


Article

Incorporating Phrases in Latent Query Reformulation for Multi-Hop Question Answering

Jiuyang Tang, Shengze Hu, Ziyang Chen, Hao Xu and Zhen Tan * 

Science and Technology on Information Systems Engineering Laboratory,
National University of Defense Technology, Changsha 410073, China; jiuyang_tang@nudt.edu.cn (J.T.);
springsun@nudt.edu.cn (S.H.); chenziyangnudt@nudt.edu.cn (Z.C.); xuhao@nudt.edu.cn (H.X.)

* Correspondence: tanzhen08a@nudt.edu.cn

Abstract: In multi-hop question answering (MH-QA), the machine needs to infer the answer to a given question from multiple documents. Existing models usually apply entities as basic units in the reasoning path. Then they use relevant entities (in the same sentence or document) to expand the path and update the information of these entities to finish the QA. The process might add an entity irrelevant to the answer to the graph and then lead to incorrect predictions. It is further observed that state-of-the-art methods are susceptible to reasoning chains that pivot on compound entities. To make up the deficiency, we present a viable solution, i.e., incorporate phrases in the latent query reformulation method (IP-LQR), which incorporates phrases in the latent query reformulation to improve the cognitive ability of the proposed method for multi-hop question answering. Specifically, IP-LQR utilizes information from relevant contexts to reformulate the question in the semantic space. Then the updated query representations interact with contexts within which the answer is hidden. We also design a semantic-augmented fusion method based on the phrase graph, which is then used to propagate the information. IP-LQR is empirically evaluated on a popular MH-QA benchmark, HotpotQA, and the results of IP-LQR consistently outperform those of the state of the art, verifying its superiority. In summary, by incorporating phrases in the latent query reformulation and employing semantic-augmented embedding fusion, our proposed model can lead to better performance on MH-QA.

Keywords: multiple documents; multi-hop question answering; phrase incorporation; latent query reformulation



Citation: Tang, J.; Hu, S.; Chen, Z.; Xu, H.; Tan, Z. Incorporating Phrases in Latent Query Reformulation for Multi-Hop Question Answering. *Mathematics* **2022**, *10*, 646. <https://doi.org/10.3390/math10040646>

Academic Editors: Manuel Vilares-Ferro, Pavel Brazdil and Gaël Dias

Received: 11 January 2022

Accepted: 16 February 2022

Published: 19 February 2022

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



Copyright: © 2022 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

Question answering (QA), or reading comprehension, has long been the holy grail of machine intelligence. In pursuit of it, efforts are dedicated to enabling machines to automatically retrieve information and perform inference over texts, just like the human cognitive reasoning process. Pioneering research focuses on single-document QA, using benchmarks including CNN&Dailymail [1], SQuAD [2] and RACE [3]. The progressive development of human cognition and reasoning follows a rule of “creep-before-you-walk”. We begin with some straightforward questions which can be simply answered by referring to a single document. Then, we proceed to more challenging questions which might need to be inferred from different information sources. The same applies for the machine. Starting from single-document QA, it enhances its reasoning ability step by step and eventually addresses multiple-document problems. There are some works exploring multiple-document scenarios [4], but they basically still fall into the single-document QA category, as only one document contains the supporting evidence to answer the question, while others are noises to confuse the model. To the best of our knowledge, QAngaroo [5] is among the first to concentrate on predicting answers from several documents. However, questions in QAngaroo are in the form of knowledge triplets, and the task fails to capture the explicit

reasoning path of the documents since the metric does not contain the part about the reasoning path.

During the growth of the logic inference ability of people, people only need one document for a single question in senior schools, while many references are required in junior schools. Inspired by the development, machines are also supposed to enhance cognitive reasoning ability to address multiple-documents problems. The latest effort was focused on multi-hop QA (MH-QA) within multiple documents [6–9], among which HotpotQA is one of the representative works. In HotpotQA, questions are in the natural language form, and answers need to be inferred from several pieces of similar texts without candidate answers. In comparison with previous tasks, the aim of MH-QA is to infer answers by exploiting the interaction between relevant documents and the given question.

MH-QA is difficult due to the fact that there exists a reasoning chain hidden behind the QA process, through which multiple documents are connected and interact. The challenges lie in the need to explicitly sort out a reasoning path through multiple content-related and logically connected documents. Previous works tried to put several subtasks, such as candidate sentence selection, answer prediction, and answer type prediction, under a unified multi-task learning framework [6,8], aiming to capture the interactions among various parts of several texts. To handle the unique challenge, graph neural network (GNNs) methods have been widely explored to understand the inter-relations among texts [7,9]. Specifically, they construct an interaction graph by extracting entities in text as nodes and treating all possible connections between those nodes as relations. GNN is then incorporated to learn the intrinsic interactions among entities.

