The annotated data were then tokenized, with irrelevant words and punctuation removed to reduce noise. Finally, the dataset was split into training and test sets at an 8:2 ratio to ensure both model generalization and effective training.

4.2. Experimental Environment and Parameter Settings

The experiment was conducted using the deep learning framework TensorFlow 1.14.0 and Keras 2.2.4 in a Python 3.7 environment, the version of Neo4j is community 5.21.0,

Langchain version is v0.2. The hardware configuration included an Intel Core i7-10750H CPU, 32 GB RAM, and an NVIDIA RTX 2060 GPU with CUDA 12.6. During training, the Adam optimizer was employed with a learning rate of 2×10^{-5} and a batch size of 32. To mitigate overfitting, Dropout was applied to both the input and output of the BiLSTM with a dropout rate of 0.1. The detailed parameter settings are provided in Table 5.

Table 5. Parameter configuration.

Parameter Category	Value
Embedding_dim	100
Max_seq_len	256
Hidden_dim	512
Batch_size	32
Learning_rate	2×10^{-5}
Optimizer	Adam
Drop out	0.1
Epoch	50

4.3. Model Evaluation Metrics

To evaluate the effectiveness of the experimental results, the evaluation metrics primarily include Precision (P), Recall (R), Accuracy (A), and F1-score. The calculation formulas are as follows:

$$P = \frac{TP}{TP + FP} \times 100\% \tag{4}$$

$$R = \frac{TP}{TP + FN} \times 100\% \tag{5}$$

$$F1 = \frac{2 \times P \times R}{P + R} \times 100\% \tag{6}$$

$$A = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{7}$$

In the formula, TP denotes the number of true entities correctly predicted as true entities; FP represents the number of non-entities predicted as true entities; FN is the number of entities not predicted; and TN indicates the number of non-entities correctly predicted.

4.4. Result

To validate the performance of different models, this study selected IDCNN-CRF, BiLSTM-CRF, BiLSTM-Attention-CRF, and BERT-BiLSTM-CRF as comparison models. The results of these models are shown in Table 6. Due to the construction of a specialized corpus, all models performed well in recognizing journal names and their attribute entities. Among them, the BERT-BiLSTM-CRF model outperformed the other three models across all evaluated metrics, including precision, recall, F1-score, and accuracy. Compared with the IDCNN-CRF, BiLSTM-CRF, and BiLSTM-Attention-CRF models, the F1-score of the BERT-BiLSTM-CRF model improved by 6.60%, 3.98%, and 2.30%, respectively. These results demonstrate that integrating the BERT model significantly improved the model's ability to comprehend complex contexts in the journal domain, particularly in journal entity recognition.

Table 6. Entity identification results of different models.

Model	P	R	F1	ACC
IDCNN-CRF	75.62%	67.85%	69.12%	68.59%
BiLSTM-CRF	79.75%	70.42%	72.01%	71.21%
BiLSTM-Attention-CRF	80.12%	72.10%	73.69%	72.89%
BERT-BiLSTM-CRF	82.38%	73.75%	76.69%	75.19%

Figure 7 illustrates the loss curves of the BERT-BiLSTM-CRF model. train_loss represents the cross-entropy loss on the training set, which measures the difference between the model's predictions and the actual labels in the training data. test_loss refers to the cross-entropy loss on the test set, evaluating the model's performance on the test data and helping prevent overfitting. As the number of epochs increases, both train_loss and test_loss gradually decrease and approach zero, indicating that the model is training properly and performing well. In the early stages of training, the loss curves show a rapid decline. As training progresses, the model adjusts the learning rate through the Adam optimizer, gradually learning the mapping between the input data and output labels. This results in distinct phases of change in the curve. In the later stages of training, the rate of loss reduction slows down and eventually stabilizes.

