

Article

Research on Automatic Question Answering of Generative Knowledge Graph Based on Pointer Network

Shuang Liu ^{1,*}, Nannan Tan ¹, Yaqian Ge ¹ and Niko Lukač ² 

¹ School of Computer Science and Engineering, Dalian Minzu University, Dalian 116600, China; tannn761@gmail.com (N.T.); gey160213@gmail.com (Y.G.)

² Faculty of Electrical Engineering and Computer Science, University of Maribor, SI-2000 Maribor, Slovenia; niko.lukac@um.si

* Correspondence: liushuang@dlmu.edu.cn; Tel.: +86-156-4119-6451

Abstract: Question-answering systems based on knowledge graphs are extremely challenging tasks in the field of natural language processing. Most of the existing Chinese Knowledge Base Question Answering (KBQA) can only return the knowledge stored in the knowledge base by extractive methods. Nevertheless, this processing does not conform to the reading habits and cannot solve the Out-of-vocabulary (OOV) problem. In this paper, a new generative question answering method based on knowledge graph is proposed, including three parts of knowledge vocabulary construction, data pre-processing, and answer generation. In the word list construction, BiLSTM-CRF is used to identify the entity in the source text, finding the triples contained in the entity, counting the word frequency, and constructing it. In the part of data pre-processing, a pre-trained language model BERT combining word frequency semantic features is adopted to obtain word vectors. In the answer generation part, one combination of a vocabulary constructed by the knowledge graph and a pointer generator network (PGN) is proposed to point to the corresponding entity for generating answer. The experimental results show that the proposed method can achieve superior performance on WebQA datasets than other methods.

Keywords: knowledge base question answering; entity recognition; pointer generator network; answer generative model; pre-trained language model



Citation: Liu, S.; Tan, N.; Ge, Y.; Lukač, N. Research on Automatic Question Answering of Generative Knowledge Graph Based on Pointer Network. *Information* **2021**, *12*, 136. <https://doi.org/10.3390/info12030136>

Academic Editors: Pierpaolo Basile and Annalina Caputo

Received: 24 February 2021

Accepted: 19 March 2021

Published: 21 March 2021

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



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Natural language processing is a hot research field in artificial intelligence. A question-answering system, as a sub-field of natural language processing, studies how to interact with machines naturally. In 1950, Turing [1] published an article titled “Computing Machinery and Intelligence” in the *Mind* journal. A classic Turing Test is proposed by him [2]. That is, testing the machine by means of human-machine question answering to see whether a machine can think and speak like a human. If it is impossible to distinguish whether it is a machine or a human after a certain round of conversation, the machine can be considered intelligent enough. Since then, scholars have done a lot of related research. The data sources of traditional search engines are often some web documents and other unstructured data, but with the advent of the era of big data [3], this method can no longer effectively deal with the problem of rapid increase in the amount of information, we need a new data processing method to solve this problem. In order to better meet the needs of this aspect, the knowledge graph technology is proposed [4].

The knowledge graph is essentially a semantic network. Its nodes represent entities or concepts, and the edges represent semantic relationships between entities or concepts. The concept of knowledge graph was proposed by Google [4] in May 2012. Its original intention was to improve the search capabilities and results of search engines, and to enhance the user experience. Immediately thereafter, high-quality large-scale knowledge graphs at home and abroad have been developed rapidly, and knowledge graphs in many

general fields have been produced one after another. For example: English knowledge base YAGO [5], Wikidata [6], DBPedia [7], etc., Chinese knowledge bases XLORE [8], zhishi.me [9], Ownthink [10], etc. The emergence of these structured knowledge bases has accelerated the process of intelligent automatic question answering.

The goal of the knowledge base question answering (KBQA) is to find the corresponding one or more triples in the knowledge base after receiving the user's natural language question to introduce knowledge and help complete the answer. However, KBQA faces two important challenges. First, the answer cannot be returned when the entities stored in the knowledge base cannot meet the needs of the question. Second, the result returned by the current KBQA is the entity stored in the knowledge base. So, the answer is short, and it does not contain other information and does not conform to people's reading habits. For example, if the user enters the question "Which is the largest city in France?" and gets the answer "Paris". However, the users often hope that the question answering system can return a sentence containing the answer, such as "The largest city in France". Based on the above analysis, this paper draws on the idea of machine reading comprehension, and combines generative methods to solve related problems on the basis of KBQA.

In recent years, many researchers have proposed the method of combining heterogeneous data to achieve question answering. Rajarshi et al. [11] proposed to use a large amount of text to make up for the problem of the lack of knowledge base, extending the general mode to the application of natural language question answering, and using memory network to focus on a large number of facts combining text and knowledge base. Sun et al. [12] emphasized the association between entities and text in the knowledge base, and proposed the GRAFT-Net model, which is used to extract answers from question-related subgraphs and linked text. Wang et al. [13] proposed a knowledge-based dialogue generation model, which transferred question representation and knowledge matching from the knowledge base question answering task to promote discourse understanding and factual knowledge selection in the process of dialogue generation. Recently, end-to-end training has become a popular method to achieve KBQA. Xiong et al. [14] proposed a new end-to-end question answering system model, which accumulates the knowledge of entities from the subgraph of the knowledge base related to the question, and then re-states the question in the latent space, and reads the text with the accumulated entity knowledge. In addition, the method finally obtains the evidence information for predicting similarity from the knowledge base and text. Wei et al. [15] proposed a new method based on multi-instance learning to solve the problem of noisy answers by exploring the consistency between the answers to the same question in the training end-to-end KBQA model. These methods can only extract knowledge from existing data and return them as answers, and the returned results are simple. To solve these problems, this paper proposes to use the end-to-end generative question answering model for knowledge graph question answering. The proposed method also introduces word frequency semantic features for feature extraction to further enhance the importance of entities in the text. The specific idea is to use the deep learning model to identify the entity in the source text in the dataset first to count the relationship and the tail entity pointed to by the entity in the knowledge graph, and then to use the Term Frequency (TF) algorithm to count the occurrence frequency of the entity to construct a knowledge dictionary. Then BERT pre-trained language model is adopted and word frequency semantic feature splicing is used to obtain word vectors. Finally, the pointer generator network (PGN) [16] is used to conduct generative automatic question answering with the knowledge vocabulary generated by the triples in the knowledge graph. In summary, the main contributions of this paper are as follows:

- BERT pre-trained language model is adopted to obtain the word vectors of the question sentence, combining the semantic features of the word frequency of the entity words in the question sentence, which are combined together as the input sequence.
- Knowledge graph combined with the source text information is used to construct a vocabulary as a pointer to generate a soft link of the network, and point to the corresponding entity for fusion to generate the corresponding answer.