Nevertheless, the current methods tend to fall short in the following aspects. First, the construction of an interaction graph is highly reliant on the entities extracted from questions and candidate texts, but the entities are too fuzzy to express a concrete meaning in most cases. Hence, it may not be easy for the reasoning process to figure out which entity to follow successively in order to arrive at the correct answer. For example (as depicted in Figure 1), when “Soviet statesman” is used as the reasoning starting point, there are three viable subsequent entities to follow, namely, “Mikhail Gorbachev”, “Nikolai Viktorovich Podgorny” and “Andrei Pavlovich Kirilenko”. It can be non-trivial to choose from the three candidates to ensure that the reasoning process will eventually lead to the right answer. In contrast, if we choose “former Soviet statesman” as the starting point, it will be much easier to locate the right subsequent entity “Mikhail Gorbachev” to update the query. Therefore, we hypothesize that the phrase, rather than entities, is better at fulfilling the task of connecting reasoning chains.

Question: What type of forum did a former Soviet statesman initiate?

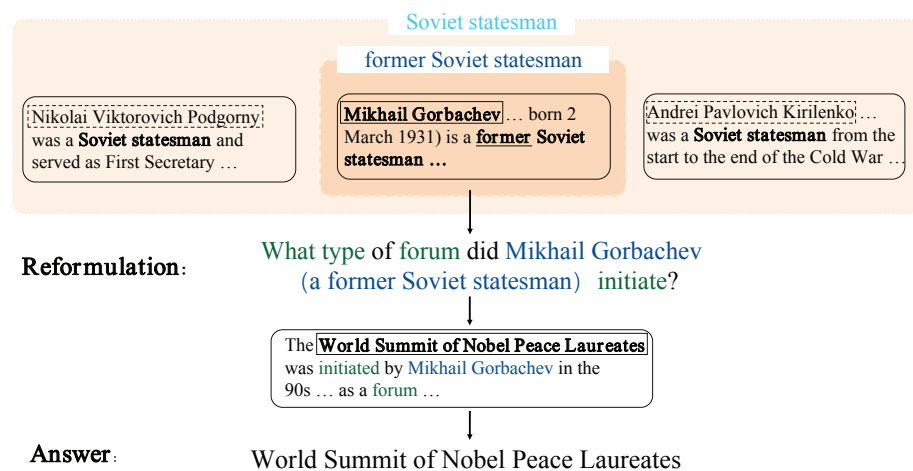


Figure 1. Examples of challenges in HotpotQA.

Second, combining all possible information from the texts results in an information overload, which is likely to overwhelm the model. Statistics show that there can be as many as 111 entities in a context in HotpotQA (based on our data processing). If they are all fed into a GNN, irrelevant information inevitably becomes involved, which undermines the performance. For instance, besides “Mikhail Gorbachev”, the existence of the other irrelevant entities “Nikolai Viktorovich Podgorny” and “Andrei Pavlovich Kirilenko” inevitably introduces noise. In this connection, it is beneficial to clarify the corresponding part of the question, using “Mikhail Gorbachev” in the first place. Therefore, it can be beneficial to refine the contextual information for fusion before conducting MH-QA.

To investigate the ideas, we implement them in a novel framework inspired by the classic query reformulation technique in information retrieval. Specifically, we propose to incorporate phrases to latently reformulate the questions such that the answer prediction can better perform. For the first challenge, we employ phrases as the basic units of our model to construct a reasoning chain in which information from different documents is transferred. To obtain quality phrases, we first filter out relevant documents and then extract phrases under the guidance of certain grammar rules. To tackle the second challenge, we introduce a phrase-level latent query reformulation method by using semantic information to latently update the question representations, without changing the original literal texts. It is realized by (1) constructing a similarity graph, in which phrases are treated as nodes, and pair-wise similarity as edges; and (2) learning a semantic-augmenting model to update the representations of phrases and hence, latently reformulating the question. The process helps our model to accurately locate the significant words, which avoids information overload.

To summarize, the key contributions of the paper is at least two-fold:

- We propose to incorporate phrases in the latent query reformulation method (IP-LQR) for the multi-hop QA problem. IP-LQR utilizes information from relevant contexts to reformulate the question in the semantic space. Then the updated query representations interact with contexts within which the answer hides.
- We design a semantic-augmented fusion method based on the phrase graph. Phrases in the question are regarded as central nodes in the graph, and then phrases from relevant contexts are connected with them based on both literal and semantic relevance. The graph is finally used to propagate the information.

2. Related Work

The related work can be divided into three categories: pseudo-multi-document QA, MH-QA and dialogue QA.

