



Article OEQA: Knowledge- and Intention-Driven Intelligent Ocean Engineering Question-Answering Framework

Rui Zhu ¹^(D), Bo Liu ^{2,3,*}^(D), Ruwen Zhang ¹^(D), Shengxiang Zhang ¹^(D) and Jiuxin Cao ^{1,3}^(D)

- ¹ School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China; zhurui@seu.edu.cn (R.Z.); ruwen_zhang@seu.edu.cn (R.Z.); zhang_sx@seu.edu.cn (S.Z.); jx.cao@seu.edu.cn (J.C.)
- ² School of Computer Science and Engineering, Southeast University, Nanjing 211189, China
- ³ Purple Mountain Laboratories, Nanjing 211111, China
- * Correspondence: bliu@seu.edu.cn

Abstract: The constantly updating big data in the ocean engineering domain has challenged the traditional manner of manually extracting knowledge, thereby underscoring the current absence of a knowledge graph framework in such a special field. This paper proposes a knowledge graph framework to fill the gap in the knowledge management application of the ocean engineering field. Subsequently, we propose an intelligent question-answering framework named OEQA based on an ocean engineering-oriented knowledge graph. Firstly, we define the ontology of ocean engineering and adopt a top-down approach to construct a knowledge graph. Secondly, we collect and analyze the data from databases, websites, and textual reports. Based on these collected data, we implement named entity recognition on the unstructured data and extract corresponding relations between entities. Thirdly, we propose an intent-recognizing-based user question classification method, and according to the classification result, construct and fill corresponding query templates by keyword matching. Finally, we use T5-Pegasus to generate natural answers based on the answer entities queried from the knowledge graph. Experimental results show that the accuracy in finding answers is 89.6%. OEQA achieves in the natural answer generation in the ocean engineering domain significant improvements in relevance (1.0912%), accuracy (4.2817%), and practicability (3.1071%) in comparison to ChatGPT.

Keywords: intelligent QA; knowledge graph; ocean ontology; ocean engineering

1. Introduction

The intelligent question-answering (QA) system holds significant importance in various fields, including ocean engineering. The utilization of oceanic knowledge to augment the effectiveness of QA systems is still an open problem. The concept of knowledge graph (KG) originates from knowledge engineering and has been regarded as the development of knowledge engineering in the era of big data. Generally, KG can be classified into domain-specific knowledge graphs and open-world knowledge graphs. The domainspecific KG is oriented to specific fields, which are mainly used for complex application analysis or auxiliary decision-making. It has the characteristics of high expert participation, complex knowledge structure, high knowledge quality, and fine knowledge granularity. Comparatively, the open-world KG is public-oriented and does not require a professional background or industry knowledge. Ocean engineering refers to engineering projects on the development, utilization, and protection of the ocean, including project information, marine resources, human activities, etc. The ocean engineering knowledge graph project involves various subfields such as industrial sea, transportation, tourism, and entertainment. So, an ocean engineering knowledge graph is a typical domain-specific KG. Constructing such a domain-specific knowledge graph requires analyzing and clarifying the different relations between various concepts and establishing ontology.



Citation: Zhu, R.; Liu, B.; Zhang, R.; Zhang, S.; Cao, J. OEQA: Knowledgeand Intention-Driven Intelligent Ocean Engineering Question-Answering Framework. *Appl. Sci.* **2023**, *13*, 12915. https://doi.org/ 10.3390/app132312915

Academic Editor: Wenzhong Li

Received: 21 October 2023 Revised: 28 November 2023 Accepted: 29 November 2023 Published: 2 December 2023



Copyright: © 2023 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/). Currently, the QA systems in ocean engineering primarily rely on information systems that utilize relational databases. However, a large amount of useful information is usually hidden in unstructured data, such as public websites and engineering documents. Building a knowledge graph can transform the massive multiple heterogeneous unstructured data into knowledge and organize them by establishing a semantic network. Therefore, constructing a domain-specific knowledge graph has become a hot topic in intelligent ocean engineering.

As far as we know, there is still no general method for developing knowledge graphs or intelligent QA applications in ocean engineering. To develop an intelligent QA system for ocean engineering, we face the following challenges:

- How to develop a domain ontology for ocean engineering in the absence of standard knowledge definitions.
- How to extract knowledge from multi-source heterogeneous ocean engineering data.
- How to establish an intelligent ocean engineering QA system that satisfies the needs
 of professional fields and provides effective knowledge services.

To address the above challenges, we propose an intelligent knowledge graph questionanswering (KGQA) framework for ocean engineering. First, we summarize and classify the concepts and construct an ontology of ocean engineering. Second, we employ techniques in natural language processing to extract knowledge and integrate it with existing databases, resulting in the development of a KG dedicated to ocean engineering. Finally, we adopt an intent recognition model to capture user intentions and implement a keyword-matching method to fill templates. Answer entities are then retrieved from the knowledge graph by Cypher statements, and natural sentences are generated with knowledge representation of four tuples based on the answer entities.

Our main contributions can be summarized as follows:

- We integrate different types of concepts and data and establish the ocean engineering domain's ontology. Accordingly, we propose a novel approach to build knowledge graphs in the ocean engineering field based on marine text feature enhancement.
- For the first time in ocean engineering, we present an answer recommendation module based on intent recognition. The intent recognition module improves the efficiency of question parsing and achieves an answer retrieval accuracy of 89.6%.
- Based on the domain question-answering dataset we collected, a natural answer generation (NAG) scheme based on semantic fusion is proposed, which provides a general method for domain-oriented intelligent question answering.
- We define four artificial evaluation metrics related to KG for natural answer generation, including fluency, relevance, accuracy, and practicality. The results reveal that our OEQA Chatbot outperforms ChatGPT in terms of generating ocean engineering natural answers, proving the application value of the methodology proposed in this study.

The remaining parts of this paper are as follows. We provide a brief overview of the related work in Section 2. We detail our proposed OEQA framework in Section 3. Then, we present the experimental results on real-world datasets in Section 4. In Section 5, we give an application and compare it with ChatGPT. Finally, we conclude the paper and provide future works in Section 6.

2. Related Works

In this section, we summarize and organize research in two directions: the construction of domain-specific KGs and KGQA systems.

2.1. Domain-Specific KGs

Knowledge graphs can be classified into open-world knowledge graphs and domainspecific knowledge graphs. Except for open-world knowledge graphs such as Freebase [1] and DBpedia [2], most knowledge graphs are domain-specific. They effectively facilitate domain conceptualization and data management. The construction of a domain-specific knowledge graph involves two core steps [3]: (1) formalizing concepts and subsequently designing the ontology to describe the corresponding logic design; (2) mining semantic relationships among entities to represent the physical design on multiple data sources.