The rest of this paper is arranged as follows. Section 2 introduces the related work of generative question answering based on knowledge graph. Section 3 introduces the overall implementation scheme of generative question answering based on knowledge graph. Section 4 gives specific experiments and analysis. Section 5 summarizes the paper and gives future prospects of the research.

2. Related Work

At present, the research method of question answering based on knowledge graph has gradually changed from the previous research based on semantic analysis [17,18] to the research of question answering based on deep learning derived from information retrieval methods [19–21]. Methods based on semantic analysis often require vocabulary mapping and construction of a syntax tree to transform natural language into a semantic representation that can be understood by machine language, so as to perform reasoning or query to get the correct answer. However, this type of method requires researchers to be familiar with natural language related knowledge and also need to label a large amount of data. It is not suitable for large-scale knowledge base question answering tasks and has low generalization ability. The methods based on information retrieval start from the entities and relationships in the question, search for relevant paths in the knowledge base as candidate answers, and finally select the best answer through the classification model or the ranking model. In recent years, with the development of deep learning technology, it has significant advantages over traditional methods in terms of answer selection effects, and it has become a hot spot for research scholars. The above three methods are the traditional mainstream methods. In recent years, many researchers have made new improvements to these three methods or proposed new research methods. For example, Bordes et al. [22] proposed to apply the memory network to the knowledge graph question answering, making full use of the scalability and generalization of the memory network in combination with the knowledge base such as FreeBase [23]. Sorokin et al. [24] introduced the gated neural network (GGNN) to encode the graph structure with semantic analysis in response to the semantic analysis structure problem of the complicated problem that was easy to be ignored in the past.

Currently, the method of automatic question answering has been advanced to an end-to-end method. Lukovnikov et al. [25] proposed an end-to-end approach, training an encoder that can handle out-of-vocabulary(OOV) and rare word problems. At the same time, it can use character-level semantic encoders to extract different semantic information, and achieved good results. Here, the OOV problem is a problem often encountered in language models, that is, words that are beyond the scope of the vocabulary appear in the dataset, and the rare word is a special vocabulary in a specific field, such as a relatively uncommon surname. Xiong et al. [14] studied the end-to-end question answering model combining inadequate knowledge bases with the results of unstructured text retrieval, and achieving a good improvement in simple question answering questions and complex questions. Wang et al. [26] added a verification mechanism for the reliability evaluation and inspection of entity and prediction relationships on the basis of the existing KBQA question answering framework, which strengthened the reliability of relationship prediction, and applied the Sequence-to-Sequence(Seq2Seq) [27] framework to multi-relationship KBQA problems. Among them, Seq2Seq is also called Encoder-Decoder model. Encoder is used to encode sequence information and encode sequence information of any length into a vector, and Decoder is a decoder. After the decoder gets the context information vector, it can decode the information and output it as a sequence. The above methods are all based on search-type answers. Its characteristic is to be able to extract entities or relationships from the knowledge base as the answer to return, so the answer is precise and concise. In order to ensure users have a comfortable experience on the basis of ensuring the accuracy of the answers, this paper adopts the method of generative question answering. The difference between generative question answering and extractive question answering is that it is often used in reading comprehension style question answering system. Its main feature is that it

can generate corresponding answers to users' questions after reading a specified article or paragraph. However, the answers generated by this type of question answering often have frequent repetitive questions. Therefore, this paper combines knowledge graph with generative question answering.

At present, there are relatively few researches on the generative question answering of the knowledge graph. In this paper, the PGN is introduced into the knowledge graph automatic question answering to form an "extraction-generation" question answering framework.

3. Construction of Generative QA System

The overall implementation process of this paper is shown in Figure 1, which consists of three parts, namely the knowledge vocabulary construction module, the word vectors acquisition module, and the generative model building module. Here, vocabulary of knowledge is used in the generative model, so as to select entities with high probability in the vocabulary as the next word to be generated. In the knowledge vocabulary construction module, named entity recognition BiLSTM-CRF [28] model is used to identify entities that appear in the source text and question sentences, finding the relationship and tail entities that the entity points to in the knowledge base, counting the entities separately, then separately counting the frequency of each part of all triples corresponding to the entity. In the word vectors acquisition module, the pre-trained language model BERT [29] is used to obtain the word vectors of the question sentence and then spliced with the word frequency semantic features of the entity in the question sentence as the input sequence. In the generative model building module, a PGN model is introduced to determine whether to generate vocabulary from the vocabulary or copy vocabulary from the question sentence and return it as an answer.

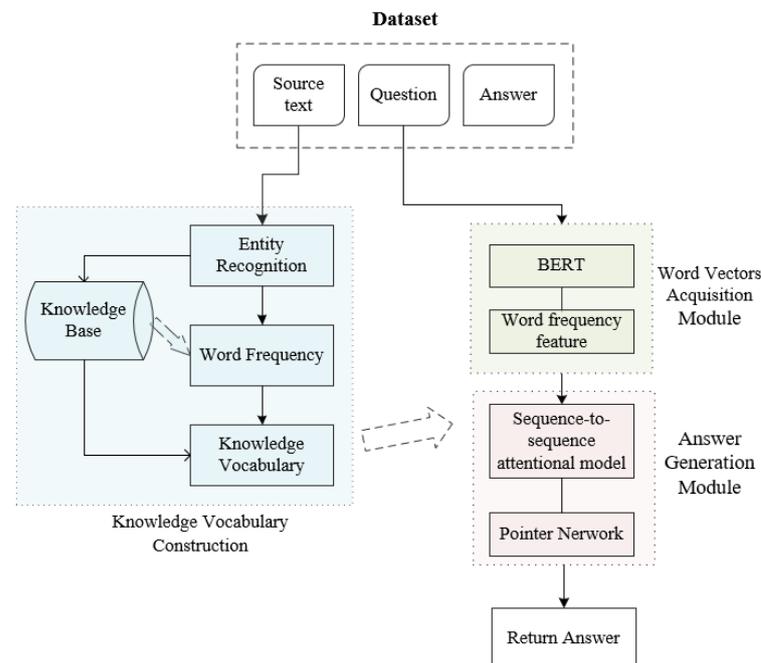


Figure 1. The realization process of this article.

Here, the construction method of the knowledge vocabulary is given in Section 3.1. The model and details of answer generation are described in Section 3.2.

3.1. Knowledge Vocabulary Construction

The applied question answering dataset is Baidu's public WebQA dataset [30], while the knowledge base uses CN-DBpedia [31]. For the triple information in the knowledge

base, the large-scale import tool neo4j-admin-import provided by Neo4j [32] is adopted to import the information into the Neo4j graph database.