Pseudo-multi-document QA. The research about pseudo-multi-document QA was pioneered by the information retrieval community and dates back to the 1980s. START [10] is the first work in this field. It uses two databases to retrieve the query by default. If the question is not covered by them, START executes a matching process between words from questions and documents from the external knowledge base, and then returns relevant results. The following works focused on expanding the search space or improving retrieval efficiency, such as [11]. However, these works are more like designing search engines rather than QA systems, as they all return the whole document that might contain the answer, instead of directly giving the answer. Recently, DrQA [4] first introduced the neural networks into a pseudo-multi-document QA. It utilizes questions to retrieve the top 5 relevant paragraphs based on TF-IDF from Wikipedia and uses a QA model to extract the answer span from these paragraphs. Then, a rerank component is added to generate the final answer. The retrieve-read framework significantly outperforms the former methods in the pseudo-multi-document question and has quickly drawn wide attention. There are numerous studies that apply a similar method to construct their models and try to improve the performance in QA [12,13] or retrieve components [14–16]. However, the retrieve-read framework cannot handle MH-QA since in many situations, golden contexts containing the answer are not readily retrievable with the question. Moreover, the framework uses the

QA model on each paragraph separately, which fails to build the context-level interactions among documents.

MH-QA. To address the MH-QA problem, many works focus on applying GNN to represent the multiple relations between different documents. Ref. [9] proposes a knowledge-enhanced graph neural network (KGNN) to build the connections of different entities. They firstly extract entities from documents as nodes to build an entity graph. Then, if there is a relation between two entities, they will set an edge between them in the corresponding place of an entity graph. However, it excessively depends on the knowledge graph. Once the knowledge graph is incomplete, the edges are missing as well. Ref. [7] introduces a dynamically fused graph network (DFGN), concentrating on the dynamic update of graphs and documents. Similar to the former work, an entity graph is constructed firstly and a graph attention method is utilized to achieve information aggregation. After that, they update the representation of original sentences, questions, and documents based on the entity graph. Though the dynamic fusion layer has good performance, DFGN ignores some important information from explicitly irrelevant but implicitly relevant entities. Besides the graph-based method, some other works explore MH-QA from different perspectives. Ref. [8] proposed a multi-task learning framework containing prediction, extraction, and explainability. The authors combined cascade models from the pipeline to strengthen the connections of different parts and obtained the final result, step by step. Ref. [6] utilized a query-focused extractor (QFE) to capture the relevance between questions and documents. Inspired by the text summarization model, they extract the summary of documents with the participant of questions to ensure the coverage of target information. Intuitively, these models cannot build the relations between nodes well because they ignore the deep interactions between information. The above methods select entities as primary units, which might increase ambiguity in the information aggregation process. Moreover, in order to add explainability, they all select a single sentence as evidence of QA to obtain the answer, which means that the information of the remaining document is abandoned directly. To overcome these shortcomings, IP-LQR is developed in our research and proven to obtain promising performance.

Multi-hop KBQA. Multi-hop knowledge-based question answering (KBQA) is the complementary line of the research relevant to the complicated question answering. The task's novelty lies in the supporting corpus (i.e., knowledge graph) of the question. Existing methods mainly fall into two approaches: information retrieval (IR), which retrieves answers from KG by learning representations of question and graph; and semantic parsing (SP), which queries answers by parsing the question into a logical form. In the IR-based category, Ref. [17] applied manually defined and extracted features to capture the relevance between questions and KG, which are time consuming and cannot efficiently exploit question semantics. Regarding these issues, the following methods convert questions and related entities into representations and treat KGQA tasks as semantic matching between embeddings of questions and candidate answers [18–20]. Typically, EmbedKGQA relaxes the requirement of answer selection from a pre-specified local neighborhood and first introduces KG embeddings into the task. After that, the similarity score is calculated as the clue to determine the final answer [21]. SP-based approaches follow a parse-then-execute procedure. These methods [22–24] can be briefly summarized into the following steps: (1) parsing relations and subjects involved in complex questions; (2) decoding the subgraph into an executable logic form, such as high-level programming languages, or structured queries, such as SPARQL; and (3) searching from KGs using the query language and providing query results as the final answer.

Dialogue QA. There is another branch of MH-QA, called dialogue QA [25]. One big challenge of this task is effectively exploiting the conversation history. Choi et al. [26] proposed a feature vector embedding and an answer vector embedding to respectively store the previous question features and answers. However, these methods ignore the previous reasoning history performed by the model when reasoning at the current turn. Huang et al. [27] introduced the idea of integration flow (IF) to allow rich information

in the reasoning process to flow through a conversation. Another challenge of dialogue QA is generating abstractive answers. Reddy et al. [28] combined an extractive reading comprehension model and a text generator to tackle the problem. Yatskar [29] designed a model which firstly makes a three classification prediction (yes/no/span) and outputs an answer span only if Yes/No is not selected. On top of that, there are also many studies concentrating on the conversational sentiment analysis area [30] and dialogue systems with commonsense [31] or audio contexts [32]. Though these branches also lay on the multi-document, the goal is to generate conversational situations between humans and machines. Thus, we do not further discuss these topics in our paper.

3. Materials and Methods

This section introduces the framework of our method, namely, IP-LQR, which fuses high-quality phrases via a similarity graph and then reformulates the question representation for answer prediction. We first formally define the task and introduce the overall framework, the relevant graph construction in Section 3.2 and latent reformulation in Section 3.3.

