applied sciences

MDPI

*Article*

# Enhancement of Question Answering System Accuracy via Transfer Learning and BERT

**Kai Duan [1], Shiyu Du [2,*], Yiming Zhang [2], Yanru Lin [1], Hongzhuo Wu [1] and Quan Zhang [1]**

[1] Department of Information Science and Engineering, Ningbo University, Ningbo 315211, China
[2] Engineering Laboratory of Advanced Energy Materials, Ningbo Institute of Materials Technology and Engineering, Chinese Academy of Sciences, Ningbo 315201, China
* Correspondence: dushiyu@nimte.ac.cn

**Abstract:** Entity linking and predicate matching are two core tasks in the Chinese Knowledge Base Question Answering (CKBQA). Compared with the English entity linking task, the Chinese entity linking is extremely complicated, making accurate Chinese entity linking difficult. Meanwhile, strengthening the correlation between entities and predicates is the key to the accuracy of the question answering system. Therefore, we put forward a Bidirectional Encoder Representation from Transformers and transfer learning Knowledge Base Question Answering (BAT-KBQA) framework, which is on the basis of feature-enhanced Bidirectional Encoder Representation from Transformers (BERT), and then perform a Named Entity Recognition (NER) task, which is appropriate for Chinese datasets using transfer learning and the Bidirectional Long Short-Term Memory-Conditional Random Field (BiLSTM-CRF) model. We utilize a BERT-CNN (Convolutional Neural Network) model for entity disambiguation of the problem and candidate entities; based on the set of entities and predicates, a BERT-Softmax model with answer entity predicate features is introduced for predicate matching. The answer ultimately chooses to integrate entities and predicates scores to determine the definitive answer. The experimental results indicate that the model, which is developed by us, considerably enhances the overall performance of the Knowledge Base Question Answering (KBQA) and it has the potential to be generalizable. The model also has better performance on the dataset supplied by the NLPCC-ICCPOL2016 KBQA task with a mean F1 score of 87.74% compared to BB-KBQA.

**Keywords:** Chinese knowledge base; question answering system; BERT; transfer learning; CNN

check for updates

**Citation:** Duan, K.; Du, S.; Zhang, Y.; Lin, Y.; Wu, H.; Zhang, Q. Enhancement of Question Answering System Accuracy via Transfer Learning and BERT . *Appl. Sci.* **2022**, *12*, 11522. https://doi.org/10.3390/app122211522

Academic Editor: Francesco Parisi

Received: 30 August 2022
Accepted: 4 November 2022
Published: 13 November 2022

## 1. Introduction

With the penetration of the internet into various fields and industries and the growing scale of users, the scale of information carried by the internet is increasing, and the content is more heterogeneous and diverse, presenting challenges for people to access information rapidly and effectively. The arrival of the era of big data has changed this status quo, which is characterized by scale, diversity, rapidity, and authenticity. Big data technology provides the possibility of large-scale knowledge acquisition. In such a background, knowledge engineering (the technology of obtaining knowledge and information with high efficiency and large capacity by using modern scientific and technological means) has ushered in new development opportunities. Especially after Google announced the knowledge graph project in 2012 to enhance its search engine performance, knowledge engineering has entered a brand-new stage led by knowledge graph technology, that is, the knowledge engineering stage in the big data era. Knowledge graphs based on big data have rapidly gained significant attention from academia, industry, and government departments and the knowledge graph slowly extends from search engines to the question and answering system.

Large-scale knowledge bases (KB) have emerged over recent years, such as DBpedia [1], Freebase [2], Yago [3], and the Chinese knowledge base published by the NLPCC-ICCPOL-2016 Knowledge Base Question Answering (KBQA) evaluation task [4], KBQA tasks are gradually becoming a hot spot in the area of natural language processing. Knowledge graphs are a special type of semantic network, which represents entities in the objective world and their relationships in graphic forms. Generally, virtual knowledge exists in the form of triples, such as <entity, predicate, target value>, which can help organize, manage, and comprehend the vast amount of information available online. Most of the knowledge on the web is unstructured or semi-structured, organized in a way that is suitable for people to read and understand but not computer-friendly [5]. Knowledge graphs are helpful in constructing heterogeneous knowledge in the domain and establishing inter-knowledge associations. Google, Facebook, and Baidu have developed some knowledge graphs, and demonstrated their value in many ways. KBQA [6] utilizes knowledge graphs as one of the sources of knowledge to understand the questions using natural language input from users and identify the entities and predicates to find the corresponding target values as answers.

Existing mainstream approaches to Chinese Knowledge Base Question Answering (CKBQA) usually split the question and answering task into two subtasks: entity linking and predicate matching. Regarding the entity linking module, Wang et al. [7] used convolutional neural networks(CNN) and Gate Recurrent Unit(GRU) models to obtain semantic representations of questions. Xie et al. [8] used CNN to develop Named Entity Recognition (NER) and Bidirectional Long Short-Term Memory (BiLSTM) and CNN to implement predicate mapping. Yang et al. [9] reported their progress in NER by extracting various features, using Gradient Boosting Decision Tree (GBDT) model exploration, and using Naive Bayes Support Vector Machine (NBSVM) and CNN support to design predicate mapping. Lai et al. [10] generated candidate entities through an alias dictionary, constructed artificial rules for entity disambiguation, and calculated cosine similarity based on word vectors to score predicates. With the appearance of the pre-trained language model Bidirectional Encoder Representation from Transformers(BERT), Liu et al. [11] finely tuned the model on the basis of the BERT pre-training task for different subtasks in the CKBQA process and obtained good results on the open-domain Chinese knowledge-based question-answering task. However, the aforementioned methods suffer from two disadvantages: firstly, while previous language knowledge can be incorporated into hand-crafted templates, template design can consume a large amount of computational time. At the same time, manual templates tend to have large granularity and are prone to exceptions, which limits the model of ability for generalization. Secondly, the performances of conventional methods are low on the Chinese dataset. Compared with the NER task for English, NER for Chinese is more difficult because Chinese sentences cannot be spatially segmented as naturally as English, and the presence of a large number of indistinguishable entities, as well as the presentation differences between Chinese questions and basic knowledge, which makes it difficult for general models to learn text features adequately. For Chinese NER, the common practice is to use the Chinese Word Segmentation (CWS) tool for segmentation before applying word order tokens. However, the noise present in the CWS tool itself can significantly affect the performance of NER models.

For the purpose of this work, we focus on the CKBQA task via the designed CWS and BERT models in order to address the difficulties such as the insufficient association between entities and predicates. To achieve the final answer selection, we link the score of entities and predicates, determining the query path and retrieve a definitive answer. The key contributions of this paper are as follows:

1. An NER transfer learning model is presented. By fusing CWS in the NER task, the accuracy of candidate entities is improved without introducing too much noise. We also use a pre-trained BERT model combined with CNN and Softmax models for entity disambiguation and predicate mapping in which richer contextual information can be learned. The model makes full use of the problem encoding and candidate information features extracted by the BERT model and has strong generalization for application in multi-domain knowledge bases.

2. The results of the experiments suggested that we have developed a satisfactory question answering system for the Chinese dataset. The current method achieves better

on the NLPCC-ICCPOL 2016 KBQA dataset. It can generate more accurate and relevant answers due to the transfer learning and BERT's powerful feature extraction capability.

The remainder of this work is organized as follows: Section 2 describes the background of the question-answering system, KBQA, and NER. Section 3 introduces the related technologies. The model, results, and discussion proposed by us will be presented in Sections 4–6. Section 7 provides a conclusion.

## 2. Background

### 2.1. Question Answering System

As a critical field in artificial intelligence [12], intelligent question and answer is an essential branch of natural language processing, usually in a question-and-answer human-computer interaction to locate the user's desired knowledge and provide personalized information services. Unlike search engines, it allows computers to answer users' questions automatically in a precise natural language format. The history of intelligent question and answering dates back to 1950, when Alan Turing, the father of computer science, came up with the Turing Test to determine if it possible for a computer might think accurately and correctly, thus opening the chapter on natural language human-computer interaction. Around the 1960s, the first question-and-answering systems were introduced, and Green et al. designed a Baseball program that could answer questions about baseball games in plain English. In 1966, Weizen-Baum designed and implemented ELIZA Chatbot [13], which can process simple problem statements. In 1971, another early chatbot [14] was developed by Kenneth Colby, a psychiatrist at Stanford University, and named "Parry". These question-answering systems based on rule matching could not be widely applied due to lacking data resources at that time.