The format of the content stored in the WebQA dataset is source text, question, answer. In order to build a vocabulary of knowledge covering the entire dataset, a named entity recognition model is used to identify the source text in the dataset. First, jieba [33] word segmentation tool is used to segment the text data to be recognized, remove stop words and other operations to reduce the occurrence of errors. Here, the preprocessed data are manually filtered to create a custom dictionary, and then the BiLSTM-CRF [28] neural network model is used to identify entities in the original text. The identified entity is queried for its corresponding head entity or tail entity through the constructed cypher sentence. If the entity does not exist in the knowledge base, it will be stored in the vocabulary after manual inspection. Furthermore, TF algorithm is used to count word frequency.

3.1.1. Entity Recognition

Chinese named entity recognition is an important task in the field of Chinese natural language processing. However, Chinese entity names are highly context-dependent, so the task is extremely challenging. In order to be able to match the golden entity, the deep learning model for named entity recognition is used. Here, the BiLSTM-CRF [28] model is selected, which mainly contains two parts, namely the Bidirectional Long Short-Term Memory (BiLSTM) [34] module and the Conditional Random Field (CRF) [35] module. The model framework is shown in Figure 2. The specific steps are as follows. The bidirectional cyclic neural network with LSTM [36] unit is used to extract the features of the input sequence information, and finally connecting the LSTM results in the two directions and input them to the CRF layer. Then, CRF is used as the output layer of the model to generate text sequence annotation results.

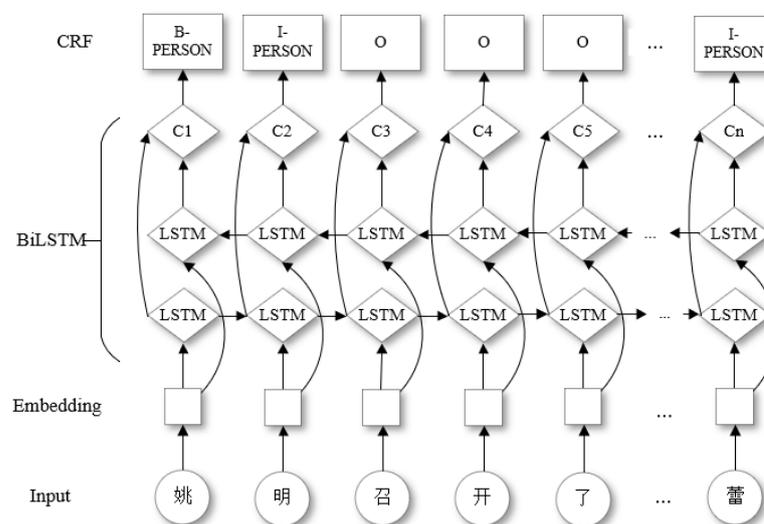


Figure 2. The main framework of the entity recognition model. The sequence of Chinese characters entered was: “Yao Ming held the first media meeting after returning to China and announced his daughter’s name. It turns out that Yao Ming’s daughter is Yao Qinlei”.

1. BiLSTM Module

LSTM is a variant of RNN [37]. According to its special design, it can solve the vanishing gradient or exploding gradient generated during RNN training. It cleverly uses the concept of gating to effectively capture long sequence information. The formula of an LSTM structure is as follows:

$$i_t = \sigma(W_i \cdot x_t + V_i \cdot h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_f \cdot x_t + V_f \cdot h_{t-1} + b_f) \quad (2)$$

$$o_t = \sigma(W_o \cdot x_t + V_o \cdot h_{t-1} + b_o) \quad (3)$$

$$\tilde{c}_t = \tanh(W_c \cdot x_t + V_c \cdot h_{t-1} + b_c) \quad (4)$$

$$c_t = i_t \otimes \tilde{c}_t + f_t \otimes c_{t-1} \quad (5)$$

$$h_t = o_t \otimes \tanh(c_t) \quad (6)$$

Among them, x_t is the unit input, σ is the activation function, i_t, f_t, o_t represent the input gate, forget gate, and output gate at time t , while W and b represent the weight matrix and bias vector. \tilde{c}_t represents the update status at time t while h_t represents the output at time t .

Compared with LSTM, BiLSTM adopts forward and backward LSTMs for each word sequence, and then obtains the final hidden layer representation through vector splicing. Therefore, BiLSTM can process context information at the same time and improve the performance of named entity recognition. The specific expression is as follows:

$$h_t = [\vec{h}_t, \overset{\leftarrow}{h}_t] \quad (7)$$

2. CRF Module

CRF combines the characteristics of the maximum entropy model and the hidden Markov model, and can consider the dependencies between tags to obtain the global optimal tag sequence. This can make up for the shortcomings of BiLSTM. The probability of CRF for a given sequence $X = (x_1, x_2, \dots, x_n)$ and corresponding predicted sequence $Y = (y_1, y_2, \dots, y_n)$ can be expressed as:

$$s(X, Y) = \sum_{i=0}^n A_{y_i, y_{i+1}} + \sum_{i=1}^n P_{i, y_i} \quad (8)$$

Here, A represents the transfer score matrix, and $A_{i,j}$ represents the score of label i transferred to label j .

The formula for predicting the probability of sequence Y is as follows:

$$P(Y|X) = \frac{e^{s(X, Y)}}{\sum_{\tilde{Y} \in Y_X} s(X, \tilde{Y})} \quad (9)$$

The formula of the likelihood function of the marker sequence during training is as follows:

$$\ln(p(Y | X)) = s(X, Y) - \ln\left(\sum_{\tilde{Y} \in Y_X} s(X, \tilde{Y})\right) \quad (10)$$

where Y_X represents all possible labeling sequences. The formula for the output sequence with the largest score after decoding is as follows:

$$Y^* = \underset{\tilde{Y} \in Y_X}{\operatorname{argmax}}(X, \tilde{Y}) \quad (11)$$

3.1.2. Vocabulary Construction

By observing the recognition results, it can be found that there are some entity types that are difficult to recognize, such as movie names. In response to this problem, this paper specifies corresponding rules for entity types to filter, such as using common surname dictionaries to match names, and formulating date formats to identify dates. The entity types that cannot be set by the above rules are all set as mixed entities. Then, the cypher statement is used to query the corresponding entity in the knowledge graph (relationship, tail entity). Then the TF algorithm is used to count the number of occurrences of each part

in the dataset (head entity, relationship, tail entity). The purpose of using the TF algorithm is to count the word frequency of the entity or relationship to emphasize its importance in the source text. If the entity or relationship does not exist in the knowledge graph, it will be stored in the dictionary, which can reduce the problem of OOV.

3.2. Answer Generation Model

The current question answering tasks are mostly done in extractive manner. In addition, the answer is a series of fragments in the text, or from the triples corresponding to the entities in the question sentence. This will cause duplicate answers or OOV phenomenon [38,39]. To solve this problem, this paper implements a joint training method with extractive and generative joints to get the answer. The model includes word vectors acquisition and PGN model. The answer generation module model is shown in Figure 3.