Domain-specific knowledge graphs have been applied in various fields such as healthcare, education, engineering, and finance, employing different approaches to design logic concepts. Gao et al. [4] integrated genotypic and phenotypic databases to construct a drug-disease knowledge graph, inferring the drug indications based on the captured genotypic and phenotypic features. Liu et al. [5] established a Chinese marine medicine ontology, leveraging Neo4j to build the knowledge graph. This framework facilitated the retrieval of the data from the Marine Traditional Chinese Medicine atlas database. To effectively manage marine accident knowledge, Fu et al. [6] constructed the knowledge graph by combining top-down and bottom-up methods. Due to the professionalism of knowledge in the field of ocean engineering, the known methods cannot be directly applicable to the KG in this field.

Knowledge graph technology plays a crucial role in the extraction and integration of diverse data sources. In a study by Liu et al. [7], a multi-data knowledge graph was constructed to detect ocean oil spills by incorporating vectors, text, and other data sources. Erik et al. [8] integrated tabular data and SPARQL endpoints to build an ecotoxicology knowledge graph, improving the prediction of ecotoxicological effects. Ahmad et al. [9] leveraged public KG resources related to COVID-19 information from Scientific Open Data and Publications, employing it to analyze the drug interactions. These studies underscored that most domain-specific knowledge graphs are composed of multiple data. However, extracting valuable knowledge from such data is a complex task and is faced with domainspecific challenges, particularly extracting entities and their underlying relations when dealing with unbalanced data.

In summary, knowledge graph construction methods are divided into a top-down approach (ontology layer to data layer) [5–8] and a bottom-up approach (data layer to ontology layer) [4,9]. The top-down method involves building concepts and gradually refining them with data, while the bottom-up method organizes inductive entities and gradually abstracts the upper-level concepts. The ontology-guided knowledge extraction process is more suitable for constructing domain knowledge graphs. However, due to the diverse data patterns and complex application requirements in the real world, previous works usually build knowledge graphs by designing specific frameworks to manually construct knowledge graphs. Our approach differs from their work in two key aspects. First, our work emphasizes a balanced approach between automation and human participation for the rapid construction of domain knowledge graphs. Second, we focus on the extraction of heterogeneous multi-source data, which presents a more complex and informative scene. Therefore, we introduce a data-augmentation method to facilitate the process from domain ontology to a data layer of knowledge. Moreover, given the absence of a mature ontology structure and knowledge network in the field of ocean engineering, our research contributes to advancing domain expansion and the practical application of knowledge graphs in engineering.

2.2. KGQA Systems

In recent years, KGQA systems have emerged as a means of utilizing KGs as knowledge bases to meet the query needs of users. There are three different types of approaches for KGQA: template construction, semantic parsing, and information retrieval. The template construction-based methods are realized by predefining question templates and query statements. The natural question is matched with the question template, and then the answer to the question is queried in the knowledge graph according to the query statements. The template construction-based methods offer several advantages, including fast response times, high precision, and suitability for domain-specific knowledge questionanswering scenarios. However, the construction of templates requires a significant amount of manpower, making it challenging to use in open-domain question answering.

The semantic parsing-based methods parse natural questions into corresponding logical expressions and then map them into structured queries to obtain answers from the KG. Early semantic parsing relied on manually constructed lexicons, syntactic templates, and grammar rules to train parsers. It was typically limited to specific domains and specific logical forms, lacking domain adaptability. With the rapid development of deep learning, neural networks have gradually been applied to semantic parsing tasks. Dong et al. [10] proposed a model based on an encoder-decoder architecture, utilizing LSTM to encode the input question and decode the logical form. Shao et al. [11] discovered that the beam width size in sequence-to-sequence and beam search methods has an impact on the accuracy of the model. They introduced a logical form reordering model based on graph attention and Transformers to capture hierarchical structural information of logical forms. The limited quantity and high cost of annotated logical form data have become a bottleneck for supervised semantic parsing models. Researchers have started exploring weakly supervised semantic parsing methods, considering learning a semantic parser using only question-answering pairs. This approach often maps the questions to graph-structured logical forms. As the complexity of the questions increases, the search space for query graphs becomes significantly large. Yih et al. [12] transformed the process of semantic parsing into the generation of query graphs and proposed the STAGG framework for staged query graph generation. Lan et al. [13] proposed an improved staged query graph generation method. They utilized a beam search algorithm and semantic matching models to expand the core inference chain while adding constraints during the expansion process, rather than adding constraints after generating the core inference chain. Chen et al. [14] adopted a two-stage approach for query graph generation. In the first stage, an encoderdecoder model is employed to predict an abstract query graph for the question. The abstract query graph is then used to constrain the generation of candidate query graphs, preventing the generation of noisy candidates. In the second stage, the candidate query graphs are ranked to obtain the best query graph. Jia et al. [15] proposed a two-stage ranking model to select the optimal query graph. Initially, they selected the top k candidates from the candidate query graphs, and then they further ranked the top k candidates based on answer type information, ultimately selecting the best query graph. However, supervised semantic parsing models face limitations due to the scarcity and high cost of annotated logical form data. Weakly supervised methods that use only question-answer pairs have been explored, but they face challenges in convergence when dealing with large search spaces for query graphs.

Information retrieval-based KGQA retrieves question-related subgraphs from a KG based on the information contained in the question and then searches for answers within these subgraphs. Sun et al. [16] proposed the GRAFT-Net method, which utilizes a personalized PageRank algorithm to expand from the topic entity to surrounding entities and obtain a subgraph of the KG. They also retrieved sentences related to the topic entity and added them to the subgraph, forming a heterogeneous graph. By classifying the entity nodes in the heterogeneous graph, they obtain the answers to the questions. Similarly, Yan et al. [17] also employed the personalized PageRank algorithm to retrieve subgraphs. The GRAFT-Net method constructs subgraphs based on heuristic rules, leading to the inclusion of many irrelevant entities and relations. To address this issue, Sun et al. [18] proposed the PullNet method, which dynamically iterates to construct small-scale subgraphs with high recall, making the answer selection process easier. Lu et al. [19] pruned the questionrelated subgraphs to reduce their size. They employed two pruning strategies during the training process to remove incorrect answer paths. Zhang et al. [20] introduced a trainable subgraph retriever. Starting from the topic of entity, they performed a beam search at each step to select the top k relation expansions to extend the current subgraph. Experimental results show that their subgraph retrieval method outperforms the PullNet method. Jiang et al. [21] unified subgraph retrieval and subgraph reasoning in their UniKGQA model. It consists of two modules: semantic matching and matching information propagation. They used a unified model architecture to learn parameters. The model was pre-trained on the semantic matching task and then fine-tuned separately on subgraph retrieval and subgraph reasoning tasks. However, subgraph retrieval faces challenges including dealing with large-scale subgraphs and achieving high recall rates. Within the retrieved subgraphs, there may be multiple potential answer paths, requiring careful analysis and reasoning to select the correct path and extract accurate answers.