As deep learning and natural language processing technology rapidly advances, the question-answering system gradually transitions from early rule matching to retrieval matching [15]. The core idea is to extract the core words in natural language questions, search for the relevant answers in documents or web pages according to the core words and return the corresponding answers using the correlation sorting algorithm. Ma et al. [16] proposed the pseudo-correlation feedback algorithm based on the method of automatic document retrieval, which used the context information in the document to retrieve the most similar answers. The retrieval matching approach achieved good results when it was first proposed. However, as the number of data increased and the diversity of user questions emerged, the quality of answers extracted from documents or web pages by this approach varied, profoundly affecting system response time and the accuracy. Until the concept of knowledge graph was proposed, the KBQA is significantly improved in quality and has realized the form of extracting questions and answers from documents. At present, KBQA has received more and more attention from researchers, and it has become a topic of intense interest in the natural language field [17].

### 2.2. Knowledge Graphs and Knowledge Bases

In 2012, Google originally put forward the concept of the knowledge graph and applied it to enhance the capabilities of conventional search engines. In the real world, the knowledge graph presents structural knowledge as triples (entity-relations-entity or concept-attribute-value), forming a multilateral relationship network. Its essence is a semantic network that can reveal the entities' relationships. According to different knowledge coverage fields, the knowledge spectrum can be divided into broad domain knowledge spectrum (e.g., Wikidata [18], DBpedia, CN-DBpedia [19], Freebase, etc.) and specific domain knowledge spectrum (e.g., Ali Commodity Atlas [20], Meituan Gourmet Atlas [21], AMiner [22]). Traditional knowledge graph construction methods include entity recognition [23], entity disambiguation [24], relationship extraction [25] and knowledge storage, etc.

With the emergence and rapid development of deep learning, the knowledge graph has gradually changed from "symbol" connection to "vector" representation. The model of TransE is suggested by Boards et al, in which the entities and relationships are embedded into the semantic space of a low-dimensional vector, and the relation vector is considered to be a translation of the head entity vector into the tail entity vector. TransR/CTransR proposed by Lin et al. [26] sets a unique relation matrix space M for each relation and incorporates entities and relations into vector semantic space via M matrix for translation calculation. The knowledge graph construction method based on knowledge representation learning fundamentally solves the long tail effect brought by the traditional method and greatly improves the usability of the knowledge graph.

### 2.3. KBQA

A crucial question related to KBQA is how to translate natural language problems into formal language that can be understood by the computer and obtain the answer to the problem through query and reasoning within the constructed KB. Therefore, KBQA mainly includes Semantic Parsing-based (SP-based) methods and Information Retrieval-based (IR-based) Methods.

The SP-based methods analyzes components of natural language questions, converts the query into logical expressions, and then converts the query into a knowledge graph query to get the answers. Hao et al. [27] parsed natural utterances into subgraphs of knowledge graph to achieve complex question answering, and the model was found practical; Meng et al. [28] designed a semantic query expansion method to solve the problem of difficulty in obtaining ideal answers from data sources, which expanded query terms in question triples from three semantic perspectives and achieved multi-semantic expansion of the question triples. This method can more clearly convert natural language problem statements into logical expressions. However, the method requires many manually defined logical expression rules, which perform well in specific domains but are not generalizable when dealing with large-scale knowledge graphs. In other words, this method is suitable in specific domains but cannot transform undefined rules when dealing with large-scale knowledge graphs.

The IR-based methods extract critical information from the question and use that to qualify the knowledge of the knowledge base and then retrieve the answer. Qiu et al. [29] suggested a Stepwise Reasoning Network (SRN) model on the basis of intensive learning. The SRN model formalizes the problems as sequential strategy ones and an attention mechanism is adopted to obtain exclusive information within the problem, which significantly enhances the effectiveness of question and answering based on information retrieval methods; Xu et al. [30] argued that KG lacks context to provide a more precise conceptual understanding, although it contains rich structural information. For this reason, they designed a model that uses external entity descriptions for knowledge understanding to assist in completing knowledge question and answering. This approach achieves optimal results on the Common-sense QA dataset and obtains the best results in the non-generative model of OpenBookQA.

### 2.4. NER

NER is the identification of named entities with particular meanings in text and classifying them into predefined entity types, such as person name, place name, institution name, time, currency, etc. Named entities usually contain rich semantics and are closely related to the critical information in the data. The NER task can resolve issues on information explosion in text data online, and obtain critical information effectively. Moreover, NER is commonly used in different areas, such as relationship extraction, machine translation, and knowledge graph construction.

Chinese-oriented NER started later. One has immediately noticed that Chinese is quite different from West Germanic languages such as English due to its language characteristics. Hence, NER in the Chinese field mainly has the following three particularities. (1) Chinese NER should solve the difficulty in boundary ambiguities. It is because Chinese unit vocabulary boundaries lack clear separators such as spaces in English text and have no apparent morphological transformation features. (2) Chinese NER must be combined with CWS and grammatical analysis, only which can correctly classify named entities and improve the performance of NER. (3) Context may have significant influence on NER tasks. In Chinese text, there may exist complex sentences, flexible expressions, and many omissions. The exact words in different fields have different meanings, and there may be multiple expressions for the same semantics, which can only be clarified by the context. In addition, the internet has developed so rapidly, and the greater personalization and randomization of text descriptions in online texts, makes identifying entities more difficult.

Nowadays, mainstream NER methods have three categories, including rule- and dictionary-based methods, statistical machine learning-based methods, and deep learning-based methods. This paper mainly studies the methods based on deep learning. The typical representative of Transformer-based methods is the pre-trained model BERT model. Souza1 et al. [31] put forward a BERT-CRF(Conditional Random Field) model for the NER task, combining BERT's transfer capability with the structured CRF prediction. Li et al. [32] addressed the lack of large-scale labelled clinical data by pre-training the BERT model on unlabeled Chinese clinical electronic medical record text, thereby leveraging unlabeled domain-specific knowledge. Wu et al. [33], who follows the work by Li et al., proposed a model based on Roberta and character root features, using Roberta to learn medical features, and using Bi-LSTM to extract character features. Yao et al. [34] designed a model using Albert-AttBiLSTM-CRF and migration learning for fine-grained entity recognition text. In this model, A more lightweight pre-training model ALBERT was adopted to embed words in the original data Bi-LSTM was employed to extract the features of words with embedding and obtain contextual information, and a decoding layer using CRF was utilized for label decoding.

## 3. Related Technologies

### *3.1. CNN*

CNN [35], a deep neural network, contains convolutional operations. Meanwhile, CNN is a representative algorithm of deep learning. It uses a convolution kernel to capture local features and has the ability of representational learning. Depending on the dimension of the input matrix, CNN can be divided into 1D-CNN and 2D-CNN, etc. Processing text usually uses 1D-CNN. The length of convolution kernel is always equal to the feature dimension of the text vector representation, whereas the width of convolution kernel determines the window size of the contextual words or characters. In practical applications, multiple convolution kernels of different widths are typically utilized in order to obtain various receptive fields and augment the feature information extracted by the model. The convolution kernels are shifted in a certain step size according to the text sequence length to obtain the feature vector representation of the text sequence. After the convolution layer, the top pooling layer or the average pooling layer is usually connected to choose features and decrease the number of parameters. For features after pooling, classification and other operations are conducted through the entire connection layer, as shown in Figure 1.

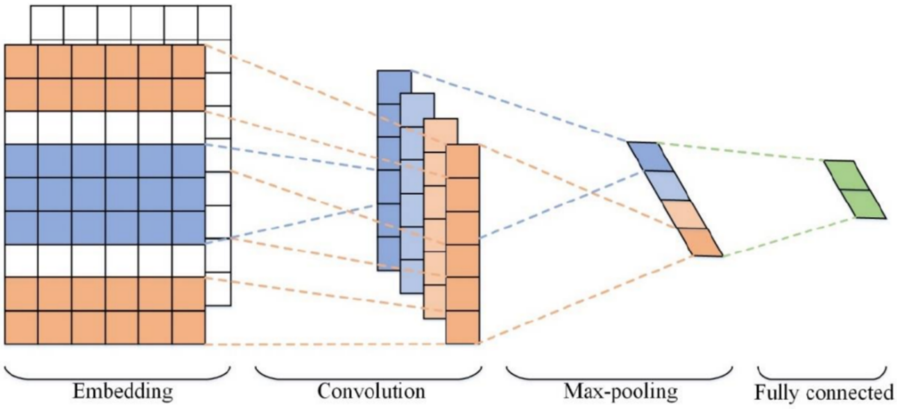

**Figure 1.** Illustration of CNN (Redrawn from [35]).

CNN has strong feature capturing capability, nevertheless, when processing text data, the convolution kernel can only pay attention to the local information in the window and lacks the learning of long-distance information, As a result, overall understanding to the text is lacking. Thus, CNN usually handles text-classification tasks rather than sequence-annotation tasks. When combined with other neural network structures, CNN can be used as a feature extractor to enhance the ability of feature representation. For example, when processing Chinese text, CNN can be used to extract stroke features or glyphs of Chinese characters.