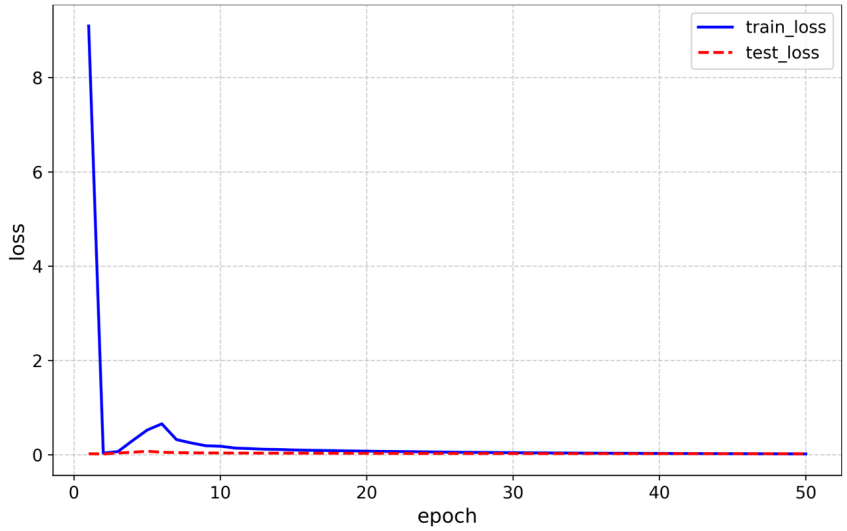


Figure 7. Change in Loss Values for Training and Test Sets.

Note that although the BERT-BiLSTM-CRF model performs well on most sentences in relation to extraction tasks, some biases still appear in the journal domains or in short-text cases. These errors are mainly categorized into two types: incorrect entity linking and incorrect triple generation, as shown in Table 7. To address these two types of errors, we implemented the following improvements: First, we increased the amount of annotated data containing similar structures to ensure that publisher entities, such as Springer, Elsevier, and IEEE, are correctly categorized as "publishers". Second, after generating the triples, we leveraged Web of Science to verify entity categories, ensuring that entities such as "Quantum" are accurately recognized as journal names.

Table 7. Errors in knowledge extraction.

Error Types	Example
Incorrect Entity Linking	"Journal of Economic Perspectives is an economics journal focusing on macroeconomic directions, published by American Economic Association.", "American Economic Association" should be recognized as a publisher entity, but the BERT-BiLSTM-CRF model incorrectly labels it as "O".
Incorrect Triple Generation	"Quantum is a physics journal focusing on quantum mechanics, published by Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften". "Quantum" should be labeled as "B-JOURNAL", but the model incorrectly predicted it as "B-DOMAIN", leading to a misclassification error.

To further evaluate the performance of JKG-LLM, we analyzed the error rate under various query scenarios. The queries were categorized as simple or complex, and we examined how factors such as query complexity, query type, and the academic disciplines involved influenced the error rate. Our experimental results demonstrate that the error rate for single-fact retrieval queries is approximately 5 errors (out of 50 examples), while the error rate for multi-step reasoning tasks rises to 12 errors (out of 50 examples). Moreover, since the majority of the datasets are from engineering disciplines, the system performs better in computer science but shows a relatively higher error rate in the social sciences, highlighting the impact of academic disciplines on system performance.

5. Conclusions

This paper presents a method for constructing a journal knowledge graph to efficiently utilize journal data. The Scrapy framework was used to collect journal datasets, followed by entity extraction with the BERT-BiLSTM-CRF framework, and the journal knowledge graph was built using Neo4j. The integration of knowledge graphs with LLMs enhances interpretability, real-time performance, and intelligence, improving the system's ability to solve complex journal-related problems.

By leveraging the journal knowledge graph as a knowledge base and integrating it with a large language model, we developed an intelligent question-answering system that improves response quality and system reliability. This work introduces novel strategies for knowledge management in the journal domain and has practical value in applications such as academic information retrieval and journal recommendation.

This paper presents a preliminary step in constructing the journal knowledge graph. However, the semi-automated approach limits efficiency, especially when processing long or complex texts. Ensuring efficient updates and expansion of the graph will be a key challenge for future research. Future work may explore integrating advanced technologies like graph neural networks to enhance reasoning and efficiency.

Author Contributions: J.N.: funding acquisition. J.Z.: conceptualization, methodology, writing—original draft. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (no. 12401593), and the Natural Science Fund of Hubei Province, China (no. 2024AFB345).

Data Availability Statement: Data are contained within the article.