3.1. Task Definition and Framework

Given a collection of paragraphs, the task of multi-hop question answering is to provide an answer to a question and also justify these answers with supporting facts (which are the facts that are necessary for reasoning to derive the answer). We refer to the sentences containing supporting facts as supporting sentences. The HotpotQA is under the distractor setting, which is also followed in this paper, and the number of paragraphs is limited to 10. The performance of such systems is measured by F1 scores of predicting answers (denoted as “Ans”) and supporting facts (denoted as “Sup”), as well as jointly (denoted as “Joint”).

An illustration of the framework is given in Figure 2. Our method starts with a context selector to filter irrelevant contexts with the question. Then, selected contexts and the question are put into the graph constructor to build a phrase graph. Based on the phrase graph, our model evaluates the similarities between phrases and the question to construct the input of the fusion block. After fusing, the final results are generated by the model. More details about these modules are discussed below.

3.2. Relevance Graph Construction

Constructing the phrase graph involves two steps: (1) context selection, to identify the context that is truly relevant to the answer; and (2) graph construction, where phrases are extracted and connected with each other. Based on these procedures, more high-quality phrases are captured, which also facilitates the following parts of the model.

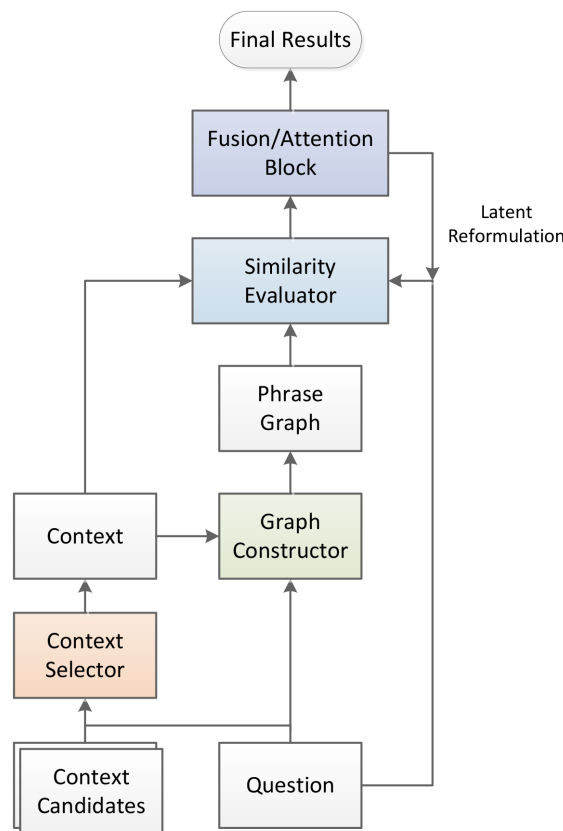


Figure 2. Framework of model IP-LQR.

3.2.1. Context Selection

Since phrases play a pivotal role in our model, it is of great importance to filter the irrelevant context in order to help us locate relevant ones, where quality phrases usually appear. We regard the task as a binary classification problem and design a selection network to achieve it.

The encoding layer of the network is based on pre-trained BERT [33]. Denoting the j -th paragraph as “context $_j$ ”, we concatenate it with a question to form a latent representation,

$$[Q, C_j] = [\text{CLS}] \text{ question } [\text{SEP}] \text{ context}_j [\text{SEP}], \tag{1}$$

where [CLS] is the class label produced by BERT, Q is for the question, and C_j is for context j . Then, a binary classifier $\mathcal{F}_{\text{relv}}$ is used to calculate a relevance score rs_j between Q and C_j based on the representation \mathbf{E}_{CLS} of [CLS] token, and a sigmoid function follows.

$$rs_j = \text{sigmoid}(\mathcal{F}_{\text{relv}}(\mathbf{E}_{\text{CLS}})), \tag{2}$$

where $rs_j \in [0, 1]$ serves as the evidence to determine the relevance between Q and C_j .

In training, the label of the question–paragraph pair $[Q, C_j]$ (following Equation (1)) is tagged as 1 if context j contains a supporting sentence, and 0 otherwise. The training sample is shown in Figure 3. These annotated samples fine-tune pre-trained BERT to approach the downstream task in the high vector space. In inference, we follow the form above and use fine-tuned BERT to acquire the prediction score of each sample. The top n -ranked paragraphs by relevance scores are chosen as the question-related context of Q .

Question: The Oberoi family is part of a hotel company that has a head office in what city?

Supporting facts: ["Oberoi family", 0], ["The Oberoi Group", 0]

Paragraph 1, Ritz-Carlton Jakarta: The Ritz-Carlton Jakarta is a hotel and skyscraper in Jakarta, Indonesia and 14th Tallest building in Jakarta. It is located in city center of Jakarta, near Mega Kuningan, adjacent to the sister JW Marriott Hotel. It is operated by The Ritz-Carlton Hotel Company.

Paragraph 2, Oberoi family: The Oberoi family is an Indian family that is famous for its involvement in hotels, namely through The Oberoi Group.

...