### 3.2. RNN

Recurrent Neural Network (RNN) consists of connected cyclic units and featured by recursion in sequences' evolutionary direction. Compared with CNN, it is better suited to modeling sequential data, such as text. At each time step (for each token position in the text), each loop unit uses shared parameters, and both the input of the current time step and the output of the previous time step can affect the output. Figure 2 shows the structure of RNN. Where $W$, $U$, and $V$ are the training process's weight matrices of learning parameters.

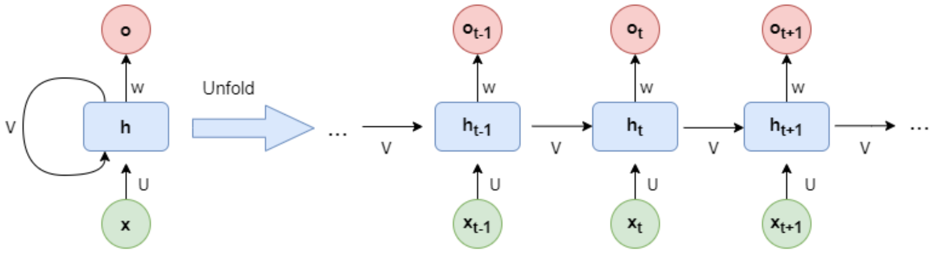

**Figure 2.** Illustration of RNN(Redrawn from [36]).

Although the model structure of RNN is good at dealing with sequence problems, it is possible that gradient disappearance or gradient explosion occur in the training process for long sequences. To effectively handle long-distance dependence, many variants of RNN have emerged, among which the most distinctive representative is Long Short Term Memory Network (LSTM) [36]. The structure diagram of its cyclic unit is shown in Figure 3. Its improvement is that the input gate, output gate, and forgetting gate are added to each cycle unit. The sigmoid function limits the output of the gate between 0 and 1. Such a gating mechanism controls how much the output from the last moment should be forgotten and how much the input from the current moment should be retained and then calculates the output of the current moment. For a given input sequence $[x_1, x_2, \ldots, x_t, \ldots x_n]$, LSTM calculation formula for moment $t$ is shown in the following formula:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{2}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$

$$\bar{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{4}$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \bar{C}_t) \tag{5}$$

$$h_t = O_t \otimes \tanh(C_t) \tag{6}$$

$i_t, f_t, o_t$ are respectively LSTM network input to the door, forgetting door and output, $W$ and $U$ are the weighting matrix, $b$ is for bias, $C_t$ is cell state, $h_t$ is the network output. In practice, LSTM with forward and reverse directions is usually used to process text, and implicit layer vectors are spliced to obtain text representation with context information.

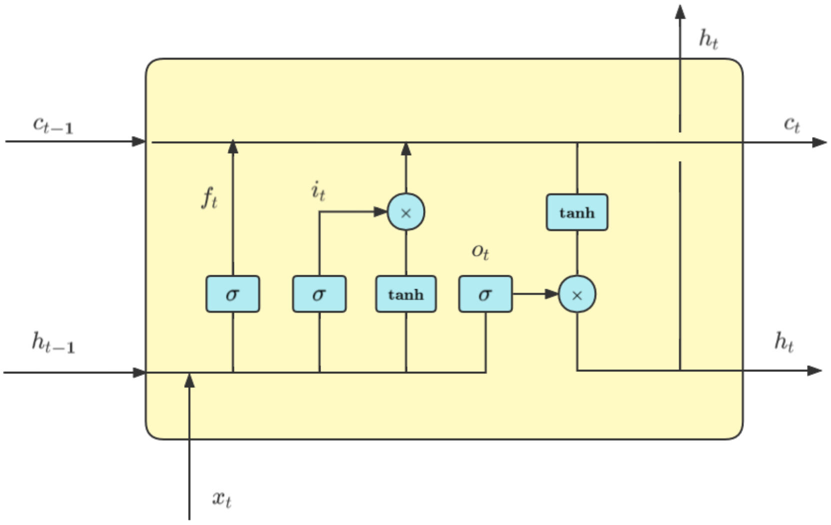

**Figure 3.** Illustration of a LSTM cell (Redrawn from [36]).

### 3.3. BERT

Using the pre-trained word vector for representing the words is better than using the randomly initialized vector, but the vector representation of the pre-trained word vector is the same in various contexts, which is a context-independent word vector. For example, “Who founded Apple?” and “How much does an apple cost?” The “apple” in the two sentences represents different meanings but uses the same vector, so the pre-trained word vector cannot solve the case of polysemy. Therefore, pre-trained language models such as ELMo (Embeddings from Language Models) [37], GPT (Generative pre-training) [38], and BERT [39] , build text sequences to obtain context-relevant Representations. The structures of these three models are shown in Figure 4. Elmo uses double-layer LSTM connected by residuals and encodes text sequences with forward and reverse directions. Its pre-training objective function is given as follows:

$$\sum_{n=1}^{N} \log P(x_n \mid x_1, \ldots, x_{n-1}) + \log P(x_n \mid x_{n+1}, \ldots, x_N) \tag{7}$$

The objective is to maximize the sum of probabilities that the context predicts the current word. After pre-training, word vectors encoded by ELMo can be input into the downstream model and applied to specific tasks. It can be seen from the objective function, however, that ELMo is a one-way language model essentially. In 2018, Google proposed BERT, which can better model bidirectional language models.

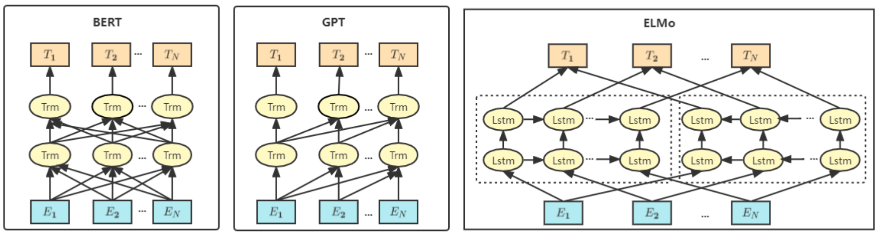

**Figure 4.** Illustration of pre-trained language models (Redrawn from [37–39]).

As shown in BERT’s model structure diagram, BERT mainly comprises the input, coding, and mask layers. Among them, the input layer splits the input text sequence into three vector representations: token vector, position vector, and clause vector, which are added together and input into the coding layer. The coding layer comprises encoders of a bi-directional connected Transformer [40]. The basic BERT model has 12 layers, which is the core structure of BERT. The transformer is a model structure suggested by Google in 2017. Figure 5 presents the structure of its encoder part, and it consists of a multi-attentional network and a feedforward network. The structure of task layer depends primarily on the task target. The SoftMax layer handles classification tasks, and the CRF layer follows sequence annotation. Since the input and coding layers are task-independent, data fine-tuning BERT is often used to achieve good results in downstream tasks.

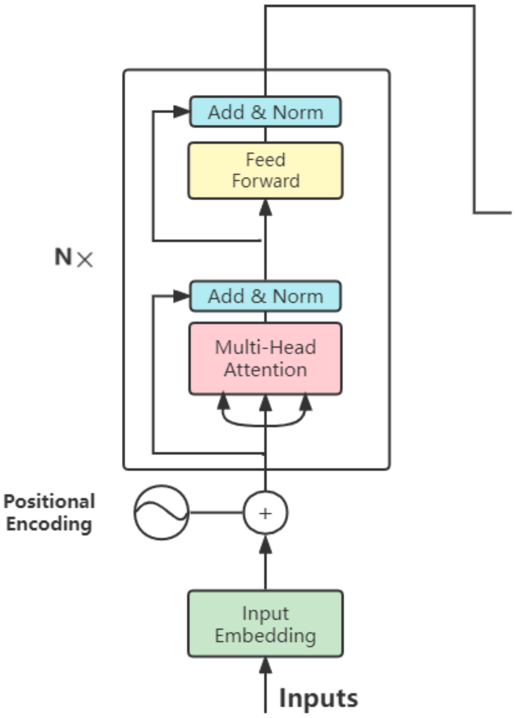

**Figure 5.** The structure of Transformer encoder (Redrawn from [40]).

BERT has two pre-training tasks: Mask Language Model (MLM) and Next Sentence Continuity Prediction (NSP). MLM replaces certain words in the text with [MASK] tags with certain rules and then predicts these words in the output layer. NSP is to predict whether two sentences are connected. It is often used in sentence pair similarity tasks. BERT has significantly improved 11 NLP tasks, which have become the focus of academic attention. After that, some BERT improvements have emerged continuously, such as XLNet [41], ALBERT [42], ELECTRA [43], ERNIE [44], etc. The work in this article is also on the basis of the BERT pre-training language model and performed well on the question and answering task.