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

References

1. Suchanek, F.; Kasneci, G.; Weikum, G. YAGO: A core of semantic knowledge. In Proceedings of the 16th International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 697–706.
2. Xu, B.; Xu, Y.; Liang, J.; Xie, C.; Liang, B.; Cui, W.; Xiao, Y. CN-DBpedia: A never-ending Chinese knowledge extraction system. In Proceedings of the 30th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Arras, France, 27–30 June 2017; pp. 428–438.
3. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 9–12 June 2008; pp. 1247–1250.
4. Vrandečić, D.; Krötzsch, M. Wikidata: A free collaborative knowledgebase. *Commun. ACM* **2014**, *57*, 78–85. [CrossRef]
5. Liu, Q.; Li, Y.; Duan, H.; Qin, Z. Knowledge graph construction techniques. *J. Comput. Res. Dev.* **2016**, *53*, 582–600.
6. Zhang, S.; Wang, Z.; Wang, Z. Prediction of wheat stripe rust based on knowledge graph and bidirectional long short-term memory network. *Trans. Chin. Soc. Agric. Eng.* **2020**, *36*, 172–178.
7. Wang, C.; Zhao, S.; Yan, T.; Song, S.; Ma, W.; Liu, K.; Wang, M. Hierarchical label-enhanced contrastive learning for Chinese NER. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, 1–11. [CrossRef]
8. Downey, D.; Broadhead, M.; Etzioni, O. Locating complex named entities in Web text. In Proceedings of the 20th International Joint Conferences on Artificial Intelligence, Hyderabad, India, 6–12 January 2007; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2007; pp. 2733–2739.
9. McCallum, A.; Li, W. Early results for named entity recognition with conditional random fields, feature induction and web enhanced lexicons. In Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003, Edmonton, AB, Canada, 31 May 2003; Association for Computational Linguistics: Stroudsburg, PA, USA, 2003; pp. 188–191.
10. Alex, G.; Jürgen, S.; Framewise, P. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. In Proceedings of the 2005 International Joint Conference on Neural Networks (IJCNN), Montreal, QC, Canada, 31 July–4 August 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 2047–2052.
11. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
12. Strubell, E.; Verga, P.; Belanger, D.; McCallum, A. Fast and accurate entity recognition with iterated dilated convolutions. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 2670–2680.
13. Qiu, Q.; Xie, Z.; Wu, L.; Tao, L. GNER: A generative model for geological named entity recognition without labeled data using deep learning. *Earth Space Sci.* **2019**, *6*, 931–946. [CrossRef]
14. Žukov-Gregorič, A.; Bachrach, Y.; Coope, S. Named entity recognition with parallel recurrent neural networks. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 69–74.
15. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
16. Yang, P.; Dong, W. Chinese named entity recognition method based on BERT embedding. *Comput. Eng.* **2020**, *46*, 40–45+52.
17. Chen, Y.; Qi, X.; Huang, C.; Zheng, J. A data fusion method for maritime traffic surveillance: The fusion of AIS data and VHF speech information. *Ocean. Eng.* **2024**, *311*, 118953. [CrossRef]
18. Zhu, S.; Jiang, Z.; Yan, W.; Gao, L.; Zhang, H. A knowledge graph-based intelligent planning method for remanufacturing processes of used parts. *J. Eng. Des.* **2025**, 1–28. [CrossRef]
19. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *Comput. Sci.* **2013**, *1301*, 4–8.
20. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.
21. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9–18.
22. Tunstall, L.; Beeching, E.; Lambert, N.; Rajani, N.; Rasul, K.; Belkada, Y.; Huang, S.; Werra, L.; Fourrier, C.; Habib, N.; et al. Zephyr: Direct distillation of lm alignment. *arXiv* **2023**, arXiv:2310.16944.
23. Siino, M.; Tinnirello, I. Prompt engineering for identifying sexism using GPT mistral 7B. In Proceedings of the CLEF 2024: Conference and Labs of the Evaluation Forum, Grenoble, France, 9–12 September 2024.
24. Abdin, M.; Aneja, J.; Awadalla, H.; Awadallah, A.; Awan, A.; Bach, N.; Bahree, A.; Bakhtiari, A.; Bao, J.; Behl, H.; et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv* **2024**, arXiv:2404.14219.