In conclusion, the aforementioned methods exhibit their respective merits and limitations. Template-based methods are efficient and accurate for specific domain Q&A scenarios, but they necessitate substantial human effort in constructing the templates. In comparison to template-based methods, semantic parsing-based methods offer greater flexibility. However, supervised semantic parsing requires a substantial amount of manually annotated logical form data to train the semantic parser, which is challenging and costly in terms of annotation efforts. Additionally, due to the lack of intermediate supervision signals during the training process, weakly supervised semantic parsing encounters difficulties in achieving model convergence when dealing with large search spaces. Information retrievalbased methods often retrieve subgraphs from the KG, which can effectively narrow down the search space of the model. However, determining the appropriate size of the subgraph is a challenge. A subgraph that is too small may lead to decreased query accuracy, while a subgraph that is too large can lead to difficulties in model convergence. In contrast, the approach presented in this paper exhibits distinct advantages. Unlike the previous works, we introduce an intent-guided template construction method, where query templates are constructed based on predefined intent categories for the ocean engineering Q&A scenario. By filling in templates, convert user questions into query statements, and retrieve answers from the knowledge base. Compared with existing methods, this approach not only avoids the labor-intensive process of manual template construction but also mitigates issues such as search space explosion and difficulty in convergence. Furthermore, it is suitable for domain-specific Q&A tasks and exhibits high accuracy.

3. The Method of OEQA

Our OEQA framework consists of five modules: ontology construction, data collection and processing, knowledge extraction, knowledge storage, and the intelligent QA system. Figure 1 depicts the overview of the OEQA framework.



Figure 1. The framework of our OEQA system.

3.1. Data Preparation

3.1.1. Data Collection and Processing

Knowledge graph sources in this domain include internal and external data. Internal data are derived from ocean engineering business data, while external data mainly include marine encyclopedias and related websites. These various data exist in the form of structured, semi-structured, and unstructured data.

- 1. Structured data are obtained from a relational database that has been standardized. However, some records may be unsuitable for experimentation due to inaccuracies or meaningless. These invalid records must be corrected or deleted to ensure accurate results.
- 2. Semi-structured data are gathered through web crawling technology from various marine-related websites. This type of data needs to be cleaned to avoid data noise.
- 3. Unstructured data mainly consist of PDF or WORD reports, which extract text semantics by natural language processing techniques. Unstructured data are the base of knowledge extraction, which needs domain experts to annotate entities and relations.

3.1.2. Ontology Construction

Ontology describes the relations and semantics between concepts in a certain domain and standardizes the representation of knowledge in the domain [22]. Although fully automated ontology establishment is efficient, errors will remain without expert supervision. Therefore, we invite the domain experts to manually develop the concept of relation in ontology and translate the table structure of the marine information system database to classes, attributes, and relations using the seven-step process [23]. In the process of ontology construction, one part of the ontology is manually extracted concepts and relations in the field of ocean engineering by experts, while the other part automatically extracted structured data from the existing ocean information database.

The concrete steps are as follows: (1) determine the ocean engineering field ontology scope; (2) examine the possibility of reusing the existing ontology; (3) collect the terms from the ocean engineering field; (4) define classes and class hierarchy; (5) define the attributes of the class; (6) define concept relationships; (7) create instances. The constructed ontology concentrates on ocean engineering and the presets of the 10 Things by Protégé [24] tool. Figure 2 depicts part of the ontology structure.



Figure 2. Structure of ocean engineering domain ontology.

3.2. Marine Text Knowledge Extraction with Data Augmentation

3.2.1. Named Entity Recognition

Entities that are either nested inside of or connected to another entity are called nested entities. Compared to other fields, the challenge of the NER work in ocean engineering is that there are numerous nested entities and ambiguous boundaries of named entities. We propose a data augmentation approach to balance the samples and adopt the RoBERTa [25] model based on MRC to enhance the context features.

The data augmentation approach is based on biological genetic variation, inspired by Hu et al. [26], to avoid being affected by the imbalance of positive and negative samples and to learn entity boundaries more effectively. Here, *n* represents the length of the input sequence and *Y* represents all the named entity label types (e.g., Project, Coordinates, Organization, etc.) in the dataset given an input sequence $X = \{x_1, x_2, ..., x_n\}$ and named entity label $y \in Y$ that corresponds to it.

As shown in Figure 3, each input sequence and named entity in the training set is regarded as a chromosome and a gene inside the chromosome, respectively. The named entity e_y randomly chosen in the sequence X is replaced with a marker [*GENE*]. Then, we regard X as X_y and store it in the chromosome library C_y . Similarly, the eliminated named entity e_y is stored in the gene bank G_y .



Figure 3. A sample of the chromosome containing several gene types.

The process of balancing samples is regarded as a heritable variation in biology. For example, when lacking a positive sample of the Project label, a chromosome $X_{Project}$ is randomly selected from the chromosome library $C_{Project}$, which is regarded as the self-replication of the chromosome. A gene $e_{Project}$ is extracted from the gene bank $G_{Project}$ and inserted into the corresponding [*GENE*] in $X_{Project}$ to form a new positive sample, which is regarded as a chromosomal genetic mutation. The method assures that the dataset can be expanded without destroying the original entity boundary and balances positive and negative samples to enhance the learning of contextual features.

We adopt the RoBERTa-MRC model to recognize named entities based on the augmented dataset. Firstly, construct a natural label description p_y for each label y; then, each sample is converted into a triplet (p_y , e_y , X). We concatenate p_y and X into a sequence as input to the RoBERTa model:

$$[CLS]p_{y}[SEP]X[SEP] \tag{1}$$

Then, the model predicts the probability p_{start} of each token as a start index according to the matrix $E \in \mathbb{R}^{n \times d}$ output from RoBERTa, and *d* represents the dimension of the last layer of RoBERTa. The start and end index predictions are as follows:

$$p_{start} = softmax_{each row}(E \cdot W_{start}) \in \mathbb{R}^{n \times 2}$$
(2)

$$p_{end} = softmax_{each row}(E \cdot W_{end}) \in \mathbb{R}^{n \times 2}$$
(3)

where $W_{start} \in \mathbb{R}^{d \times 2}$ and $W_{end} \in \mathbb{R}^{d \times 2}$ are the weights to learn. By applying argmax() to each row of p_{start} and p_{end} , we will obtain the predicted indexes that might be the start or end positions:

$$I_{start} = \{i | argmax(p_{start}^{(l)}) = 1, i = 1, 2, \dots, n\}$$
(4)

$$I_{end} = \{i | argmax(p_{end}^{(i)}) = 1, i = 1, 2, \dots, n\}$$
(5)

Given any start index $si \in I_{start}$ and end index $ei \in I_{end}$, the model predicts the probability p_{s_i,e_i} of matching through the binary matrices E_{s_i} and E_{e_i} :

$$p_{s_i,e_i} = sigmoid(w \cdot concat(E_{s_i}, E_{e_i}))$$
(6)

Here, $w \in \mathbb{R}^{1 \times 2d}$ is the weight to learn.