Paragraph 7, The Oberoi Group: The Oberoi Group is a hotel company with its head office in Delhi. Founded in 1934, the company owns and/or operates 30+ luxury hotels and two river cruise ships in six countries, primarily under its Oberoi Hotels & Resorts and Trident Hotels brands.

...

Answer: Delhi

Figure 3. A sample from HotpotQA training set. Supporting facts formatted as [document_title, sentence_id] are annotated in blue. The paragraph containing supporting facts is labeled 1 (i.e., Paragraphs 2 and 7), while others are 0.

3.2.2. Graph Construction

As there is not a complete external phrase base, we apply the Stanford CoreNLP toolkit [34] to recognize phrases from texts. In order to extract high-quality phrases, we design 3 grammar rules to refine the output: (1) the part of speech (POS) of the last word in the phrase is a noun, as key phrases in MH-QA are usually nominal phrases; (2) the POS of the start word in the phrase cannot be a verb, as phrases in the form of “verb + noun” are verbal phrases; and (3) two sequential phrases in one sentence are combined into a single phrase, as it is common that in domain applications, infrequent phrases are made of some frequently used phrases. Some examples of the refinement are given in Table 1.

Table 1. Examples of refined phrases. “Original phrase” is extracted via the Stanford CoreNLP toolkit. “Refined phrase” denotes the results constrained by designed grammar rules.

Original Phrase	Refined Phrase
Soviet statesman	former Soviet statesman
river, cruise ships	river cruise ships
the founder, chairman	the founder and chairman
a street fashion, clothing company	a street fashion clothing company

Afterward, all phrases are classified into two categories, according to whether or not they appear in the question. The phrases appearing in the question are called core nodes, and the others peripheral nodes. In addition, we supplement the graph with sentences in the question as core nodes and the context paragraphs and sentences thereof as peripheral nodes. For edges in the graph, we connect all peripheral nodes to every core node. In this way, we expect the semantic structure of relevant context with respect to the question can be sufficiently captured.

3.3. Latent Reformulation

In the information retrieval area, the query reformulation applies a set of techniques to (explicitly) rewrite the queries so that better search results can be returned. Inspired by this, we propose to incorporate reformulation into MH-QA and employ information

from relevant parts of texts to augment the question via latent reformulation. The updated question can avoid the model taking irrelevant words into the fusion block and avoid invalid information passing.

The reformulation framework is illustrated in Figure 4. Given a context and the corresponding question, we first encode them into the high vector space and acquire the representations of phrases, sentences, and paragraphs via the mean-pooling layer. Then, a similarity evaluation strategy is designed to calculate the weights of edges in the graph (cf. Section 3.3.1). The fusion layer works as an information aggregation to latently update the original question’s representation (cf. Section 3.3.2). Finally, we propose a re-attention mechanism to help locate the gold answer based on the new representation.

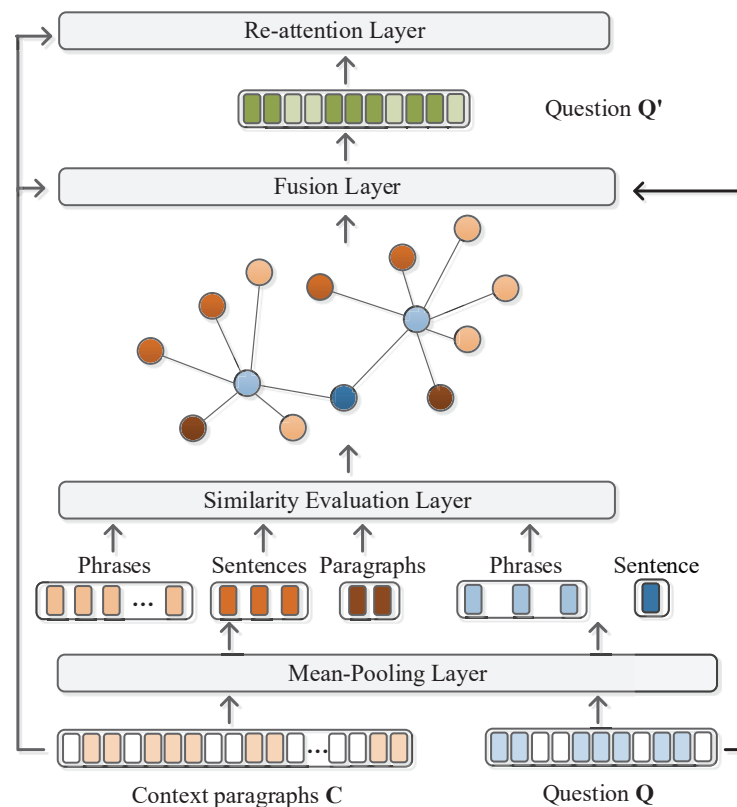


Figure 4. Reasoning process in model IP-LQR.