25. Chung, H.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.* **2024**, *25*, 1–53.
26. Baker, P. *ChatGPT Für Dummies*; John Wiley & Sons: Hoboken, NJ, USA, 2025.
27. Glm, T.; Zeng, A.; Xu, B.; Wang, B.; Zhang, C.; Yin, D.; Zhang, D.; Rojas, D.; Feng, G.; Zhao, H.; et al. ChatGLM: A family of large language models from GLM-130B to GLM-4 All tools. *arXiv* **2024**, arXiv:2406.12793.
28. Grattafiori, A.; Dubeya, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. The llama 3 herd of models. *arXiv* **2024**, arXiv:2407.21783.
29. Raiaan, M.; Mukta, M.; Fatema, K.; Fahad, N.M.; Sakib, S.; Jannat, M.M.; Ahmad, J.; Ali, M.E.; Azam, S. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access* **2024**, *12*, 26839–26874. [CrossRef]
30. Yang, H.; Liu, X.; Dan, W. FinGPT: Open-source financial large language models. *arXiv* **2023**, arXiv:2306.06031. [CrossRef]
31. Cui, J.; Li, Z.; Yan, Y.; Chen, B.; Yuan, L. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv* **2024**, arXiv:2306.16092.
32. Nie, Y.; Kong, Y.; Dong, X.; Mulvey, J.; Poor, H.; Wen, Q.; Zohren, S. A survey of large language models for financial applications: Progress, prospects and challenges. *arXiv* **2024**, arXiv:2406.11903.
33. Zhou, S.; Liu, K.; Li, D.; Fu, C.; Ning, Y.; Ji, W.; Liu, X.; Xiao, B.; Wei, R. Augmenting general-purpose large-language models with domain-specific multimodal knowledge graph for question-answering in construction project management. *Adv. Eng. Inform.* **2025**, *65*, 103142. [CrossRef]
34. Sun, J.; Xu, C.; Tang, L.; Wang, S.; Lin, C.; Gong, Y.; Ni, L.; Shum, H.; Guo, J. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024; pp. 1–9.
35. Jiang, J.; Zhou, K.; Dong, Z.; Ye, K.; Zhao, W.; Wen, J. Structgpt: A general framework for large language model to reason over structured data. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; pp. 1–7.
36. Feng, C.; Zhang, X.; Fei, Z. Knowledge solver: Teaching llms to search for domain knowledge from knowledge graphs. *arXiv* **2023**, arXiv:2309.03118,.
37. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.T.; Rocktäschel, T. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.
38. Hou, Y.; Jeffrey, R.; Liu, H.; Zhang, R. Improving dietary supplement information retrieval: Development of a retrieval-augmented generation system with large language models. *J. Med. Internet Res.* **2025**, *27*, e67677. [CrossRef]
39. Hang, C.; Tan, C.; Yu, P. MCQGen: A large language model-driven MCQ generator for personalized learning. *IEEE Access* **2024**, *12*, 102261–102273. [CrossRef]
40. Mansurova, A.; Mansurova, A.; Nugumanova, A. QA-RAG: Exploring LLM reliance on external knowledge. *Big Data Cogn. Comput.* **2024**, *8*, 115. [CrossRef]
41. Wang, J.; Guo, Y. Scrapy-based crawling and user-behavior characteristics analysis on Taobao. In Proceedings of the 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Sanya, China, 10–12 October 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 44–52.
42. Lin, H.; Wang, Y.; Jia, Y.; Zhang, P.; Wang, W. A review of knowledge fusion methods for network big data. *Chin. J. Comput.* **2017**, *40*, 1–27.
43. Tian, L.; Zhang, Z.; Zhang, J.; Zhou, W.; Zhou, X. A review of knowledge graphs representation, construction, inference and knowledge hypergraph theory. *J. Comput. Appl.* **2021**, *41*, 2161–2186.
44. Topsakal, O.; Akinci, T.C. Creating large language model applications utilizing langchain: A primer on developing llm apps fast. *Int. Conf. Appl. Eng. Nat. Sci.* **2023**, *1*, 1050–1056. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.