3.2.2. Relation Extraction

The relation extraction (RE) task aims to extract semantic relations between entities in the text based on the named entities recognition module. We utilize the R-BERT [27] model that leverages head and tail entity positions for improvisation. In the data preprocessing step, the special tokens '\$' and '#' are added to the left and right sides of the target entities, respectively, which mark the location information of the entities. For each sentence, the input of R-BERT is a concatenation of [*CLS*], [*SEP*], \$, #, and the sentence. The input is shown in Formula (7).

$$x_1, x_2, \dots, x_j, \$e_1\$, x_{j+k}, \dots, x_i, #e_2\#, x_{i+t}, \dots, x_n[SEP]$$
(7)

Here, n represents the length of the input sequence, k and t represent the length of the target entities.

$$h'_0 = W_0[tanh(h_0)] + b_0 \tag{8}$$

$$h'_{1} = W_{1}[tanh(\frac{1}{k}\sum_{l=j}^{j+k}h_{l})] + b_{1}$$
(9)

$$h'_{2} = W_{2}[tanh(\frac{1}{t}\sum_{l=t}^{i+t}h_{l})] + b_{2}$$
(10)

Then, R-BERT inputs the concatenation of the hidden layer vector h_0' of [*CLS*] and the average value of the hidden layer vector h_1' and h_2' of the entity e_1 and e_2 into the softmax layer for relation classification. Matrices W_0 , W_1 , and W_2 have the same dimensions and b_0 , b_1 , b_2 are the parameters.

$$h'' = W'[concat(h_0', h_1', h_2')] + b'$$
(11)

$$p' = softmax(h'') \tag{12}$$

Here, W' is the weights to learn, b' is bias vectors, and p' is the probability output.

3.3. Knowledge Graph Construction

We choose the Neo4j [28] graph database to store ocean engineering knowledge as it offers better application performance compared to relational databases. The method used in this paper to develop a KG combines manual establishment by domain experts with semiautomatic tools. Entities are represented as nodes, with entity attributes serving as node attribute values and relations between entities as edges. Neo4j can efficiently handle large data while supporting the inference of KG. We develop an ocean engineering knowledge graph, which contains nine types of entities and eight types of relations, comprising 1681 entities and 5811 relations. Figure 4 is the representation in Neo4j.



Figure 4. Partial display of entities and relations ocean engineering knowledge graph in Neo4j.

3.4. Answer Recommendation Based on Intent Recognition

The main function of the answer recommendation module is to analyze user questions in the field of ocean engineering, retrieve the correct answer entities from the knowledge graph, and generate the natural answer. It is composed of three modules: intent recognition, question parsing, and natural answer generation. Figure 5 demonstrates the process.



Figure 5. The flowchart of answer recommendations based on intent recognition.

3.4.1. Intent Recognition

The efficacy of answer recommendation depends on its ability to cater to the requirements and goals of users. We adopt a user-centric approach by discerning the inquiry intentions of users and subsequently seeking answers that match the intentions.

Intent recognition is commonly regarded as a text classification task [29]. UIUC [30] is a widely used standard for question classification, which classifies questions into six categories and fifty subcategories based on the answer types. Building on UIUC and concerning real user QA scenarios in the ocean engineering field, we predefine 11 user intention types, including Project-Overview, Project-Location, Project-Zonghai, SeaUse-Location, SeaUse-Type, SeaUse-Mode, ConfirmRight-Way, UseRight-Holder, SeaUse-Nature, SeaArea-Class, and SeaUse-Area, based on the ocean engineering knowledge graph.

We implement the BERT-TextCNN model to identify the user intentions and guide subsequent answer recommendations. As illustrated in Figure 6, the user question is input into the BERT layer to represent each token in the text as a vector. The vectors are then fed into TextCNN [31], which comprises a convolutional layer, a pooling layer, and a fully connecting layer. Ultimately, the model outputs the intent category to which the user's question belongs.



Figure 6. BERT-TextCNN-based intent recognition model.

We obtain the semantic vector representation of the question through BERT. Given a question *q* with length *L*, for i = 1, 2, 3, ..., L, each token in *q* is represented by x_i . We represent the question *q* as a sequence $q' = [CLS], x_1, x_2, x_3, ..., x_i, [SEP], x_{i+1}, ..., x_L, [SEP]$, which is input to BERT. Based on this, we obtain the final output of the last hidden layer of BERT(*H*), where $h_i \in \mathbb{R}^{1 \times k}$ is a *k*-dimension vector and $H = (h_1, h_2, h_3, ..., h_L)$,

$$H = BERT(q') \tag{13}$$

We concatenate *j* hidden states $h_i, h_{i+1}, h_{i+2}, ..., h_{i+j-1}$ and represent them with $h_{i:i+j}$. For t = 1, 2, 3, ..., m, a *filter*_t $\in \mathbb{R}^{j \times k}$ is used to convolve with a window of *j* hidden states. Here, *b* is a bias term and *f* is an activation function. A feature v_t^i is generated from a window of $h_{i:i+j-1}$ by:

$$v_t^i = f(filter_t \cdot h_{i:i+j-1} + b) \tag{14}$$

To learn the relation and characteristics between the context of different dimensions in the question, this filter is applied to each possible window of hidden states $\{h_{1:j}, h_{2:j+1}, h_{3:j+2}, \ldots, h_{L-j+1:L}\}$ to produce a feature map:

$$V_t = [v_t^1, v_t^2, v_t^3, \dots, v_t^{L-j+1}]$$
(15)

where $V_t \in \mathbb{R}^{L-j+1}$. For each feature map, we capture the maximum value as the most obvious feature locally by applying a max-pooling operation. For t = 1, 2, 3, ..., m,

$$c_t = \max(V_t) \tag{16}$$

$$C = [c_1, c_2, c_3, \dots, c_m]$$
(17)

Finally, we implement the classification through an MLP layer.