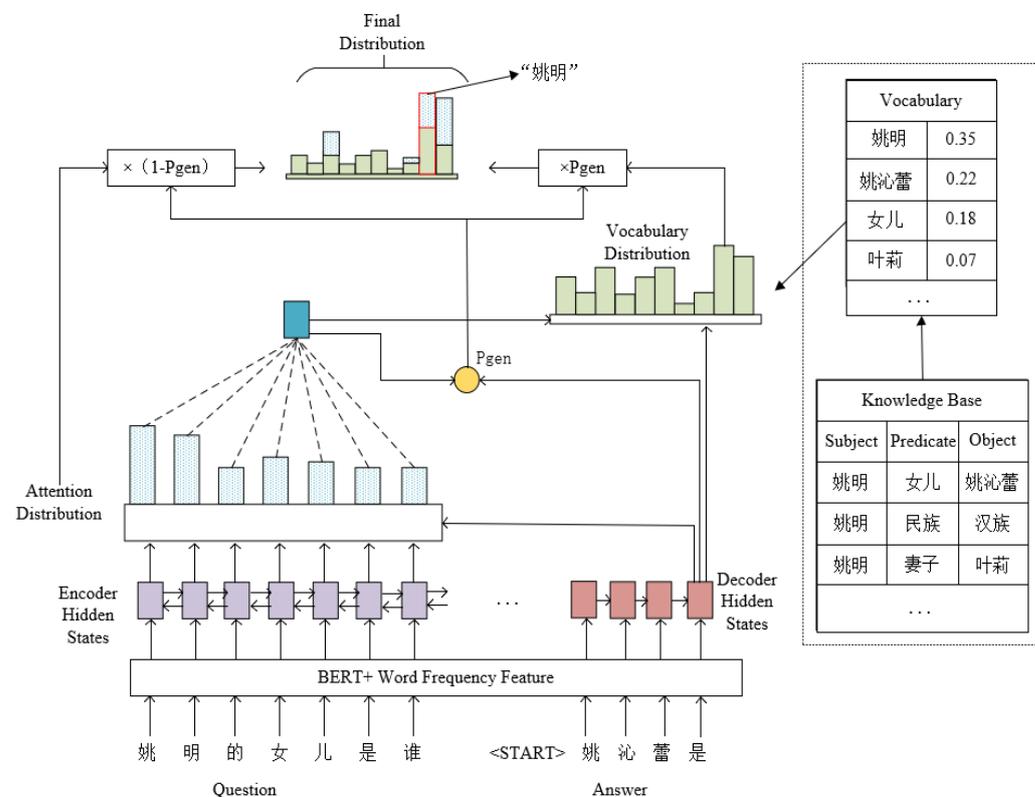


Figure 3. Answer generation model structure diagram. “姚明的女儿是谁” (in Chinese): Who is Yao Ming’s daughter?; “姚沁蕾是” (in Chinese): Yao Qinlei (Yao Ming’s daughter) is; “叶莉” (in Chinese): Ye Li (Yao Ming’s wife); “汉族” (in Chinese): Han nationality.

3.2.1. Word Vectors Acquisition

In order to better obtain the semantic information of Chinese questions, the BERT pre-trained language model is used to obtain the word vectors of the questions in the word vectors acquisition stage, and the word frequency feature is used to give the question sentences. The entities are scored, and then the two are joined as the input sequence.

1. BERT Module

At present, the research on language models has experienced one-hot, Word2Vec [40], ELMO [41], GPT [42] to BERT [29]. The first few language models have certain defects. For example, the word vector trained by the Word2Vec model is a static word vector and cannot represent a word with multiple meanings; GPT is a one-way language model and cannot obtain the context of a word. BERT has strong semantic representation capabilities. BERT is a model structure of a multilayer bidirectional Transformer encoder. The model structure

diagram is shown in Figure 4. Transformer encoder pays attention to the contextual information on both sides. The input of the BERT model is obtained by adding together position embedding, segment embedding, and word embedding. In addition, the head and tail tokens in each question are represented by special identifiers [CLS] and [SEP], respectively, which are used to disconnect two sentences in the corpus. The output of the model is the semantic representation of the corresponding fusion context information after each word vectors passes through the Transformer encoder.

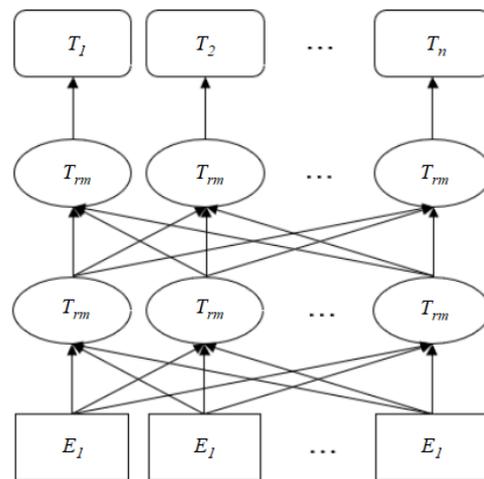


Figure 4. BERT pre-trained language model structure.

Among them, the most important part of BERT is the two-way Transformer encoder. Transformer abandons the cyclic network structure of RNN and models the text entirely based on the attention mechanism. The key to the encoder is the self-attention mechanism, as shown in the formula:

$$attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (12)$$

Among them, Q , K , and V are the input word vectors matrix, and d_k is the input vector dimension. In order to expand the model's ability of focusing on different positions, the transformer adopts a multi-head mode and finally stitches different attention results together, as shown in the following formula:

$$multihead(Q, K, V) = concat(head_1, head_2, \dots, head_n)W^O \quad (13)$$

$$head_i = attention(QW_i^Q, KW_i^K, VW_i^V) \quad (14)$$

Among them, the fully connected feedforward network in the Transformer structure has two layers of dense, the first layer is the RELU activation function, and the second layer is the linear activation function. If the output of the multi-head attention mechanism is expressed as Z , FFN can be expressed as:

$$FFN(Z) = max(0, ZW_1 + b_1)W_2 + b_2 \quad (15)$$

Using the BERT pre-trained language model allows the word vectors to obtain the semantic information of contextual interaction and better express the content of the question.

2. Frequency Feature

In view of the high frequency of keywords in the text, the traditional word frequency feature is used to describe the text to improve the performance of the important entities or relationships in the question. The word frequency feature represents the number of times a word appears in the text, and it can reflect the theme of the text. The basic TF algorithm is

used to count the number of times the entity or relationship in the currently input question appears in the source text. The method is as follows:

$$tf_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (16)$$

Among them, $n_{i,j}$ represents the number of occurrences of the entity in the text, and the denominator represents the sum of the number of occurrences of all words in the text.