## 4. Method

The purpose of this section is to demonstrate the overall structure of BAT-KBQA framework and the details of component modules. We begin by explaining general design of our framework and then describe the details of the essential components.

### *4.1. Overall Architecture*

Figure 6 shows the overall flow of the KBQA system. The three core modules are entity linking (consisting of named entity identification and entity disambiguation), predicate mapping, and answer selection in the system. The objective of the entity linking step is to discover the name of entity posed in the query, while the predicate mapping step aims to find the relevant attributes asked in the question, and the answer selection is a combination of these two steps to reach an accurate answer. We use the example of "Who is the author of Journey to the West" to describe the flow of our system. Firstly, the NER model identifies the key entity “Journey to the West” from the question (in Chinese); combined with the knowledge base, we can generate a collection of candidate entities related to “Journey to the West”, and the entity disambiguation model scores the candidate entities. The highest scoring candidate entity “Journey to the West (novel)” is adopted as the question of subject entity; combining the subject entity and the knowledge base, the predicates of the subject entity are used as a candidate predicate set. The named entity is then replaced with the question sentence of the uniform identifier “entity” and the candidate predicate is fed into the predicate matching model, obtaining the predicate “author” with the highest score; finally, the answer selection module combines the entity and the predicate and queries the knowledge base to arrive at the final answer. The modules will be described in subsequent sections with more details.

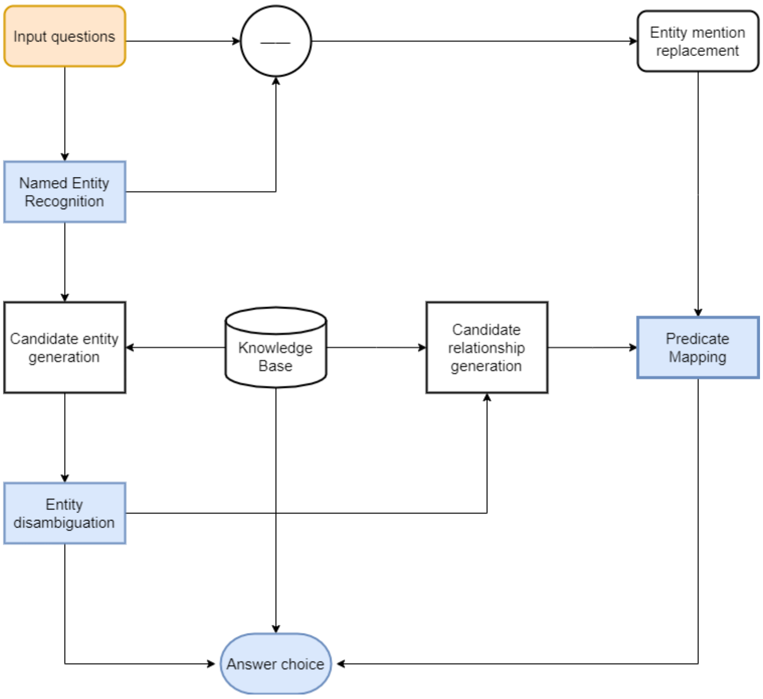

**Figure 6.** The overall process of the BAT-KBQA framework. (The modules of the three cores have been marked with bright colors.)

## *4.2. Model For Entity Linking*

Entity linking refers to the task, which links the expression in the text to the corresponding entity in the knowledge base to conduct entity disambiguation and assist humans and computers to understand the text’s particular meaning. For example, in the text “Do you know who is the author of the book ‘Journey to the West?”’, there is “Journey to the West (TV drama)”, “Journey to the West (novel)”, and “Journey to the West (game)” to express the corresponding entity of “Journey to the West” in the knowledge database. In our example, it links the expression “Journey to the West” to “Journey to the West (novel)” in the knowledge base to eliminate the ambiguity caused by other meanings. So, entity linking is the essential part of knowledge graph construction. The entity linkage model is divided into NER and entity disambiguation.

### 4.2.1. Named Entity Recognition

The goal of NER is to recognize named entities in text and assign them to the corresponding entity types. It is important since the semantic expression of Chinese corpus is sparse, many similar entities are difficult to be distinguished, and differences exist in the presentation of Chinese interrogative sentences and fundamental knowledge. Hence, it is difficult for general models to learn text features thoroughly, which makes it challenging to enhance the accuracy of entity linking. Here we suggest an adversarial transfer learning

NER model. Our model incorporates adversarial transfer learning and the CWS task [45] to solve these problems and introduces two major innovations, that is, applying CWS tasks brings shared information without introducing new noise and adding a self-attentive mechanism in the middle of the BiLSTM and CRF layers. The model tentatively learns word boundary information shared by the task from the CWS task, then filters particular information of CWS and unambiguously captures long range dependencies between two arbitrary characters in one sentence finally. Figure 7 presents the architecture of this model. The model is composed of five elements: embedding layer, shared-private feature extractor, self-attention, task-specific CRF, and task discriminator. Each portion of the model would be given a detailed elaboration in subsequent sections.

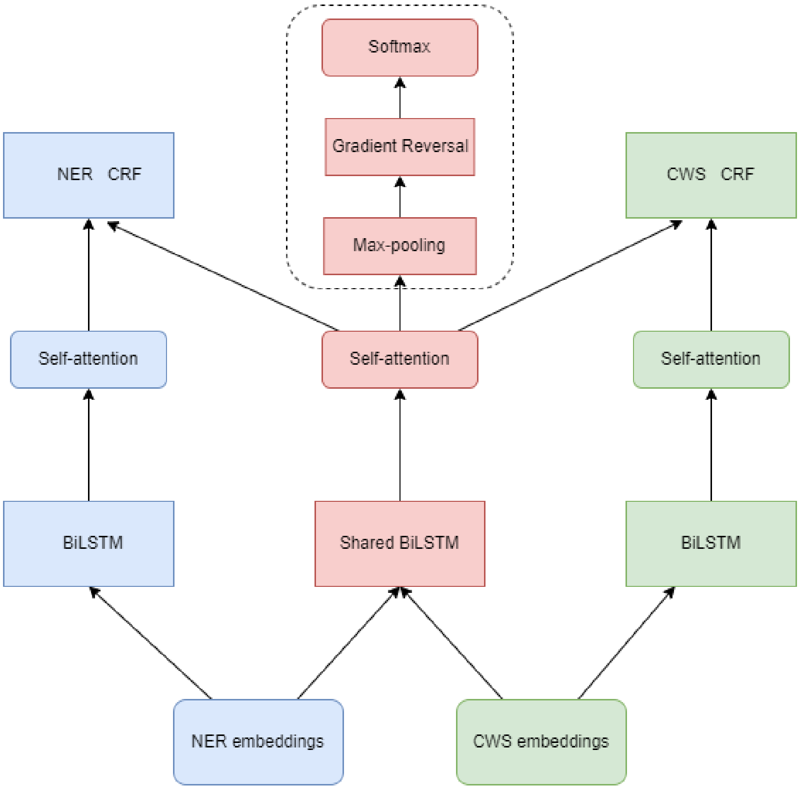

**Figure 7.** Named Entity Recognition model. (Consists of BiLSTM, Self-attention, CRF, feature recognizer).

First, the NER model is adopted to distinguish the topic entity in the question as the starting point for semantic parsing of the question, e.g., “Journey to the West?” in “Who is the author of Journey to the West?” is the reference form of the subject entity of the question. NER can be regarded as a sequence annotation task, in which the common tagging system “BIO” is used. Here “B” indicates the starting position of the entity mentioned, “I” indicates the middle or end positions of entity mentions, and “O” denotes that the character is not an entity mention. The position corresponding to the “Journey to the West” in the question is labelled as “B I I I”, and the other non-entity mentions are labelled as “O”. In this paper, the NER and CWS task datasets are labelled separately and embedded into the model for training.

The embedding layer and other neural network models have similarities. Pre-trained embedding dictionaries are loaded to map discrete NER and CWS characters into distributed embedding vectors.

Long-term memory [46] is a RNN’s variant and it can resolve the gradient disappearance and explosion problems by introducing gate mechanisms and memory units. The unidirectional LSTM utilizes only the past information but overlooks future information, so we use BiLSTM for feature extraction to fuse the information from two sides of the sequence. In our model, in addition to the two-end BiLSTM for extracting private features, we add a shared BiLSTM to extract the boundary information shared by the NER and CWS tasks. That is, for the dataset of task $m$, the hidden states of the shared and private BiLSTM layers can be calculated as shown in the following.

$$\mathbf{s}_i^m = \text{BiLSTM}\left(\mathbf{x}_i^m, \mathbf{s}_{i-1}^m; \theta_s\right) \tag{8}$$

$$\mathbf{h}_i^m = \text{BiLSTM}\left(\mathbf{x}_i^m, \mathbf{h}_{i-1}^m; \theta_m\right) \tag{9}$$

where $\theta_s$ is the shared BiLSTM parameter and $\theta_m$ is the private BiLSTM parameters.