3.4.2. Question Parsing

User questions in the field of ocean engineering are diverse and complex. The user questions belonging to the same intent category often exhibit similar semantic characteristics. We analyze the semantic information of the user question and map it into Cypher based on the intent recognition module. The Cypher templates are constructed based on the semantic attributes of each intent category and filled through keyword matching. The general form of the Cypher template is shown in Equation (18):

$$match(e_{h} : EntityCategoryA\{name : EntityNameA\}) - [r : Rel] \rightarrow (e_{t} : EntityCategoryB)$$
(18)
return e_t

Here, e_h represents the head entity, e_t represents the answer entity, Rel represents the relation between the head entity and the answer entity, EntityCategoryA and EntityCategoryBrepresent the head entity and the answer entity category, and EntityNameA represents the name of the head entity. To fill the Cypher templates, we identify entities and relations from the user questions through keyword matching. The Aho–Corasick algorithm [32] is a classic multi-pattern string-matching algorithm based on the Trie and the KMP. Given a string *S* and multiple pattern strings P_1 , P_2 , P_3 , P_4 , the Aho–Corasick algorithm finds all the pattern strings that appear in the *S* at once. Hence, we construct an entity and relation dictionary in the field of ocean engineering, respectively. Since the relations may be described variously, a relation dictionary involves not only incorporating all the relations in the knowledge graph but also supplementing with their synonyms. We adopt the Aho–Corasick algorithm for question parsing:

- 1. Construct a dictionary tree and establish the basic automaton based on the entity and relational dictionaries in the field of ocean engineering.
- 2. Input the user question into the automaton as the main string for pattern matching.
- 3. Match the entity and relation names that appear in the question to obtain output.

When the intent category of user questions is determined, the answer entity category (*EntityCategoryB*) is determined accordingly. We fill the template with the output entity and relation names (*EntityNameA* and *Rel*) and corresponding entity category (*EntityCategoryA*). So, we obtain a complete Cypher statement by filling in the Cypher template. Subsequently, the Cypher statement is executed as a query in the Neo4j to retrieve the corresponding answer entity.

3.4.3. Four Tuples-Based Natural Answer Generation

We generate the natural answers based on the user questions and corresponding answer entities in this section. Natural language generation pre-training models such as GPT3 [33], BART [34], and T5 [35] have superior performance currently. Unfortunately, these models are trained mostly in English or multilingual languages. As a result, we adopt a Chinese pre-training language model named Text-to-Text Transfer Transformer Pegasus (T5-Pegasus) to effectively generate the natural answers.

Four tuples are made up of a user question q, head entity e_h , relation r, and answer entity e_t . We divided the NAG task into two steps. Firstly, combine the triple (e_h, r, e_t) obtained in Section 3.4.2 with the user question q as model input (q, e_h, r, e_t) to help the model understand user semantics. Secondly, transfer the four tuples into natural answer sentences *ans* with T5-Pegasus.

4. Experiments

In this section, we evaluate the performance of the proposed OEQA on real datasets. We introduce the datasets first and then present the experiment results.

4.1. Dataset

Our approach is evaluated on the following datasets.

OceanNER is used for named entity recognition. It is extracted from unstructured data derived from text documents in the field of ocean engineering. Some experts are invited to annotate the named entities in BIO format. Referring to the "Classification Standard for the Use of Sea Areas of the People's Republic of China (HY/T123-2009) [36]", the 870 samples are divided into 9 categories. Here, UseType and UseMode represent the sea area used type and mode of project, respectively. UseType and UseMode are defined by two levels, UseMode_A represents level one and UseMode_B represents level two. The same definition method is also for UseType_A and UseType_B. The dataset contains a total of 2580 entities, with nested entities in the total numbering 75.93%. The experiment divides the dataset into a training set, verification set, and testing set in a 6:2:2 ratio.

OceanRE is used for relation extraction. It contains 1288 samples, divided into 1030 samples for training and 258 samples for testing. The dataset contains 8 relation types. Each example is a sentence annotated for a pair of entities and the corresponding relation type in this dataset.

OceanQA consists of 2746 user questions collected from real ocean engineering scenarios. It is divided into three parts: OceanQA1, OceanQA2, and OceanQA3. We extracted 1210 pieces of data from OceanQA and manually labeled the intent categories to constitute OceanQA1. OceanQA2 consists of 1036 pieces of user questions with collected corresponding natural answers. OceanQA1 and OceanQA2 are both divided into training sets, verification sets, and testing sets with a ratio of 6:2:2, and they are used for intent recognition and answer generation models, respectively. The remaining 500 pieces of data constitute OceanQA3, which is used to evaluate the accuracy of answer recommendations.

4.2. Knowledge Extraction

4.2.1. The Experiment of Named Entity Recognition

The positive and negative samples in OceanNER are extremely imbalanced. According to Table 1, the proportions of named entities, except the Project type, do not approach 1.0, indicating that the positive and negative samples needed to be balanced.

Entity Type	Scale (Nested Proportion)	Positive Samples (Number)	Negative Samples (Number)	Positive to Negative Samples (Rate)
Project	575 entities (22.29%)	231	226	1.0221
Location	207 entities (8.02%)	114	343	0.3324
Organization	279 entities (10.81%)	109	348	0.3132
Coordinates	82 entities (3.18%)	17	440	0.0386
Area	539 entities (20.89%)	182	275	0.6618
UseType_A	160 entities (6.20%)	89	368	0.2418
UseType_B	201 entities (7.79%)	145	313	0.4633
UseMode_A	126 entities (4.88%)	63	394	0.1599
UseMode_B	411 entities (15.93%)	109	348	0.3132
Total	2580 entities (75.93%)	1059	3055	0.3466

Table 1. The statistics of named entities in OceanNER.

In practice, entities are presented in flat and nested forms. Nested entities usually contain semantic relations of multiple entities with blurred entity boundaries (such as Project, UseType, etc.). Therefore, the testing set randomly picks 0% to 70% of the nested entities to comprehensively evaluate the model. Figure 7 demonstrates that when the nested entity does not exist, the model F1 score is 91.98%; when the nested entity increases

to a proportion of 70%, the F1 score is down to 67.52%. When the proportion of nested entities is less than 20%, the entity boundaries are easy to identify; when the proportion is more than 20%, the model check-all rate drops. The model is weak at identifying entity boundaries but great at correctly categorizing entity classes, as seen by the reasonably steady overall precision. Generally, the results can meet the current needs of relation extraction in this domain. As we can see, natural label description enables the model to learn correlations between texts and labels to predict more accurately.



Figure 7. The performance of NER with different nested percent.

4.2.2. The Experiment of Relation Extraction

The relation extraction results of the R-BERT model are shown in Table 2. Precision, recall, and F1 score are used as evaluation metrics. The model obtains remarkable results, with the macro F1 score reaching 98.63%. According to our analysis, the reason for the highest score of Project–Coordinates is that samples have a single description, which is easy to identify. However, there are three relation types involving the UseType_B entity, which are not easy to distinguish by the model.