3.2.2. Pointer-Generator Network Model

Pointer-Generator Networks [16] is a combination of the pointer network [43] and the Encoder-Decoder model based on Attention. It allows pointers to point to the generated words, and also allows the generation of words through a fixed vocabulary. In this paper, the question sentence is first passed into the BERT word frequency feature, and then entered into the Encoder-Decoder model based on Attention. The PGN calculates a generation probability for each decoder time step to determine whether to generate words from the knowledge vocabulary or copy words from the question sentence.

In the encoder and decoder model, the encoder integrates the input of a complete sentence into a fixed-dimensional vector, and then inputs the vector to the decoder, and the decoder predicts the output based on the vector. However, when the input sentence is long, the fixed-dimensional vector cannot store all the information. In response to this problem, Bahdanau et al. [44] proposed an attention mechanism model in 2015. The attention mechanism allows the decoder to view the words or fragments of the input sentence in the encoder at any time, thus solving the problem of using intermediate vectors to store information. In order to better understand the semantic information of the question entered by the user, it is to ensure that it will not forget the previous key information in the longer text. In this paper, the encoder part of the model uses the BiLSTM model, which captures the long-distance dependence and location information of the source text, so it can better understand the user's intention to enter the question. The LSTM model is used in the decoding part of the model. The detailed description of the model is as follows.

After the question is spliced by BERT and word frequency semantic features, a new input sequence is generated, and the input sequence is input to the BiLSTM encoder. Then the hidden layer state h_i is generated after a single-layer BiLSTM. At the time t , the LSTM decoder receives the previous generated word vectors, getting the decoding state sequence s_t . Then the state of the encoder and the decoder are calculated to obtain the attention distribution a^t , so as to determine the characters that need attention at this time step. The formula is as follows:

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + b_{attn}) \quad (17)$$

$$a^t = \text{softmax}(e^t) \quad (18)$$

v^T refers to the coefficient matrix of attention mechanism, and W_h and W_s represent the coefficient parameters obtained through training. The important context vector h_t^* is obtained by summing the attention weight and the h_i weight, the formula is as follows:

$$h_t^* = \sum_i a_i^t h_i \quad (19)$$

The context vector is the question information read from the encoder at the current time step.

When the model is used to generate words, the words generated by the model are generated in the knowledge vocabulary. The probability distribution is concatenated and spliced by the decoding state sequence s_t and the context vector h_t^* . After passing through two fully connected layers, the current predicted word list distribution P_{vocab} is generated. The formula is as follows:

$$P_{vocab} = \text{softmax}(V'(V[s_t, h_t^*] + b) + b') \quad (20)$$

$$P(w) = P_{vocab}(w) \quad (21)$$

In the formula, V' , V , b , b' are the parameters obtained through learning, and $P(w)$ represents the probability of the generated word at the current moment being the word w in the knowledge vocabulary.

When the model is used to copy words, the probability of pointing to words in the input sequence w is determined according to the attention distribution a^t at time t . The formula is as follows:

$$P_a(w) = \sum_{i:w_i=w} a_i^t \quad (22)$$

The model uses the generation probability P_{gen} to determine whether to copy words from the question sentence or generate words from the knowledge vocabulary. The formula is as follows:

$$P_{gen} = \sigma(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr}) \quad (23)$$

The vectors w_{h^*} , w_s , w_x and scalar b_{ptr} are the parameters obtained through training, and σ is the sigmoid function.

Finally, the weighted average of the vocabulary distribution and the attention distribution by P_{gen} is used to obtain the final probability distribution of the generated word w . The formula is as follows:

$$P(w) = P_{gen}P_{vocab}(w) + (1 - P_{gen})P_a(w) \quad (24)$$

It can be seen from the above formula that when the word w does not appear in the knowledge vocabulary $P_{vocab}(w) = 0$, and when the word w does not appear in the question sentence $P_a(w) = 0$.

4. Experiments

4.1. Datasets

The dataset used in this experiment is Baidu's open source WebQA dataset. In the WebQA (<http://idl.baidu.com/WebQA.html>, accessed on 21 March 2021) dataset, each sample is composed of source text, questions, and answers, among which the questions are based on the source text. In the WebQA dataset, there are 36,145 items in the training set, 3024 items in the test set, and 3018 items in the validation set. The questions in this dataset are provided by "Baidu Know".

When constructing the vocabulary of knowledge, the knowledge base used in this paper is CN-DBPedia (<http://openkg.cn/dataset/cndbpedia>, accessed on 21 March 2021). It is a large-scale general-purpose domain structured encyclopedia developed and maintained by the Fudan University Knowledge Works Research Laboratory. It contains 9 million encyclopedia entities and 67 million triple relationships. Among them, mention2entity information is 1.1 million, summary information is 4 million, label information is 19.98 million, and infobox information is 41 million.

4.2. Evaluation Metrics

In order to better evaluate the effect of the model, the standard evaluation methods precision(P), recall(R), and F1-score(F1) are used in the entity recognition module. Accuracy is used in the answer generation module, which are as follows:

$$P = \frac{TP}{TP + FP} \times 100\% \quad (25)$$

$$R = \frac{TP}{TP + FN} \times 100\% \quad (26)$$

$$F1 = \frac{2PR}{P + R} \times 100\% \quad (27)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (28)$$

4.3. Experimental Environment

The experimental environment of this paper is shown in Table 1.

Table 1. Experimental environment.

Project	Environment
Operating system	Ubuntu16.04
GPU	Geforce GTX 1660Ti
Hard disk	200 G
Memory	8 G

4.4. Parameter Setting

Tensorflow is a deep learning framework developed by Google's artificial intelligence team and is widely used to implement various machine learning algorithms, which is used in the paper to build the model.

In the entity recognition model, the experimental parameter settings in this paper are shown in Table 2.

Table 2. Hyper parameter settings for entity recognition experiments.

Parameter	Value
Epochs	30
Batch_size	128
dropout	0.5
Learning_rate	0.001
embed	300
hidden	200

In the answer generation model, the Bert_base version is used to obtain the question vector in the experiment. It has a total of 12 layers of transformers, 768 hidden units, and 12 attention-heads. The parameter sets the maximum sequence length to 128 and batch_size to 16. The batch_size set by the PGN model is 16, and the size of the custom knowledge vocabulary is 50,000, arranged in reverse order of word frequency to ensure that the most common words appear in the vocabulary. The experimental parameter settings used by the model are shown in Table 3.

Table 3. Hyper parameter settings for answer generation experiments.