In the third step, we draw self-attention as applied to machine translation and semantic role labelling. After the feature extractor, we add a multi-headed attention mechanism [40] in order to learn the dependencies between two arbitrary words in one sentence and seize the internal structural information of the sentence. The formula is as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{QK}^T}{\sqrt{\mathbf{d}}}\right)\mathbf{V} \tag{10}$$

where $Q \in R^{N\times 2dh}$, $K \in R^{N\times 2dh}$and $V \in R^{N\times 2dh}$ are query matrix, keys matrix and value matrix, respectively. $d$ is the dimension of BiLSTM hidden layers. More information about self-attention mechanisms are detailed in [40].

The next layer is the CRF [47]. The CRF layer can add some constraints to the final predicted labels to ensure the reasonableness of the predicted labels. Because the labels are different, we create a unique CRF layer for each task separately. For each sentence of task m, the final representation is to concatenate the BiLSTM layer with the self-attention representation.The final decoding process uses the Viterbi algorithm.

As shown in the Figure 7, the middle layer enters our task recognizer. We propose a task recognizer to determine from which task the sentences come. The following formula can signify the task recognizer:

$$\mathbf{s}'^m = \text{ Maxpooling } (\mathbf{S}'m) \tag{11}$$

$$D\left(\mathbf{s}'^m; \theta_d\right) = \text{softmax}\left(\mathbf{W}_d\mathbf{s}' + \mathbf{b}_d\right) \tag{12}$$

where $\theta_d$ denotes the parameters of task recognizer . $W_d \in R^{K\times 2dh}$ and $b_d \in R^K$ are trainable parameters. $m$ denotes the number of tasks. The middle layer is the shared features of CWS and NER tasks, to prevent task-specific features from being incorporated into the shared features. Inspired by adversarial networks, we introduce adversarial loss to train the shared BiLSTM, resulting in a task recognizer that cannot reliably identify the task from which the sentence comes. The following equation calculates the adversarial loss:

$$L_{Adv} = \min_{\theta_s}\left(\max_{\theta_d} \sum_{m=1}^{m} \sum_{i=1}^{T_m} \log D\left(E_s\left(\mathbf{x}_m^{(i)}\right)\right)\right) \tag{13}$$

where $\theta_s$ refers to the shared BiLSTM's trainable parameters. $E_s$ is the shared feature extractor. $T_m$ is the number of training examples of task $m$ and $x_m^{(i)}$ is the $i$-th example. The shared BiLSTM develops a representation to deceive the task recognizer as part of a minimax optimization, while the task recognizer tries its best to accurately recognize the type of work. To tackle the minimal-extreme optimization problem, we add a gradient inversion layer below the Softmax layer. We reduce the task recognizer's error during the training stage, and invert the gradient through the gradient inversion layer, which favors the shared feature extractor for studying the word boundary information shared by the task. The shared feature extractor and the task recognizer arrive at a point when the recognizer is unable to discriminate tasks based on the representations acquired from the shared feature extractor after training phrases.

In training, the following is the final loss function of our model:

$$L = L_{NER} \cdot I(\mathbf{x}) + L_{CWS} \cdot (1 - I(\mathbf{x})) + \lambda L_{Adv} \tag{14}$$

where $\lambda$ denotes a hyper-parameter. Equation (21) can be used to calculate $L_{NER}$ and $L_{CWS}$. $I(x)$ is a switching function determining which task the input is from. The formula is as follows:

$$I(\mathbf{x}) = \begin{cases} 1, & \text{if} \quad \mathbf{x} \in \mathcal{D}_{NER} \\ 0, & \text{if} \quad \mathbf{x} \in \mathcal{D}_{CWS} \end{cases} \tag{15}$$

where $D_{NER}$ is Chinese $NER$ training corpora and $D_{CWS}$ is $CWS$ training corpora. In each iteration of the training phase, we sequentially select a task from $\{NER, CWS\}$ and obtain different training samples to update the parameters. We optimize the final loss function by utilizing the Adam algorithm. Since the convergence rates of the Chinese $NER$ task and the $CWS$ task could be distinct, we duplicate the above iterations until early stopping based on the performance of the Chinese $NER$ task.

#### 4.2.2. Model for Entity Disambiguation

Next, since the entities mentioned in the natural language question may correspond to multiple entities saved in the knowledge base. After obtaining the entity identification of the subject entity in each question, it is essential to generate a set of candidate entities related to the entity mention from the knowledge base and disambiguate these candidate entity sets and to select the correct one. Accurately matching the subject entities asked in the question can also reduce the candidate set size for the next step of predicate matching and enhance the efficiency of the question answering system. We suggest a BERT-CNN model that introduces the predicate features of the entities once they are chained in order to boost the performance of the entity disambiguation task.

In the entity disambiguation part, to obtain the set of candidate entities close to the subject entities in the question, we firstly generate the set of candidate entities by mapping the entity mentions identified in the previous step to the mention2id library provided by the NLPCC- ICCPOL-2016KBQA evaluation. For entity mentions that could not be mapped, we relied on the knowledge base to retrieve entities with similar character as the candidate entity set. Then, we input the interrogative sentences and the set of candidate entities into the BERT-CNN [35] model (as shown in Figure 8). This task can be viewed as a binary task, with the output label 1 if the candidate entity is a subject entity in a labeled triplet and 0 otherwise. The input data are in the form of [CLS], problem character sequences, [SEP], candidate entities for concatenation with predicate features, [SEP]. Among them, the predicate feature is the chained relationship from candidate entities in the knowledge graph to the connected predicates, as shown in Formula (16), where $q$ represents the problem, $e$ represents the candidate entity, and $p_i$ represents the chained relationship starting from $e$.

$$x = [CLS], q, [SEP], e, p_1, \ldots, p_n, [SEP] \tag{16}$$

After BERT network coding, the hidden layer vectors of the last four encoder outputs are obtained, and the hidden layer outputs $H$ after addition. Formula (17) illustrates the features of the convolution layer $C$ :

$$C = \sigma(H \otimes W + b) \tag{17}$$

where $\sigma$ is the sigmoid function, $\bigotimes$ is the convolution operation,$W$ is the weight in the convolution kernel, and $b$ is the bias. $H$ extract features through three convolution layers corresponding to stepsize 1, 3 and 5, respectively . Then the features feed into the highest pooling layer and the Softmax layer carries out classification after concatenation the three vectors, and the output label is 0 or 1. Formula (18) explains that the loss function is the cross-entropy loss function, which is minimized during training. In the prediction, the probability that the candidate entity is predicted as tag 1 is taken as the candidate entity's score.

$$L = -[y \cdot \ln(x) + (1 - y) \cdot \ln(1 - x)] \tag{18}$$

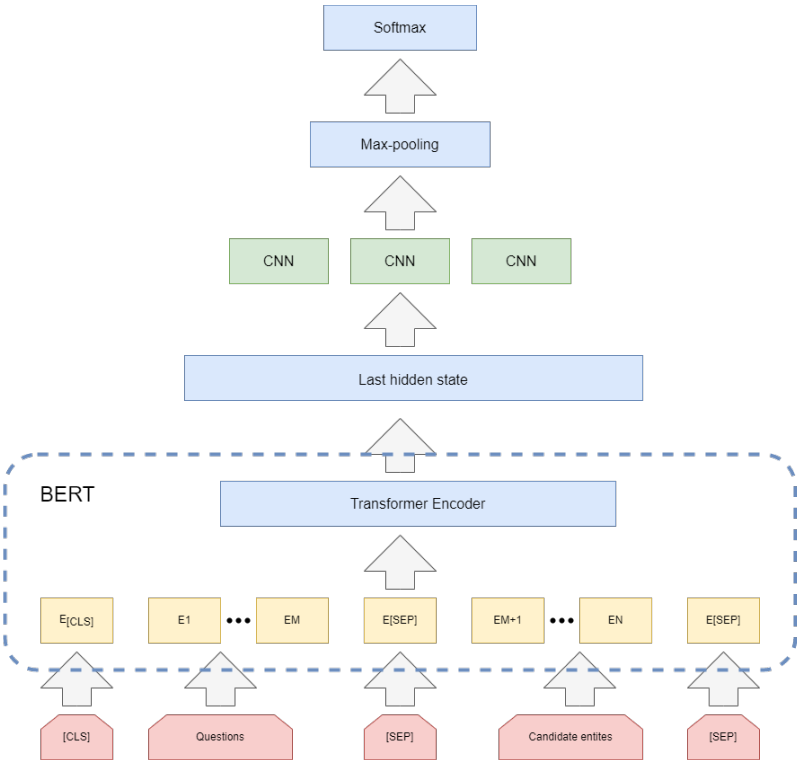

**Figure 8.** Entity Disambiguation model.

### *4.3. Model for Predicate Mapping*

Due to the variety of natural language questions, different expressions may correspond to the same question intent, and the same subject entity may also generate different predicates, which poses a major challenge to the domain KBQA task. Such as matching between the question "Who wrote the Journey to the West" and "Author ". The predicate matching model is suitable for matching the predicates in the question with the predicates in the knowledge base, understand the intent of question, and select the predicate that best matches the question. First, the entity disambiguation results can reduce the size of candidate predicates set, so we start from the sample of candidate entities acquired in the entity disambiguation task and retrieve the set of predicates of that entity in the knowledge base as candidate predicates set. Next, we notice that in answering the question "Who wrote the Journey to the West?", the information of the candidate predicates can be enriched by adding the information of the first-degree chained predicates of the candidate answer entities retrieved by the candidate predicates. For example, "The candidate entities of Journey to the West include release date, director, author, etc." are related to the candidate predicate "author".