Table 2. Relation extraction results between relative entities.

Relation Type	Precision	Recall	F1 Score
Project-Location	0.9741	1.0000	0.9869
Project-Area	0.9800	0.9700	0.9750
Project-Coordinates	1.0000	1.0000	1.0000
Project-UseType_B	0.9667	0.9822	0.9743
Project-UseMode_B	0.9875	0.9869	0.9872
Project-Organization	0.9822	1.0000	0.9910
UseType_A-UseType_B	0.9913	1.0000	0.9956
Other	0.9909	0.9699	0.9803
Macro avg	0.9841	0.9886	0.9863

4.3. Intelligent QA Module

4.3.1. The Experiment of Intent Recognition

We use the OceanQA1 dataset to evaluate the Intent Recognition model. We choose precision, recall, and F1 score as the evaluation metrics of intent recognition. As shown in Table 3, in each category, the three metrics have achieved relatively high values, and the average classification precision of 11 intent categories is 94.50%, with an average recall rate of 94.02% and an average F1 score of 94.16%. The results indicate that the intention recognition model can classify user questions in the ocean engineering field effectively.

Intent Category	Precision	Recall	F1 Score
Project-Overview	0.9737	0.9024	0.9367
Project-Location	0.8889	0.9756	0.9302
Project-Zonghai	0.8814	0.9630	0.9204
SeaUse-Location	0.9714	0.9189	0.9444
SeaUse-Type	0.9821	0.9167	0.9483
SeaUse-Mode	0.9286	0.9750	0.9512
ConfirmRight-Way	0.9565	0.9167	0.9362
UseRight-Holder	0.9787	0.9583	0.9684
SeaUse-Nature	0.9355	0.9667	0.9508
Sea-Class	0.9756	0.8889	0.9302
SeaUse-Area	0.9231	0.9600	0.9412
Macro avg	0.9450	0.9402	0.9416

Table 3. Results of the intent recognition experiment.

4.3.2. The Experiment of Answer Recommendation

We utilize the OceanQA3 dataset containing 500 user questions to evaluate the accuracy of answer recommendations. The experimental results, shown in Table 4, list the number of users' questions identified in each intention category, the number of questions correctly answered, and the corresponding accuracy.

Table 4. Results of answering the recommendation experiment.

Intent Category	Classified Number	Answered Number	Accuracy
Project-Overview	44	41	0.9318
Project-Location	42	42	1.0000
Project-Zonghai	39	38	0.9743
SeaUse-Location	41	40	0.9756
SeaUse-Type	40	40	1.0000
SeaUse-Mode	43	41	0.9535
ConfirmRight-Way	41	41	1.0000
UseRight-Holder	42	41	0.9762
SeaUse-Nature	42	42	1.0000
Sea-Class	42	41	0.9762
SeaUse-Area	43	41	0.9535
Total	459	448	0.9760

Questions that fail to recognize the intention cannot obtain the correct answer entities due to the strong coupling between the intent recognition module and the answer recommendation module. Answer recommendation is based on intention recognition, and the accuracy of answer recommendation varies with different intentions. For example, for intent categories such as SeaUse-type and ConfirmRight-Way, answer accuracy could reach 100% because of their explicit semantics. In contrast, the accuracy of the Project-Overview category is the lowest at 93.18% because the summative description is difficult to extract clear information. As shown in Table 4, 4459 questions are correctly classified by the intent recognition module, while 41 questions fail to be recognized. Out of the correctly classified questions, 448 questions were answered correctly. Since questions that can be correctly classified into the intent category have a 97.6% probability of obtaining the correct answer entity, the effectiveness of intention recognition in answering recommendations is shown. Overall, 448 of the 500 test questions obtained the correct answer—an accuracy of 89.6%.

4.3.3. The Experiment of Natural Answer Generation

The experiment chose BLEU [37], Rouge [38], Meteor [39], and Distinct [40] to fully evaluate the authenticity, fluency, and diversity of the generated text. Each metric with the average value is used as the final result. Table 5 shows the randomly drawn samples from the testing set.

Table 5. Samples of natural answer generation.

Sample	Natural Answer Output
Where is the location of theJiangsu Dafeng200 MW WindPower Project?Dafeng District,Yancheng CityYancheng City	Jiangsu Dafeng 200 MW Wind Power Project is located in Dafeng District, Yancheng City, Jiangsu Province. The project is equipped with 200-megawatt wind turbines, which can provide a large amount of clean electricity for the local area, and can also promote the development of clean energy in the local area.
What kind of sea method does the Jiangsu Xiangshui China Three Gorges 12.5 MW Test Fan Project use? A permeable structure and a submarine cable pipeline	Regarding the sea-based method of Jiangsu Xiangshui China Three Gorges 12.5 MW Test Fan Project, the project adopts two methods: permeable structures and submarine cable pipelines. The function of permeable structures is to protect the coastline and prevent waves from eroding the seawall. Submarine cable pipelines deliver the power generated by wind power generators to the land power grid. While protecting the marine environment, this sea-based method can also make full use of offshore wind energy resources.

Table 6 shows the results of the four indicators. BLEU-4 and Rouge-l are considered to be the correspondence between the generated answers and the original answers. The low score of BLEU-4 is due to some synonyms, or similar expressions are not considered. Meteor considers word morphology, synonyms, and others, which obtains an obvious score of 0.5704. It shows the generation of natural language answers can vary to a certain extent according to the known vocabulary and retain a high consistency with the reference sentences. Distinct emphasizes the diversity of the generated natural answer. A Distinct score of 0.3840 is ideal, owing to the large overlapping phrases between the answer and the head entity in the dataset, reducing the diversity of generated answer sentences.

Table 6. Evaluation of natural answer generation.

BLEU-4	Rouge-l	Meteor	Distinct
0.3457	0.5498	0.5704	0.3840

5. Case Study

5.1. The Application of OEQA

We design an intelligent QA system named OEQA Chatbot for the ocean engineering field. The system identifies the intent of the user question and returns natural answers. We compare OEQA Chatbot with ChatGPT to fully evaluate the QA performance. In total, 100 questions are selected at random from OceanQA2, as well as the corresponding natural answer generated by OEQA Chatbot and ChatGPT.

Table 7 demonstrates the details by bolding the keywords of users' questions, underlining the answer entities in OEQA Chatbot, and marking the incorrect replies in ChatGPT with wavy lines. Here, we select the better answer of ChatGPT and compare it with the OEQA Chatbot. The aspects of sea mode, geographic location, and sea nature were taken into consideration as we compared the generated responses from OEQA Chatbot and Chat-GPT. The underline shows the correct knowledge answer entities that the OEQA Chatbot retrieved. Instead, ChatGPT delivered inaccurate or irrelevant responses. For instance, "the test wind turbine is located" is an incorrect answer, while "issued in 2010" is an erroneous factual answer. Despite "still needs to consult the relevant local departments or the most recent relevant policies and regulations." being practical, the expert review found a lack of guidance.