Parameter	Value
BERT_batch_size	16
BERT_Learning_rate	0.001
BERT_dropout	0.5
PGN_batch_size	16
PGN_hidden	256
Optimistic algorithm	Adagrad

4.5. Experimental Results and Analysis

4.5.1. Entity Recognition Module

The BiLSTM-CRF named entity recognition model is tested in the source text of the WebQA dataset, and the evaluation results are shown in Table 4.

Table 4. Bi-LSTM-CRF named entity recognition model results.

P%	R%	F1%
89.91	87.85	88.87

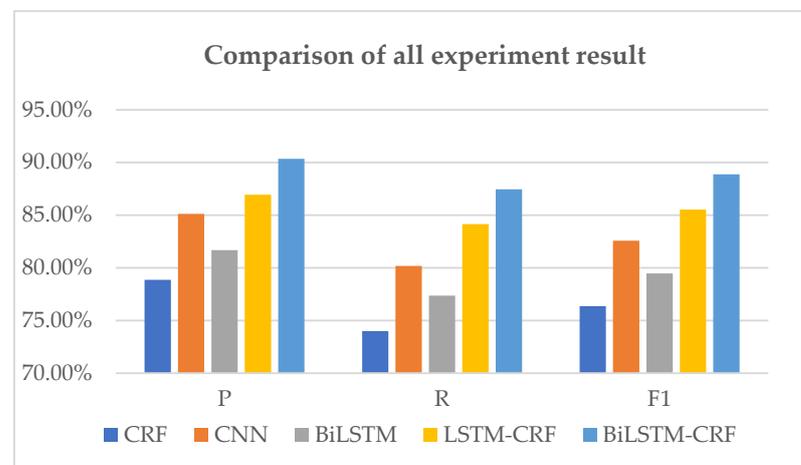
In order to verify the recognition effect of the BiLSTM-CRF model used in this article, it is compared with the recognition results of CRF, CNN, BiLSTM, and LSTM-CRF, as shown in Table 5.

Table 5. Comparison of experimental results of multiple models.

Method	F1%
CRF	76.35
CNN	82.57
BiLSTM	79.46
LSTM-CRF	85.52
Our approach	88.87

It can be seen from Table 5 that the recognition effect of our model is better than other models. 1. BiLSTM-CRF can provide better sequence characteristics, so it is better than using a single CRF. 2. Although CNN can extract features effectively, it extracts static features, so for dynamic sequences, using BiLSTM-CRF can more comprehensively obtain the contextual semantic information of the question answering text.

It can be clearly seen from the model comparison histogram in Figure 5 that the results of BiLSTM-CRF are better than other models.

**Figure 5.** Entity recognition model comparison results.

4.5.2. Answer Generation Module

At present, there are few research works on generative based on knowledge graph, and there is no unified standard for the evaluation of generating a completed sentence containing the answer. In this regard, researchers usually propose standards based on the research question. Therefore, in this experiment, whether the specified entity of the knowledge graph can be included in the generated answer, the evaluation standard adopted is the coverage of the entity in the knowledge graph in the output answer.

In order to prove the effectiveness of our method, the (question, answer) in the CN-DBpedia knowledge base and the WebQA dataset in the following models for comparative experiments are used. The results are shown in Table 6.

Table 6. Comparison results with other models.

Method	Accuracy%
Xie et al.	76.83
Liu et al.	78.26
Luo et al.	78.79
Our approach	79.62

- Xie et al. [45]: The model proposed by Xie et al. uses TEEM for entity recognition, and then applies a deep structured semantic model to calculate the semantic similarity of the question and predicate in the candidate knowledge triples.
- Liu et al. [46]: Liu et al. proposed a method based on entity sorting and joint fact selection. This method uses BiLSTM-CRF for pattern extraction, and then uses similarity matching to sort the candidate entities, and finally uses the joint fact selection model to enhance the selection of the correct entity relationship pair using multi-level coding.
- Luo et al. [47]: Luo et al. proposed a relationship detection model based on a multi-angle attention mechanism, which uses the attention mechanism to extract the correlation between question patterns and candidate relationships from word level and relationship level.

Since there are currently fewer extraction-generative methods based on knowledge graphs, a comparative model of extraction methods is chosen for experiments. It can be seen from Table 6 that our method has achieved good results. After research and analysis, it can be found that the errors in the experiment are mainly caused by the lack of entities in the knowledge vocabulary, the existence of wrong entities, and the words describing relationships in the knowledge base and the answers, which are caused by the recognition effect of the entity recognition module. The impact and the impact of the vocabulary size setting thereby affects the performance of our model.

Table 7 shows examples of answers returned by the above three models and the methods in this paper. It can be seen from the sample that the CN-DBpedia knowledge base corresponds to the entity “Huo Xiaolin” in the question sentence with (Huo Xiaolin, father, Huo Shaochang), (Huo Xiaolin, starring, Yang Zhigang) and other triples, but there is no entity “Huo Shaochang” corresponds to the triple, so the answer returned by the extractive model is wrong. However, the method adopted in this paper first reads the entity in the original text to ensure that the entity is stored in the knowledge vocabulary. Therefore, the above experiments show that the method proposed in this paper can effectively return the correct answer when the knowledge base is not fully stored, and generate a humanized complete sentence answer while ensuring the accuracy of the answer.

Table 7. Example of the answer returned by the experimental model.

Question	Method	Answer
Chinese: 勇敢的心霍啸林的父亲是谁出演的?	Xie et al.	杨志刚 (Yang Zhigang)
	Liu et al.	霍绍昌 (Huo Shaochang)
	Luo et al.	霍绍昌 (Huo Shaochang)
English: Brave Heart Who starred in Huo Xiaolin’s father?	Our approach	寇振海是霍啸林的父亲 (Kou Zhenhai is Huo Xiaolin’s father.)

Based on the above analysis, this paper considers the impact of the size of the knowledge vocabulary on the answer generation. Therefore, three vocabulary sizes are used for experiments. In the comparison model, the same triples corresponding to the knowledge vocabulary are used. As shown in Figure 6, the accuracy of the extractive model is improving with the increase of knowledge base data. However, the main reason that the accuracy

of the method proposed in this paper decreases after the word expression reaches a certain level is that after the scale of the vocabulary expands, more useless nouns increase, which will also lead to inaccurate answers.