3.3.1. Similarity Evaluation

In the information retrieval community, studies usually locate the most relevant part in contexts with the question based on the max string similarity. The part is then employed to replace the confusing part in the question, which is named literal reformulation. Intuitively, the process tends to increase the irrelevant information of the question. The problem with literal similarity is that it sometimes introduces false negatives. Take the phrases “town plaza” and “city square” as an example. The two phrases imply the same annotation but have no literal overlapping. Obviously, *literal_sim* underestimates the connection between synonyms. As a remedy, we further incorporate *semantic_sim* over embeddings to make up for the deficiency, which exploits the semantic relevance between phrases. Different from conventional methods, we reformulate the question on its representation, namely latent reformulation, based on *literal_sim* and *semantic_sim*.

As Figure 1 shows, many phrases have word overlaps or even are identical in MH-QA. Thus, we take a phrase “Soviet statesman” from questions as the start of reasoning. Then

we use the character-level Levenshtein distance to measure the connection between two nodes, a and b . That is,

$$\text{literal_sim}(a, b) = 1 - LD(a, b) / |a|, \tag{3}$$

where LD is the Levenshtein distance, and $|\cdot|$ is the string length.

Given that we already have word embeddings by using BERT in context selection, sentence embeddings are simply the average of the embeddings of words in a sentence. The output [CLS] is used to represent the paragraph embedding. Following Equation (1), we can obtain representations \mathbf{E}_a and \mathbf{E}_b of phrases a and b from the representations of the question and the context, respectively. With the embeddings of phrases, an attention network is applied to calculate the semantic similarity. Mathematically,

$$\tilde{\mathbf{E}}_a = \text{Mean-Pooling}(\mathbf{E}_a), \tag{4}$$

$$\beta_{ab} = (\mathbf{W}_a \tilde{\mathbf{E}}_a + \mathbf{B}_a)(\mathbf{W}_b \tilde{\mathbf{E}}_b + \mathbf{B}_b), \tag{5}$$

$$\text{semantic_sim}(a, b) = \text{softmax}(\beta_{ab}), \tag{6}$$

where \mathbf{W}_a and \mathbf{W}_b are learnable linear projection matrices, and \mathbf{B}_a and \mathbf{B}_b are biases. So far, the edges in the relevance graph were weighted by `literal_sim` and `semantic_sim`.

In addition, we employ a post-processing procedure over the similarity results. Considering the MH-QA's core, the directly relevant phrases to the answer can hardly be identified via one hop. Intuitively, these phrases tend to exist in the sentences containing core nodes (i.e., phrases in the question). Thus, we emphasize that the phrases co-appear with the core node to concentrate on the true relevant phrases. To be more specific, a max-pooling strategy is leveraged to intensify the signal from highly relevant phrases. For each phrase t in the question and each supporting sentence, we obtain the maximum similarity score between t and every phrase of the sentence and use this value to update all the others' values. Moreover, such post-processing can also prevent less relevant information from influencing the aggregation of important signals.

3.3.2. Reformulation via Fusion

After deriving the weighted relevance in the graph, we are ready to fuse them with the question representation. The information to be fused can be described by

$$\mathbf{o}_i^{\text{lit}} = \sum_{j \in m} \text{literal_sim}(S_i, S_j) \mathcal{F}_{\text{lit}}(\tilde{\mathbf{E}}_j) \tag{7}$$

$$\mathbf{o}_i^{\text{sem}} = \sum_{j \in m} \text{semantic_sim}(\tilde{\mathbf{E}}_i, \tilde{\mathbf{E}}_j) \mathcal{F}_{\text{sem}}(\tilde{\mathbf{E}}_j), \tag{8}$$

where $i \in n$ denotes one of the n core nodes. m are the peripheral nodes' set of core node i . $S_{i(j)}$ denotes the character string of phrase $i(j)$. $\tilde{\mathbf{E}} \in R^d$ denotes the representation of nodes smoothed by mean-pooling. \mathcal{F}_{lit} and \mathcal{F}_{sem} define transformations on the literal and semantic level representations, implemented with a multi-layer perceptron.

Then, we design a fusion layer to propagate the information over the graph. Using weighting by literal similarity,

$$\mathbf{u}_i^{\text{lit}} = [\tilde{\mathbf{E}}_i, \mathbf{o}_i^{\text{lit}}, \mathcal{F}_e(\tilde{\mathbf{E}}_i), \mathcal{F}_o(\mathbf{o}_i^{\text{lit}})], \tag{9}$$

$$\mathbf{g}_i = \text{sigmoid}(\mathcal{F}_g(\mathbf{u}_i^{\text{lit}})), \tag{10}$$

$$\tilde{\mathbf{o}}_i^{\text{lit}} = \mathbf{g}_i \odot \text{relu}(\mathcal{F}_u(\mathbf{u}_i^{\text{lit}})) + (1 - \mathbf{g}_i) \odot \tilde{\mathbf{E}}_i, \tag{11}$$

where $\mathbf{u}_i^{\text{lit}} \in R^{4d}$ is the combination of the computed update and the original representation, $\mathcal{F}_e, \mathcal{F}_o, \mathcal{F}_g, \mathcal{F}_u$ are transformations on the representation, \odot is element-wise multiplication, and \mathbf{g}_i is a balancing parameter. Similarly, the computation via weighting by semantic similarity can be conducted.