[INPUT] Question	What Is the Sea Area Used Mode of Jiangsu Xiangshui China Three Gorges 12.5 MW Wind Turbine Test Project?		
Correct Answer Entities	permeable structures, submarine cable pipelines		
	OEQA Chatbot	Jiangsu Xiangshui China Three Gorges 12.5 MW Wind Turbine Test Project adopts two methods of permeable structures and submarine cable pipelines as the sea area used mode. The role of the permeable structure is to protect the coastline and prevent waves from eroding the seawall, and the submarine cable pipeline is to transmit the electrical energy generated to the land grid by the wind power generators. This way can not only protect the marine environment but also make full use of offshore wind energy resources to achieve efficient use of clean energy.	
[OUTPUT] Instances	ChatGPT	The sea area used mode of Jiangsu Xiangshui China Three Gorges 12.5 MW Wind Turbine Test Project should refer to the sea area where the test wind turbine is located. According to the "Regulations on the Administration of the Use of Sea Areas" issued in 2010 and subsequent relevant regulations, wind power belongs to the sea for production and operation. Therefore, it can be judged that the sea used by Jiangsu Xiangshui China Three Gorges 12.5 MW Wind Turbine Test Project belongs to the sea for production and operation. However, the specific way of using the sea still needs to consult the latest relevant policies and regulations or the relevant local departments to obtain accurate answers.	

Table 7. Samples of natural answer generation for OEQA Chatbot and ChatGPT.

Correct knowledge answer entities are underlined; Incorrect knowledge answer entities are highlighted with a wavy underline.

Due to the one-sidedness of the automatic evaluation, we designed four manual evaluation metrics, including Fluency, Relevance, Accuracy, and Practicality [41].

Definition 1. Fluency refers to whether the generated natural answer is smooth and fluent and corresponds to the language habits, which checks the grammar of the sentence.

Definition 2. Relevance refers to whether the generated natural answer is related to the input question. Relevance can be assessed by checking whether the generated natural answer contains the answer entities, is coherent with the context, and so on.

Definition 3. Accuracy refers to whether the generated natural answer corresponds to the question and contains all of the correct information.

Definition 4. *Practicality refers to whether the generated natural answer is useful to the questioner.*

We found three domain experts to anonymously score the sentences on a scale of 0~5 based on the given definitions. The metrics with the higher score indicate better-generated effects. Figure 8 shows the average results of the manual evaluation of the systems.



Figure 8. Manual evaluation of the results of answering generation.

The results revealed that both systems generate fluent natural answers, although ChatGPT obtains a higher relative score of 4.9683 on Fluency. ChatGPT is advantageous in terms of Fluency, which benefits from a large number of parameters. Additionally, OEQA Chatbot achieves improvements of 1.0913, 4.2817, and 3.1071 in Relevance, Accuracy, and Practicality, respectively. The reason for the analysis is that the stored knowledge in the OEQA Chatbot is more fine-grained in this field. In contrast, ChatGPT has the problem of an inaccurate understanding of the user's intentions and provides precise answers and guidance. Therefore, the OEQA Chatbot has a good performance in intelligent question-answering in the ocean engineering field.

5.2. Discussion and Limitations

Although OEQA Chatbot performs well in the field of marine Q&A, we acknowledge that the system has a few limitations. As an end-to-end model, the OEQA Chatbot is influenced by the effects of its three components. In our experiments, to minimize the impact of user ambiguous intention as much as possible, intents are divided into subclasses determined by the average score from three experts. Despite us taking these measures, the OEQA Chatbot may still struggle to capture the correct intent, leading to inaccuracies in the feedback of answer entities and a subsequent decline in system accuracy.

6. Conclusions and Future Work

This paper proposed a novel framework to construct an ocean engineering KG and develop an intelligent QA system. We first built an ocean engineering knowledge graph based on the constructed ontology and extracted entities and relations from heterogeneous data. Then, we designed an answer recommendation system based on intention recognition, which can retrieve answer entities from the KG and generate corresponding natural answers. Finally, we proved that the OEQA Chatbot outperforms ChatGPT in Relevance, Accuracy, and Practicality by questioning the ocean engineering field. Our research and system demonstrated the effectiveness of OEQA in ocean monitoring and management.

In future work, considering that the efficacy of the OEQA relies on the performance of the user's question intention recognition module for accurate answer generation, we may leverage the capabilities of large language models as a potential avenue to improve the OEQA system's language understanding and response generation. Furthermore, we plan to regularly update the ocean engineering knowledge graph with new data sources to guarantee that the system stays current with the latest developments in the field.

Author Contributions: Study conception and design: R.Z. (Rui Zhu), B.L., R.Z. (Ruwen Zhang), S.Z. and J.C.; data collection and experiments: R.Z. (Rui Zhu), R.Z. (Ruwen Zhang) and S.Z.; analysis and interpretation of results: R.Z. (Rui Zhu), R.Z. (Ruwen Zhang) and S.Z.; draft manuscript preparation: R.Z. (Rui Zhu), B.L., R.Z. (Ruwen Zhang) and S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by National Key R&D Program of China under Grants No. 2022YFB3104300. National Natural Science Foundation of China under Grants No. 61972087 and No. 62172089. Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No. BM2003201, Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No. 93K-9, and Marine Science and Technology Innovation Program under of Jiangsu Province under Grants No. JSZRHYKJ202308.

Data Availability Statement: Data are available from the authors upon reasonable request. The data is not publicly available because it contains business secrets. Public disclosure will damage the lawful rights and interests of third parties, so it should not be made public.

Acknowledgments: The authors wish to thank Linfeng Qian of the Jiangsu Marine Economic Monitoring and Evaluation Center.