Therefore, this paper proposes a BERT-Softmax model [11] (as shown in Figure 9) for entity-predicate matching. We treat BERT-Softmax as a binary sequence classification problem, where the output label is 1 for candidate relation samples that accurately reflect the intent of question and 0 for those inaccurately reflecting the aim of question. The input data are composed of two portions: the question and predicate. The input data of the question part consist of [CLS], a sequence of question characters with the entity character replaced by the entity character, [SEP], the candidate relation, and [SEP], which is encoded by the BERT network to acquire the hidden vector of the last four encoder layers, stitched together and input to the Softmax layer for classification. The output label is 0 or 1. The loss function is also the cross-entropy loss function, which was minimized during training. In the prediction, the probability of predicting a candidate relationship as label 1 is used as the score of candidate relationship.

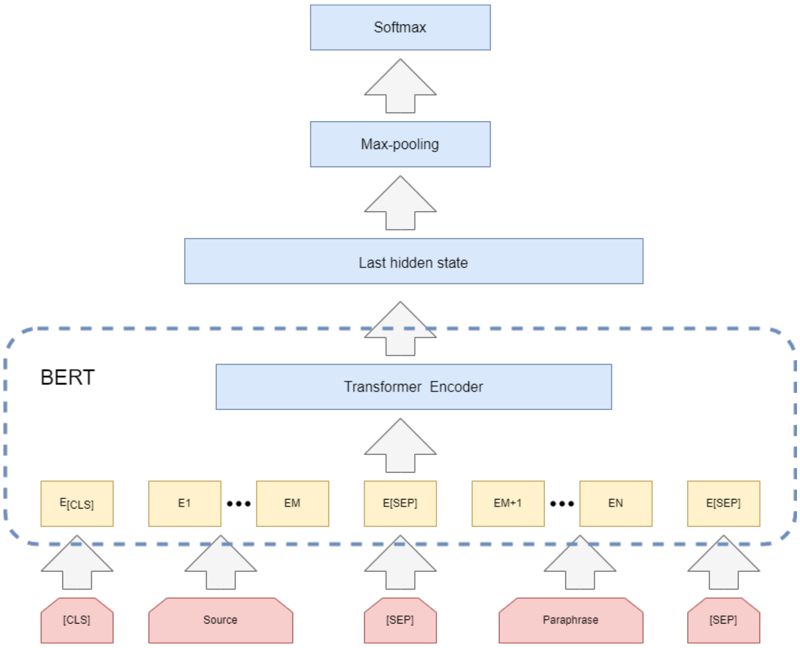

**Figure 9.** BERT-Softmax model.

### *4.4. Answer Selection Module*

The answer selection module is responsible for integrating the two processes that came before it. To get the best answer, when our entity can match the corresponding predicate, we calculate the score $S^e$ of the candidate entity and the score $S^p$ of the candidate predicate. The final score $S$ is obtained by weighting $S^e$ and $S^p$. The formula is as follows.

$$\mathcal{S} = \theta \times \mathcal{S}^e + (1-\theta) \times \mathcal{S}^p \tag{19}$$

where $\theta$ is a hyperparameter . We choose the highest score as the best matching relationship and acquire the answer by querying the KB.

## **5. Experimental Results**

### *5.1. Data and Preprocessing*

#### 5.1.1. Datasets

The NLPCC-ICCPOL 2016 KBQA task offers the knowledge base adopted in this paper. It is the first large-scale general knowledge base in the Chinese domain, containing 6,502,738 entities, 587,875 attributes, and 43,063,796 triples. The knowledge base is essentially a collection of triples (entity, attribute, attribute value). Some examples of triples and dataset annotation are presented in Table 1 and Table 2, respectively.

**Table 1.** Triads in the Knowledge Base.

| **Subject** | **Predicate** | **Object** |
|---|---|---|
| Journey to the West | Author | Wu Chengen |
| Higher mathematics | Press | Wuhan University Press |
| Carotenoids | Nature | Pigment |
| Hamlet | Director | Michelle Amiriad |
| Facebook | Field | Social networking site |
| … | … | … |

**Table 2.** Dataset sample annotation sample.

| Tagging Content | |
|---|---|
| Question | Who wrote the novel Journey to the West? |
| Named entity recognition | Journey to the West |
| Triple | Journey to the West\|\|Author\|\|Wuchengen |

### 5.1.2. Data Preprocessing

The data in the knowledge base are directly and automatically extracted from the Infobox of Baidu Encyclopedia, so there is much noise, especially in the attribute part, which often appears as useless characters, bringing interference to standard experiments. Therefore, before designing and starting the experiment, this paper firstly carries out a certain degree of de-noising on the knowledge base, such as converting traditional Chinese into simplified Chinese, removing redundant spaces in the relation of triples, truncating the entity names that are too long, converting English letters to lowercase for the experiment, etc. The subtask data set is divided as shown in Table 3.

**Table 3.** Dataset statistics.

| The Data Set | The Training Set | The Validation Set | The Test Set |
|---|---|---|---|
| NLPCC 2016 | 13,609 | 1000 | 9870 |
| Named entity recognition | 13,609 | 1000 | 9870 |
| Entity disambiguation | 24,985 | 4744 | 36,208 |
| Relational prediction | 76,009 | 10,600 | 121,958 |

Triples of the same entity in the knowledge base may be distributed in various places in the file of the knowledge base. To enhance the efficiency of the knowledge base query, this paper collects all triples about the same entity and creates index files for the knowledge base. The format of each line in the index file is entity name, start position, and content length. It indicates the start position of all content of each entity in the knowledge base and the total length of the content (both in bytes). When searching for an entity, it is firstly obtained from the index file, and all the information about the entity can be discovered directly from the knowledge base without traversing the knowledge base from start to finish, leading to considerably enhancements in the query efficiency of the knowledge base [48].

## 5.2. *Environment Setup*

For evaluation purpose, we utilize the Precision ($P$), Recall ($R$), and $F1$ scores as metrics in the experiment. The formula is shown as follows.

$$\begin{aligned} P &= \frac{TP}{TP+FP} \\ R &= \frac{TP}{TP+FN} \\ F1 &= \frac{2*P*R}{P+R} \end{aligned} \tag{20}$$

For the hyperparameter settings for NER, we tune it on the basis of the performance of the development set. We set the character embedding size $d_e$ to 100, the dimension of the LSTM hidden state $d_h$ to 120, the initial learning rate to 0.001, the loss weight coefficient $\lambda$ to 0.06, and the dropout rate to 0.3, the number of projections h is set to 8. For the initialization of trainable parameters, use an initializer to initialize the parameters. In terms of entity disambiguation and predicate mapping, a BERT-base model has 12 layers, 768 actual states, 12 heads, and a total number of parameters of 110 M. For fine-tuning of BERT, all hyperparameters are tuned on the development set. For the dataset, the maximum sequence length is set to 60, and the batch size is set to 32. ADAM (Adaptive Moment Estimation) is used for optimization, $\beta_1$ = 0.9, $\beta_2$ = 0.999, the dropout rate is set to $1 \times 10^{-5}$, and the initial learning rate of BERT-Softmax is set to $5 \times 10^{-5}$ by applying a learning rate warm-up strategy. The training epoch of BERT-Softmax is 3. In the answer selection module, the hyperparameter $\alpha$ is set to 0.6. For all baseline models, the word embeddings in the experiments are pre-trained by the word2vec toolkit on the Baidu Encyclopedia corpus, where the size of embedding is set to 300.