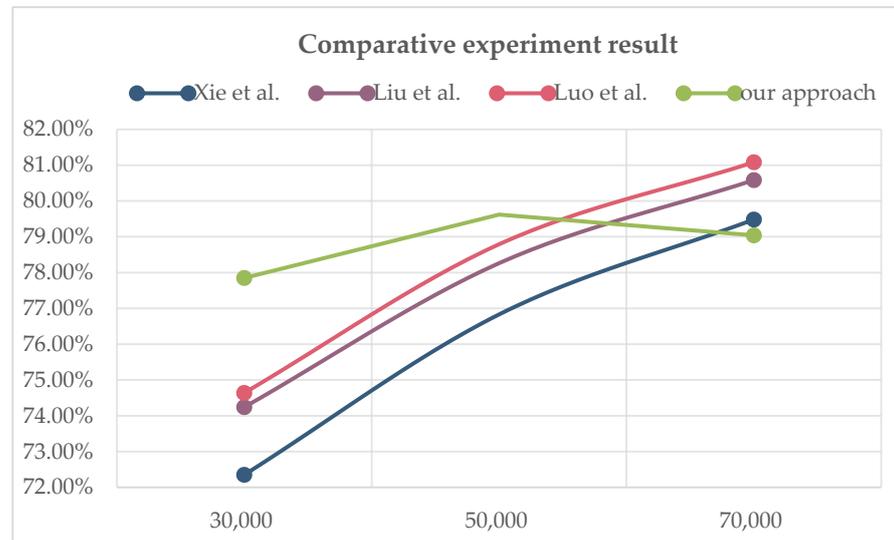


Figure 6. Experiment on the influence of knowledge vocabulary on answers.

In the process of realizing answer generation, the BERT-word frequency semantic feature + pointer generation network model is used as the answer generator. In the word vectors acquisition stage, the semantic features of BERT and word frequency are spliced as the input sequence of the answer generator. In addition, in the word vector acquisition phase, the influence of using Term Frequency-Inverse Document Frequency (TF-IDF) [48] on the output is considered. In order to verify the effectiveness of our method, a comparative experiment was carried out. The results are shown in Table 8.

Table 8. Comparison results with other models.

Method	Accuracy%
PGN	76.43
BERT-PGN	78.37
Our approach	79.62
Our approach(use TF-IDF)	79.94

According to Table 8, the accuracy of using only PGN is 3.19% lower than that of using the BERT pre-trained language model to splice word frequency semantic features, and our method is 1.25% higher than the method using no word frequency semantic features. Experiments show that our method can further improve the effect of the model. The word vectors representation obtained by the pre-trained language model and word frequency semantic features can more accurately understand the semantic information of the question, and then improve the accuracy of the generated answer. Then our method is compared with a further improved method using TF-IDF. It can be found that the accuracy rate was increased by 0.32% compared to the previous one. Combining the pre-trained language model with TF-IDF, our method can express the importance of the entities of the question in the dataset more clearly.

In order to facilitate viewing, the experimental results of each part of the model as a histogram are compared, as shown in Figure 7.

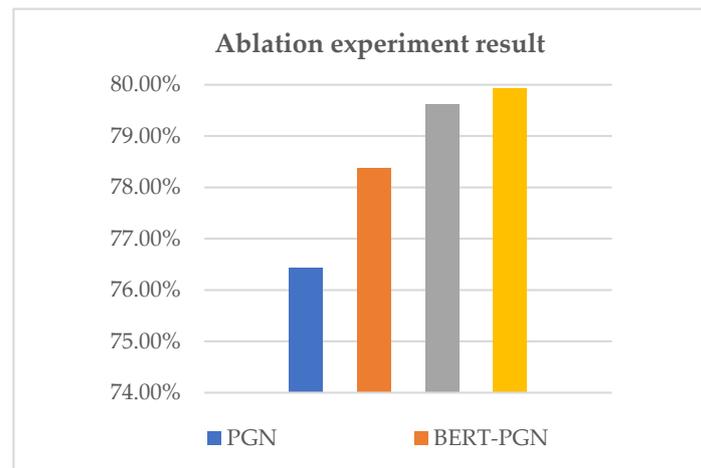


Figure 7. Comparison results with other models.

5. Conclusions

This paper innovatively proposes a question-and-answer method that combines a generative knowledge graph. First, it introduces how to construct a knowledge vocabulary for generating words, and then gives a method for obtaining the BERT-word frequency semantic feature of word vectors, and proposes using the knowledge graph as a pointer to generate a soft link of the network to generate answers.

In future work, other feature fusion will be considered to improve the understanding of question semantics, and add BERT to the entity recognition module to improve the accuracy of identifying entities. In order to ensure the quality of the expanded knowledge vocabulary, deep learning methods will be considered to further extract the entities and relationships of the original part of the dataset. Next, more datasets will be chosen for experiments to verify the performance of this method. Furthermore, knowledge graph technology will be combined with a variety of generative models to generate answers, and further improve the effect of generating answers.

Author Contributions: S.L. wrote the paper and supervised the work; N.T. performed the experiments; Y.G. conceived and designed the experiments; N.L. gave some suggestions on the experiments. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Social Research Project on Economic Development in Liaoning Province (grant no. lslybkt-022), Natural Science Foundation of Liaoning Province, China (grant no. 20180550921 and 2019-ZD-0175) and Scientific Research Fund Project of the Education Department of Liaoning Province (LJYT201906).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All datasets publicly available is used in this paper. We have provided links in the paper.

Acknowledgments: The authors would like to thank all anonymous reviewers and editors for their helpful suggestions for the improvement of this paper.

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

References

1. Turing, A.M. Computing Machinery and Intelligence. *Mind* **1950**, *59*, 433–460. [[CrossRef](#)]
2. Abramson, D. Turing's Responses to Two Objections. *Minds Mach.* **2008**, *18*, 147–167. [[CrossRef](#)]
3. Fedushko, S.; Ustyianovych, T.; Gregus, M. Real-time high-load infrastructure transaction status output prediction using operational intelligence and big data technologies. *Electronics* **2020**, *9*, 668. [[CrossRef](#)]