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

References

- 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, 10–12 June 2008; pp. 1247–1250.
- Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. Dbpedia: A nucleus for a web of open data. In Proceedings of the International Semantic Web Conference, Busan, Republic of Korea, 11–15 November 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 722–735.
- 3. Abu-Salih, B. Domain-specific knowledge graphs: A survey. J. Netw. Comput. Appl. 2021, 185, 103076. [CrossRef]
- 4. Gao, Z.; Ding, P.; Xu, R. KG-Predict: A knowledge graph computational framework for drug repurposing. *J. Biomed. Inform.* 2022, 132, 104133. [CrossRef] [PubMed]
- Liu, L.; Li, X. Research and construction of marine Chinese medicine formulas knowledge graph. In Proceedings of the 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Houston, TX, USA, 9–12 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 3853–3855.
- Fu, P.; Yu, W.; Ren, Y.; Wang, Y. Construction and Application of Marine Accidents Knowledge Graph. In Proceedings of the ISCTT 2021: 6th International Conference on Information Science, Computer Technology and Transportation, Xishuangbanna, China, 26–28 November 2021; VDE: Frankfurt am Main, Germany, 2021; pp. 1–5.
- Liu, X.; Zhang, Y.; Zou, H.; Wang, F.; Cheng, X.; Wu, W.; Liu, X.; Li, Y. Multi-source knowledge graph reasoning for ocean oil spill detection from satellite SAR images. *Int. J. Appl. Earth Obs. Geoinf.* 2023, *116*, 103153. [CrossRef]
- Myklebust, E.B.; Jimenez-Ruiz, E.; Chen, J.; Wolf, R.; Tollefsen, K.E. Knowledge graph embedding for ecotoxicological effect prediction. In Proceedings of the Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, 26–30 October 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 490–506.
- Sakor, A.; Jozashoori, S.; Niazmand, E.; Rivas, A.; Bougiatiotis, K.; Aisopos, F.; Iglesias, E.; Rohde, P.D.; Padiya, T.; Krithara, A.; et al. Knowledge4COVID-19: A semantic-based approach for constructing a COVID-19 related knowledge graph from various sources and analyzing treatments' toxicities. *J. Web Semant.* 2023, 75, 100760. [CrossRef] [PubMed]
- 10. Dong, L.; Lapata, M. Language to Logical Form with Neural Attention. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 33–43.
- Shao, B.; Gong, Y.; Qi, W.; Cao, G.; Ji, J.; Lin, X. Graph-based transformer with cross-candidate verification for semantic parsing. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8807–8814.
- 12. Yih, S.W.; Chang, M.W.; He, X.; Gao, J. Semantic parsing via staged query graph generation: Question answering with knowledge base. In Proceedings of the the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing of the AFNLP, Beijing, China, 26–31 July 2015.
- 13. Lan, Y.; Jiang, J. Query Graph Generation for Answering Multi-hop Complex Questions from Knowledge Bases. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 969–974.
- Chen, Y.; Li, H.; Hua, Y.; Qi, G. Formal query building with query structure prediction for complex question answering over knowledge base. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021; pp. 3751–3758.
- Jia, Y.; Tan, C.; Chen, Y.; Zhu, M.; Chao, P.; Chen, W. Two-Stage Query Graph Selection for Knowledge Base Question Answering. In Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing, Guilin, China, 24–25 September 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 16–28.
- Sun, H.; Dhingra, B.; Zaheer, M.; Mazaitis, K.; Salakhutdinov, R.; Cohen, W. Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 4231–4242.
- Yan, Y.; Li, R.; Wang, S.; Zhang, H.; Daoguang, Z.; Zhang, F.; Wu, W.; Xu, W. Large-scale relation learning for question answering over knowledge bases with pre-trained language models. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 3653–3660.
- 18. Sun, H.; Bedrax-Weiss, T.; Cohen, W.W. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. *arXiv* **2019**, arXiv:1904.09537.
- Lu, J.; Zhang, Z.; Yang, X.; Feng, J. Efficient subgraph pruning & embedding for multi-relation QA over knowledge graph. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–8.
- 20. Zhang, J.; Zhang, X.; Yu, J.; Tang, J.; Tang, J.; Li, C.; Chen, H. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. *arXiv* 2022, arXiv:2202.13296.
- Jiang, J.; Zhou, K.; Zhao, X.; Wen, J.R. UniKGQA: Unified Retrieval and Reasoning for Solving Multi-hop Question Answering Over Knowledge Graph. In Proceedings of the Eleventh International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
- 22. Gruber, T.R. A translation approach to portable ontology specifications. *Knowl. Acquis.* 1993, *5*, 199–220. [CrossRef]
- 23. Swartout, B.; Patil, R.; Knight, K.; Russ, T. Toward distributed use of large-scale ontologies. In Proceedings of the Tenth Workshop on Knowledge Acquisition for Knowledge-Based Systems, Banff, AB, Canada, 9–14 November 1996; Volume 138, p. 25.

- 24. Musen, M.A. The protégé project: A look back and a look forward. AI Matters 2015, 1, 4–12. [CrossRef] [PubMed]
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. arXiv 2019, arXiv:1907.11692.
- Hu, X.; Jiang, Y.; Liu, A.; Huang, Z.; Xie, P.; Huang, F.; Wen, L.; Yu, P.S. Entda: Entity-to-text based data augmentation approach for named entity recognition tasks. arXiv 2022, arXiv:2210.10343.
- Wu, S.; He, Y. Enriching pre-trained language model with entity information for relation classification. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2361–2364.
- Miller, J.J. Graph database applications and concepts with Neo4j. In Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA, 23–24 March 2013; Volume 2324, pp. 141–147.
- Karahan, M.; Hakkani-Tur, D.; Riccardi, G.; Tur, G. Combining classifiers for spoken language understanding. In Proceedings of the 2003 IEEE Workshop on Automatic Speech Recognition and Understanding, St Thomas, VI, USA, 30 November–4 December 2003; IEEE: Piscataway, NJ, USA, 2003; pp. 589–594.
- Silva, J.; Coheur, L.; Mendes, A.C.; Wichert, A. From symbolic to sub-symbolic information in question classification. *Artif. Intell. Rev.* 2011, 35, 137–154. [CrossRef]
- 31. Kim, Y. Convolutional neural networks for sentence classification. arXiv 2014, arXiv:1408.5882.
- Aho, A.V.; Corasick, M.J. Efficient string matching: An aid to bibliographic search. *Commun. ACM* 1975, *18*, 333–340. [CrossRef]
 Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al.
- Language models are few-shot learners. Adv. Neural Inf. Process. Syst. 2020, 33, 1877–1901.
- Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7871–7880.
- Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 2020, 21, 5485–5551.
- 36. *HY/T123-2009*; Classification Standard for the Use of Sea Areas of the People's Republic of China. State Oceanic Administration: Beijing, China, 2009.
- Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002; pp. 311–318.
- Rouge, L.C. A package for automatic evaluation of summaries. In Proceedings of the Workshop on Text Summarization of ACL, Barcelona, Spain, 25–26 July 2004; Volume 5.
- Banerjee, S.; Lavie, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, MI, USA, 29 June 2005; pp. 65–72.
- Li, J.; Galley, M.; Brockett, C.; Gao, J.; Dolan, W.B. A Diversity-Promoting Objective Function for Neural Conversation Models. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 110–119.
- Li, X.; Hu, S.; Zou, L. Knowledge based natural answer generation via masked-graph transformer. World Wide Web 2022, 25, 1403–1423. [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.