Finally, we obtain two globally aggregated representations as

$$\tilde{\mathbf{o}}_i = [\tilde{\mathbf{o}}_i^{\text{lit}}, \tilde{\mathbf{o}}_i^{\text{sem}}]. \quad (12)$$

Consequently, we define $\tilde{\mathbf{O}} = [\tilde{\mathbf{o}}_1, \dots, \tilde{\mathbf{o}}_n]$, $\tilde{\mathbf{O}} \in R^{2n \times d}$ as an information *clue* to update the question. To latently reformulate the question, we compute the attention between questions and clues. We use the fusion layer to combine the information by

$$\alpha_i = \text{relu}(\mathcal{F}_q(\mathbf{E}_q) \cdot \tilde{\mathbf{O}}_i), \quad (13)$$

$$\mathbf{z} = \text{relu}(\sum_{i \in 6n} \frac{\alpha_i}{\sum_{j \in 2n} \alpha_j} \tilde{\mathbf{O}}_i), \quad (14)$$

$$\mathbf{E}_{Q'} = \text{fusion}(\mathbf{E}_Q, \mathbf{z}), \quad (15)$$

where \mathbf{E}_Q is the representation of the question, and fusion is the same layer as in Equations (9)–(11). Finally, we derive the new representation of the question, denoted by $\mathbf{E}_{Q'}$. Additionally, we utilize re-attention mechanism [35] to update context paragraphs. This post-processing is to help the model locate the correct span of the answer.

3.4. Multi-Task Prediction

Distinct from the popular structure of the prediction layer [8], we conceive a revised prediction method via multi-task learning. It learns to infer (1) supporting context paragraph; (2) answer paragraph; (3) supporting sentences; (4) starting position of the answer; (5) ending position of the answer; and (6) answer type.

The intuition to incorporate two additional prediction tasks (i.e., supporting context paragraph and answer paragraph) is that in many situations, a single supporting sentence is insufficient to obtain the complete and accurate reason path. The task of predicting supporting context paragraphs is expected to help further reduce the number of context paragraphs from n (5 in our experiment) to what we really need (2 in our experiment). In this way, the information is filtered for effective answer prediction. Akin to that, the task of predicting the answer paragraph further helps to locate the selected golden context paragraph from the predicted context paragraphs.

Merging the tasks into a global optimization framework, we have

$$\begin{aligned} \mathcal{L} = & \gamma_1 \mathcal{L}_{\text{sta}} + \gamma_2 \mathcal{L}_{\text{end}} + \gamma_3 \mathcal{L}_{\text{sup_cont}} + \gamma_4 \mathcal{L}_{\text{ans_cont}} \\ & + \gamma_5 \mathcal{L}_{\text{sup_sent}} + \gamma_6 \mathcal{L}_{\text{ans_type}}, \end{aligned} \quad (16)$$

where γ_i is a hyperparameter, and each component is defined by cross-entropy loss corresponding to the six tasks, respectively.

4. Results

This section reports experiments with in-depth analysis.

4.1. Experimental Setup

4.1.1. Dataset

We evaluate competing methods on HotpotQA benchmark (<https://hotpotqa.github.io>, accessed on 27 February 2018). HotpotQA is the first multi-hop QA dataset concentrating on the explanation ability of models, based on Wikipedia articles containing many domains, e.g., education and history. The dataset is constructed in the way that, given multiple documents, crowd workers are required to provide a question, corresponding answer and support sentences, which are used to reach the answer as shown in Figure 3. The gold documents annotation can be derived from the support sentences annotation. There are about 90,000 training samples, 7400 development samples, and 7400 test samples. Please refer to the original paper [8] for more details.

Note that there are two settings in HotpotQA—distractor and full wiki—the latter of which expands the scale of context to the whole Wikipedia. It can be seen that the main

difference between the two tasks lies in context retrieval. To concentrate on the MH-QA part, we use the distractor setting and omit the results on the full wiki in the interest of space.

4.1.2. Implementation Details

The encoding layer of IP-LQR is built on uncased BERT, and the parameters follow the setting of Ref. [33]. We set $n = 5$ to guarantee that the outcomes contain both the support context and answer context and acquire 98.86% coverage rate of the two goals. In the training phase, we use grid search to obtain the value of γ , and the weights of losses are set as $\gamma_1 = \gamma_2 = 0.019$, $\gamma_3 = \gamma_4 = \gamma_5 = 0.189$, $\gamma_6 = 0.385$. α is set as 0.7, which is the probability threshold to select support sentences. The version of the Stanford CoreNLP toolkit is full (<https://nlp.stanford.edu/software/corenlp-backup-download.html>, accessed on 27 February 2018), and we mainly apply pos_tag module to acquire the POS of each word in English for further processing.

Standard evaluation metrics for HotpotQA are adopted. Exact match (EM) means that the prediction is identical to the ground truth; F_1 considers the prediction and recall of words in prediction.