4. Amit, S. Introducing the Knowledge Graph: Things, Not Strings. Available online: <https://blog.google/products/search/introducing-knowledge-graph-things-not/> (accessed on 19 March 2021).
5. Biega, J.; Kuzey, E.; Suchanek, F.M. Inside YAGO2s: A transparent information extraction architecture. In Proceedings of the 22nd International Conference on World Wide Web Companion, Rio de Janeiro, Brazil, 13–17 May 2013; International World Wide Web Conferences Steering Committee: Rio de Janeiro, Brazil, 2013; pp. 325–328.
6. Erxleben, F.; Günther, M.; Krötzsch, M.; Mendez, J.; Vrandečić, D. Introducing wikidata to the linked data web. In Proceedings of the 13th International Semantic Web Conference, Riva del Garda, Italy, 19–23 October 2014.
7. Bizer, C.; Lehmann, J.; Kobilarov, G.; Auer, S.; Becker, C.; Cyganiak, R.; Hellmann, S. DBpedia-A crystallization point for the Web of data. *J. Web Semant.* **2009**, *7*, 154–165. [[CrossRef](#)]
8. Wang, Z.; Li, J.; Wang, Z.; Li, S.; Li, M.; Zhang, D.; Shi, Y.; Liu, Y.; Zhang, P.; Tang, J. XLORE: A Large-scale English-Chinese Bilingual Knowledge Graph. In Proceedings of the Meeting of the International Semantic Web Conference (Posters & Demos), Sydney, Australia, 21–25 October 2013.
9. Niu, X.; Sun, X.; Wang, H.; Rong, S.; Qi, G.; Yu, Y. Zhishi me-weaving chinese linking open data. In Proceedings of the Semantic Web-ISWC 2011, Bonn, Germany, 23–27 October 2011; Springer: Berlin, Germany, 2011; pp. 205–220.
10. MrYener. OwnThink Knowledge Graph. Available online: <https://www.ownthink.com/> (accessed on 17 March 2021).
11. Rajarshi, D.; Manzil, Z.; Siva, R.; Andrew, M. Question Answering on Knowledge Bases and Text using Universal Schema and Memory Networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 358–365.
12. Sun, H.; Bhuwan, D.; Manzil, Z.; Kathryn, M.; Ruslan, S.; Cohen, W. Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 30 July–4 August 2018; pp. 4231–4242.
13. Wang, J.; Liu, J.; Bi, W.; Liu, X. Improving Knowledge-aware Dialogue Generation via Knowledge Base Question Answering. *arXiv* **2019**, arXiv:1912.07491. [[CrossRef](#)]
14. Xiong, W.; Yu, M.; Chang, S.; Guo, X.; William, Y. Improving Question Answering over Incomplete KBs with Knowledge-Aware Reader. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 4258–4264.
15. Wei, M.; He, Y.; Zhang, Q.; Si, L. Multi-Instance Learning for End-to-End Knowledge Base Question Answering. *arXiv* **2019**, arXiv:1903.02652.
16. See, A.; Liu, P.J.; Manning, C.D. Get to the Point: Summarization with Pointer-Generator Networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1073–1083.
17. Shen, Y.; He, X.; Gao, J.; Deng, L.; Grégoire, M. A latent semantic model with convolutional-pooling structure for information retrieval. In Proceedings of the ACM Conference on Information and Knowledge Management, Shanghai, China, 19–23 November 2011; pp. 101–110.
18. Hu, S.; Zou, L.; Yu, J.; Wang, H.; Zhao, D. Answering natural language questions by subgraph matching over knowledge graphs. *IEEE Trans. Knowl. Data Eng.* **2017**, *30*, 824–837. [[CrossRef](#)]
19. Xu, K.; Feng, Y.; Huang, S.; Zhao, D. Hybrid question answering over knowledge base and free text. In Proceedings of the 26th International Conference on Computational Linguistics, Osaka, Japan, 11–16 December 2016; pp. 2397–2407.
20. Bordes, A.; Chopra, S.; Jason, W. Question Answering with Subgraph Embeddings. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 615–620.
21. Hao, Y.; Zhang, Y.; Liu, K.; He, S.; Wu, H.; Zhao, J. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 221–231.
22. Bordes, A.; Usunier, N.; Chopra, S.; Weston, J. Large-scale simple question answering with memory networks. *arXiv* **2015**, arXiv:1506.02075.
23. 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.
24. Sorokin, D.; Gurevych, I. Modeling Semantics with Gated Graph Neural Networks for Knowledge Base Question Answering. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 13–18 December 2018; pp. 3306–3317.
25. Denis, L.; Asja, F.; Jens, L.; Sören, A. Neural Network-based Question Answering over Knowledge Graphs on Word and Character Level. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 1211–1220.
26. Wang, Y.; Zhang, R.; Xu, C.; Mao, Y. The APVA-TURBO Approach To Question Answering in Knowledge Base. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 13–18 December 2018; pp. 1998–2009.
27. Ilya, S.; Oriol, V.; Quoc, V. Sequence to Sequence Learning with Neural Networks. *arXiv* **2017**, arXiv:1409.3215v3.
28. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
29. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.

30. Li, P.; Li, W.; He, Z.; Wang, X.; Cao, Y.; Zhou, J.; Xu, W. Dataset and Neural Recurrent Sequence Labeling Model for Open-Domain Factoid Question Answering. *arXiv* **2016**, arXiv:1607.06275.
31. 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; Springer: Cham, Switzerland, 2017; pp. 428–438.
32. Khodra, M.L.; Wahyudi; Prihatmanto, A.S.; Machbub, C. Knowledge-Based Graph Compression Using Graph Property on Yago. In Proceedings of the 3rd International Conference on Science in Information Technology (ICSITech), Bandung, Indonesia, 25–26 October 2017; pp. 136–141.
33. Sun, J. Jieba. Available online: <https://github.com/foxsjy/jieba> (accessed on 17 March 2021).
34. 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, San Diego, CA, USA, 2–7 June 2016; pp. 260–270.
35. Lafferty, J.; McCallum, A.; Pereira, F. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning, San Francisco, CA, USA, 28 June–1 July 2001; pp. 282–289.
36. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
37. Zaremba, W.; Sutskever, I.; Vinyals, O. Recurrent Neural Network Regularization. *arXiv* **2014**, arXiv:1409.2329.
38. Fu, Z.; Lam, W.; So, A.; Shi, B. A Theoretical Analysis of the Repetition Problem in Text Generation. *arXiv* **2020**, arXiv:2012.14660.
39. Sean, W.; Ilia, K.; Jaedeok, K.; Richard, Y.; Kyunghyun, C. Consistency of a Recurrent Language Model with Respect to Incomplete Decoding. *arXiv* **2020**, arXiv:2002.02492.
40. Rong, X. word2vec Parameter Learning Explained. *arXiv* **2014**, arXiv:1411.2738.
41. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.
42. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. OpenAI, 2018. Available online: <https://www.cs.ubc.ca/~jamuham01/LING530/papers/radford2018improving.pdf> (accessed on 19 March 2021).
43. Vinyals, Q.; Fortunato, M.; Jaitly, N. Pointer Networks. *arXiv* **2015**, arXiv:1506.03134.
44. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.
45. 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: Cham, Switzerland, 2016; pp. 300–311.
46. Liu, Y.; Zhang, L.; Yang, Y.; Zhang, G.; Zhang, C. Simple question answering with entity ranking and joint fact selection. *Appl. Res. Comput.* **2020**, *37*, 3321–3325.
47. Luo, D.; Su, J.; Li, P. Multi-view Attentional Approach to Single-fact Knowledge-based Question Answering. *Comput. Sci.* **2019**, *46*, 215–221.
48. Xu, T.; Wu, M. An Improved Naive Bayes Algorithm Based on TF-IDF. *Comput. Technol. Dev.* **2020**, *30*, 75–79.