### *5.3. Entity Link Experiment*

Table 4 shows the experimental results of entity reference recognition, and P, R, and F values are used as evaluation indicators. The results suggest that the model can achieve accurate outcomes. Traditional NER tasks need to distinguish different types of entities and determine the boundary positions at both ends of the entities. In contrast, entity mention recognition sub-tasks only need to identify the locations mentioned by the entities and do not need to distinguish the types, and simple questions that usually only contain a single entity mention, which reduces the difficulty of this sub-task. In the entity disambiguation subtask training set, the ratio of positive and negative cases was 1:5, and all candidate entities were selected for prediction in the verification set and test set. The sub-task is prone to over-fitting in the training process, so fewer training rounds and lower learning rates are selected. The evaluation indexes were Acc@N. The formula is shown as follows. For the evaluation set of Q questions, each question predicts the set of $N$ candidate answers, $C_i$ that contains the correct answer $A_i$ is recorded as 1, otherwise 0. Each question is summed and then averaged. The average precision is obtained. In this paper, we merely focus on the mean accuracy for $N = 1, 2, 3$.

$$\text{Accuracy@N} \ = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \delta(C_i, A_i) \tag{21}$$

Table 5 presents the experimental results of solid disambiguation on the test set. Among them, the introduction of relational features in the Acc@1 index increased by about 10%. This is because the benchmark model that only uses questions and entity names as input has less available information. At the same time, the relationship features contain topological information of topic entities in the knowledge graph and their associated relationship content, thus enhancing the matching degree between topic entities and questions, effectively improving entity disambiguation. Moreover, the feature extraction capability of CNN has also brought improvement. By reserving the top three candidate entities, a high accuracy rate can be acquired, which is helpful in reducing the size of the candidate relation set for relationship prediction and enhancing the efficiency of the question answering system. To sum up, entity linking can accurately identify the entities mentions and accurately and link effectively to the entities in the knowledge graph.

**Table 4.** Named Entity Recognition.

| **Models** | **F1** |
|---|---|
| BiLSTM-CRF | 90.28 |
| BERT-BiLSTM-CRF | 96.90 |
| Ours | 98.66 |

**Table 5.** Entity disambiguation.

| **Models** | **Accuracy@1** | **Accuracy@2** | **Accuracy@3** |
|---|---|---|---|
| Siamese BiLSTM | 87.85 | 92.58 | 94.59 |
| Siamese CNN | 88.04 | 92.68 | 94.88 |
| BERT-CNN | 89.14 | 93.19 | 95.05 |

### *5.4. Relational Prediction Experiment*

The performance of the BERT-Softmax model for predicate mapping is shown in Table 6. We demonstrate that a higher performance is attained, suggesting that the fine-tuned BERT model may extract deeper semantic knowledge compared with other neural network models.

**Table 6.** Predicate Mapping.

| Models | Accuracy@1 | Accuracy@2 | Accuracy@3 |
|---|---|---|---|
| Siamese BiLSTM | 92.54 | 96.74 | 98.12 |
| Siamese CNN | 86.47 | 93.80 | 96.16 |
| BERT-Softmax | 94.81 | 97.68 | 98.60 |

### *5.5. Comparison with the Baseline Model*

We compare the developed model with the baseline models published in the NLPCC-ICCPOL 2016KBQA task. Table 7 presents the outcomes from the NLPCC-ICCPOL 2016KBQA task and it is found that the developed model outperforms all the other ones, indicated by the higher index F1 value. Because the model adopted combines the BERT model, the training time is longer.

**Table 7.** NLPCC-ICCPOL 2016 KBQA result .

| Models | Averaged F1 |
|---|---|
| Baseline model(C-DSSM) | 52.47 |
| Wang et al. [49] | 79.14 |
| Xie et al. [8] | 79.57 |
| Lei et al. [50] | 80.97 |
| Zhou et al. [51] | 81.06 |
| Yang et al. [9] | 81.59 |
| Xie et al. [52] | 82.43 |
| Lai et al. [10] | 82.47 |
| Liu et al. [11] | 84.12 |
| Ours | 87.74 |

## 6. Discussion

Table 7 presented performance comparison between our method and other published methods. The systems of Lai et al. [10], Xie et al. [52], and Yang et al. [9] are the top three for the NLPCC 2016 KBQA evaluation task, and they all combine neural networks and manually constructed rules to ensure the quality of question and answering. BB-KBQA [11] fine-tunes the pre-training task based on BERT to achieve three subtasks of NER, entity disambiguation, and predicate mapping, but the performance on the first two subtasks is weaker than the model of this paper, resulting in a lower final performance than the KBQA in this paper. The results of experiments demonstrates that the BAT-KBQA can achieve an Averaged F1 value of 87.74%, which achieves the best results and improves the question answering system accuracy compared with other published methods.

Finally, we analyzed the effect of the CWS task of this work, and the results show that word boundary information from the CWS task is effective for the Chinese NER task. Particularly when various entities appear at the same time, our model can correctly classify the word in different scenarios. Then, the performance of the question-answering system is evaluated. The results are found accurate except for some triples and wrong answers. For example, for the question "Where did Jie Wang debut?" (in Chinese), corresponding to the knowledge triple <Jie Wang, debut place, Taiwan>, our system predicts the subject entity and predicate as <Jie Wang (male singer from Hong Kong and Taiwan), debut place>, indicating that the entity disambiguation module can correctly select the answer Jie Wang, who is a singer. For the question "Which factory built submarine type 212?" (in

Chinese), the corresponding knowledge triple is <212 type submarine, built, Hathaway Shipyard (hdw)>, and the subject entity and predicate predicted by our system is <212 type submarine, manufacturing plant>, which shows that although the predicate chosen by the predicate matching module is different from the labeled predicate, the correct predicate is chosen by understanding the intention of the question, and the correct answer is finally found. In practice, the method employs some artificial rules because of the introduction of CWS task, which requires data annotation such as the NER task. Therefore, additional effort should be spent on annotation and the need for manual labeling data is sometimes tedious. However, labeling is relatively easy and within an acceptable range.

In summary, we can see that introduction of CWS in the NER task can enhance the accuracy of entity linking. In our work, entity disambiguation and predicate mapping combined with BERT pre-training exhibit better performance, and the entities also have a strong correlation with the predicates. Compared to other models that use more complex features and artificial rules, our KBQA system achieve better results with only neural network models and a small number of simple text features. Therefore, our system achieves sufficient accuracy and can effectively answer the questions asked by the users.

## 7. Conclusions

In this paper, by fusing CWS in the NER task, the accuracy of candidate entities is improved without introducing too much noise. A multi-channel entity disambiguation model is proposed to enhance the features of candidate entities to bridge the semantic gap between the question and the knowledge base. The model makes full use of the problem encoding and candidate information features extracted by the BERT model and has strong generalization for application in multi-domain knowledge bases. The current method is evaluated from experiments on the NLPCC-ICCPOL-2016KBQA Chinese open question and answering dataset, with an average F1 score of 87.74%.

The results of experiments indicate that we have developed a satisfactory question answering system for the Chinese dataset. Our method performs better on the NLPCC-ICCPOL 2016 KBQA dataset. The system we developed may fill the gap in the technical research of the tutor selection service system. Later, we will use the BAT-KBQA framework on the Chinese Academy of Sciences tutor dataset to develop a tutor KBQA for students, so that students can quickly select tutors.

In future work, different subtasks will be trained in combination, or more knowledge graph representation learning methods can be introduced to obtain richer features.

**Author Contributions:** Conceptualization, S.D.; Data curation, Q.Z. and H.W.; formal analysis, Y.L. and Q.Z.; investigation, S.D. and Y.L.; Funding acquisition, S.D.; Supervision, S.D.; resources, K.D., S.D. and Y.Z.; software, K.D.; writing—original draft preparation, K.D. and S.D.; writing—review and editing, S.D. and Y.Z.; visualization, H.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Key R&D Program of China (Grant No. 2021YFB3500100), the National Natural Science Foundations of China (Grant Nos. 52250005, 21875271, U20B2021, 21707147, 51372046, 51479037, 91226202, and 91426304), the Key R&D Projects of Zhejiang Province (No. 2022C01236, 2019C01060), the Entrepreneurship Program of Foshan National Hi-tech Industrial Development Zone, the Major Project of the Ministry of Science and Technology of China (Grant No. 2015ZX06004-001), Ningbo Natural Science Foundations (Grant Nos. 2014A610006, 2016A610273, and 2019A610106).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## References

1. Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P.N.; Hellmann, S.; Morsey, M.; Van Kleef, P.; Auer, S.; et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semant. Web* **2015**, *6*, 167–195. [CrossRef]
2. 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.
3. Suchanek, F.M.; 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.
4. Duan, N. Overview of the nlpcc-iccpol 2016 shared task: Open domain chinese question answering. In *Natural Language Understanding and Intelligent Applications*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 942–948.
5. Chen, D.; Fisch, A.; Weston, J.; Bordes, A. Reading Wikipedia to Answer Open-Domain Questions. *arXiv* **2017**, arXiv:1704.00051.
6. Unger, C.; Freitas, A.; Cimiano, P. An introduction to question answering over linked data. In Proceedings of the Reasoning Web International Summer School, Athens, Greece, 8–13 September 2014; pp. 100–140.
7. Wang, L.; Yu, Z.; Liu, T. A Deep Learning Approach for Question Answering Over Knowledge Base. In Proceedings of the International Conference on Computer Processing of Oriental Languages National Ccf Conference on Natural Language Processing & Chinese Computing, Kunming, China, 2–6 December 2016.
8. Xie, Z.; Zeng, Z.; Zhou, G.; He, T. Knowledge base question answering based on deep learning models. In *Natural Language Understanding and Intelligent Applications*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 300–311.
9. Yang, F.; Liang, G.; Li, A.; Huang, D.; Chou, X.; Liu, H. Combining Deep Learning with Information Retrieval for Question Answering. In Proceedings of the International Conference on Computer Processing of Oriental Languages National CCF Conference on Natural Language Processing and Chinese Computing, Kunming, China, 2–6 December 2016.
10. Lai, Y.; Lin, Y.; Chen, J.; Feng, Y.; Zhao, D. Open domain question answering system based on knowledge base. In *Natural Language Understanding and Intelligent Applications*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 722–733.
11. Liu, A.; Huang, Z.; Lu, H.; Wang, X.; Yuan, C. BB-KBQA: BERT-Based Knowledge Base Question Answering. In Proceedings of the Chinese Computational Linguistics, Kunming, China, 18–20 October 2019.
12. Wu, W.; Deng, Y.; Liang, Y.; Lei, K. Answer Category-Aware Answer Selection for Question Answering. *IEEE Access* **2020**, *9*, 126357–126365. [CrossRef]
13. Weizenbaum, J. ELIZA—A computer program for the study of natural language communication between man and machine. *Commun. ACM* **1966**, *9*, 36–45. [CrossRef]
14. Colby, K.M.; Weber, S.; Hilf, F.D. Artificial paranoia. *Artif. Intell.* **1971**, *2*, 1–25. [CrossRef]
15. Ren, M.; Huang, H.; Gao, Y. SKR-QA: Semantic ranking and knowledge revise for multi-choice question answering. *Neurocomputing* **2021**, *459*, 142–151. [CrossRef]
16. Ma, H.; Hou, J.; Zhu, C.; Zhang, W.; Tang, R.; Lai, J.; Zhu, J.; He, X.; Yu, Y. QA4PRF: A Question Answering Based Framework for Pseudo Relevance Feedback. *IEEE Access* **2021**, *9*, 139303–139314. [CrossRef]
17. Tang, Y.; Han, H.; Yu, X.; Zhao, J.; Liu, G.; Wei, L. An Intelligent Question Answering System based on Power Knowledge Graph. In Proceedings of the 2021 IEEE Power & Energy Society General Meeting (PESGM), Denver, CO, USA, 17–21 July 2022; pp. 1–5.
18. Vrandečić, D.; Krötzsch, M. Wikidata: A free collaborative knowledgebase. *Commun. ACM* **2014**, *57*, 78–85. [CrossRef]
19. 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 International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Arras, France, 27–30 June 2017; pp. 428–438.
20. Luo, X.; Liu, L.; Yang, Y.; Bo, L.; Cao, Y.; Wu, J.; Li, Q.; Yang, K.; Zhu, K.Q. AliCoCo: Alibaba e-commerce cognitive concept net. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, Portland, OR, USA, 14–19 June 2020; pp. 313–327.
21. 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, Online, 7–11 November 2021; pp. 3653–3660.
22. Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; Su, Z. Arnetminer: Extraction and mining of academic social networks. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 990–998.
23. Ju, S.G.; Li, T.N.; Sun, J.P. Chinese Fine-grained Name Entity Recognition Based on Associated Memory Networks. *J. Softw.* **2021**, *32*, 2545–2556.
24. Liu, Q.; Li, Y.; Duan, H.; Liu, Y.; Qin, Z. Knowledge graph construction techniques. *J. Comput. Res. Dev.* **2016**, *53*, 582–600.
25. Peng, H.; Gao, T.; Han, X.; Lin, Y.; Li, P.; Liu, Z.; Sun, M.; Zhou, J. Learning from context or names? An empirical study on neural relation extraction. *arXiv* **2020**, arXiv:2010.01923.
26. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. In Proceedings of the 27th Annual Conference on Neural Information, Lake Tahoe, NV, USA, 5–10 December 2013; Volume 26.
27. Hao, Z.; Wu, B.; Wen, W.; Cai, R. A subgraph-representation-based method for answering complex questions over knowledge bases. *Neural Netw.* **2019**, *119*, 57–65. [CrossRef] [PubMed]

28. Meng, M.; Zhang, K.; Lun, B.; Zhang, X. A Semantic Query Expansion Method for Question Answering Based on Knowledge Graph. *Comput. Eng.* **2019**, 45, 276–283+290. [CrossRef]
29. Qiu, Y.; Wang, Y.; Jin, X.; Zhang, K. Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision. In Proceedings of the 13th International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 February 2020; pp. 474–482.
30. Xu, Y.; Zhu, C.; Xu, R.; Liu, Y.; Zeng, M.; Huang, X. Fusing Context Into Knowledge Graph for Commonsense Question Answering. In Proceedings of the Meeting of the Association for Computational Linguistics, Bangkok, Thailand, 1–6 August 2021.
31. Souza, F.; Nogueira, R.; Lotufo, R. Portuguese named entity recognition using BERT-CRF. *arXiv* **2019**, arXiv:1909.10649.
32. Li, X.; Zhang, H.; Zhou, X.H. Chinese clinical named entity recognition with variant neural structures based on BERT methods. *J. Biomed. Informatics* **2020**, *107*, 103422. [CrossRef]
33. Wu, Y.; Huang, J.; Xu, C.; Zheng, H.; Zhang, L.; Wan, J. Research on Named Entity Recognition of Electronic Medical Records Based on RoBERTa and Radical-Level Feature. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 2489754. [CrossRef]
34. Yao, L.; Huang, H.; Wang, K.W.; Chen, S.H.; Xiong, Q. Fine-grained mechanical Chinese named entity recognition based on ALBERT-AttBiLSTM-CRF and transfer learning. *Symmetry* **2020**, *12*, 1986. [CrossRef]
35. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the NIPS 2012: Neural Information Processing Systems Conference, Lake Tahoe, NV, USA, 3–6 December 2012; Volume 25.
36. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
37. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Zettlemoyer, L. Deep Contextualized Word Representations. *arXiv* **2018**, arXiv:1802.05365
38. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 4 May 2020).
39. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
40. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
41. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. In Proceedings of the Thirty-third Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
42. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942 Representations. 2021.
43. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv* **2020**, arXiv:2003.10555.
44. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Chen, X.; Zhang, H.; Tian, X.; Zhu, D.; Tian, H.; Wu, H. Ernie: Enhanced representation through knowledge integration. *arXiv* **2019**, arXiv:1904.09223.
45. Cao, P.; Chen, Y.; Kang, L.; Zhao, J.; Liu, S. Adversarial Transfer Learning for Chinese Named Entity Recognition with Self-Attention Mechanism. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018.
46. Palangi, H.; Deng, L.; Shen, Y.; Gao, J.; He, X.; Chen, J.; Song, X.; Ward, R. Semantic modelling with long-short-term memory for information retrieval. *arXiv* **2014**, arXiv:1412.6629.
47. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural Architectures for Named Entity Recognition. 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.
48. Zhou, B.; Sun, C.; Lin, L.; Liu, B. InsunKBQA: A question-answering system over knowledge base. *Intell. Comput. Appl.* **2017**, *7*, 150–154.
49. Wang, L.; Zhang, Y.; Liu, T. A deep learning approach for question answering over knowledge base. In *Natural Language Understanding and Intelligent Applications*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 885–892.
50. Kai, L.; Yang, D.; Bing, Z.; Ying, S. Open Domain Question Answering with Character-Level Deep Learning Models. In Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 9–10 December 2017.
51. Zhou, B.; Sun, C.; Lin, L.; Liu, B. LSTM Based Question Answering for Large Scale Knowledge Base. *Acta Sci. Nat. Univ. Pekin.* **2018**, *54*, 286–292.
52. Xie, Z.; Zhao, Z.; Zhou, G.; Wang, W. Topic enhanced deep structured semantic models for knowledge base question answering. *Sci. China* **2017**, *60*, 110103. [CrossRef]