4.1.3. Competitors

The following three methods are employed for comparison: (1) Baseline [8] incorporates three NLP techniques: character-level models, self-attention, and bi-attention. To resolve yes/no questions, a three-way classifier is designed after the last recurrent layer to produce the probabilities of “yes”, “no”, and span-based answers. (2) KGNN [9] extracts entities from documents to build an entity graph. Then, if there is a relation between two entities in the external knowledge graph, they set an edge between them in the entity graph. (3) DFGN [7] also constructs an entity graph and then utilizes a graph attention method to achieve information aggregation. Finally, the updated representations help the final prediction.

4.2. Results and Analysis

4.2.1. Overall Performance

In Table 2, we show the performance comparison among different models on the development set of HotpotQA. Our method improves more than 21% and 20% absolutely in terms of joint EM and F_1 over the baseline.

Table 2. Performance results of comparing methods. Ans denotes answer accuracy, Sup Fact denotes the support paragraphs accuracy and Joint denotes the accuracy where the answer and support paragraphs are all correct. The bold denotes the best result in each metric.

Model	Ans		Sup Fact		Joint	
	EM	F_1	EM	F_1	EM	F_1
Baseline	44.44	58.28	21.95	66.66	11.56	40.86
KGNN	50.81	65.75	38.74	76.79	22.40	52.82
DFGN	55.66	69.34	53.10	82.24	33.68	59.86
IP-LQR	53.89	70.40	56.46	84.06	33.66	61.10

In the Sup Fact class, IP-LQR outperforms others in both EM and F_1 , which proves the superiority of our design. As we focus on the message passing through phrases, more clues can be exploited during the model’s inference, making it more accurate to locate supporting sentences than the entity-based method (i.e., KGNN and DFGN). Moreover, the reformulation of the question can also benefit not only the key sentences referring to the first hop, but also in the second hop. These features contribute to the results.

In Ans EM, we obtain the highest score in F_1 but second in EM. The explanation might simply be that we use the Stanford CoreNLP toolkit to generate phrases, which cannot

guarantee the quality of phrases. Empirically, the phrase is a domain concept and might vary in different fields, which makes its boundary hard to determine. This is also a potential direction we would like to research in the future. Most of the answers in HotpotQA are entities and the results of DFGN are constrained in entities, which might also increase the EM of Ans.

We proceed with the example in Figure 1 for a case study. In Table 3, “literal” represents the literal_sim, and “semantic” represents semantic_sim. The phrase “former Soviet statesman” is suitable for narrowing the field for next-hop reasoning, while the entity “Soviet statesman” brings in a noisy reasoning path. To evaluate whether phrases can propagate more information to the query, we use the phrase-updated question representation to compute the semantic_sim with the answer “World Summit of Nobel Peace Laureates”. The relevance score of phrase-based method is 0.14 higher than the score of the entity-based method. Finally, the phrase-based method correctly predicts “World Summit of Nobel Peace Laureates” as the answer, but the entity-based method extracts “Organizations” from other sentences. This cases proves the superiority of phrases in information propagation.

Table 3. Similarities of phrases in the exapmle.

Relevant Phrases	Former Soviet Statesman		Soviet Statesman	
	Literal	Semantic	Literal	Semantic
Mikhail Gorbachev Nikolai Viktorovich Podgorny	1.00	0.92	0.70	0.80
Andrei Pavlovich Kirilenko	0.74	0.88	1.00	0.82
World Summit of Nobel Peace Laureates	0.74	0.85	1.00	0.84
	0.26	0.60	1.00	0.46

4.2.2. Ablation Study

To evaluate the contributions of different components in IP-LQR, we also design an ablation study on different model components and choose the joint accuracy as a metric.

As Table 4 shows, every module contributes to the IP-LQR performance. Once a module is removed, the result will meet a decrease, clarifying our method’s effectiveness.

Table 4. Performance results of ablation study.

Setting	EM	F_1
Full model	33.66	61.10
- phrase nodes	30.32	58.31
- literal_sim	31.59	59.54
- semantic_sim	31.38	59.23
- max-pooling strategy	31.67	59.75

Specifically, when phrases in the graph are replaced by entities (Stanford CoreNLP), the loss of the metric dramatically decreases, i.e., 3.34% in EM and 2.79% in F_1 . The reason might be that, compared with the entity, the phrase contains more clues relevant to the message passing. Moreover, the result proves its utility in accurately locating the support sentences and the target answer. When it comes to the similarity methods, the decrease in the scores indicates that our latent reformulation methods can effectively capture the implicit connections among nodes. The performance of the model with semantic_sim slightly beats literal_sim, suggesting that semantic connections between different nodes outweigh the literal counterparts in most cases. The last item shows the performance that passes the original relevance score of the peripheral nodes—instead of values after max-pooling—to the core nodes. The result confirms the effectiveness of our method in finding relevant nodes.

5. Conclusions

We propose incorporating phrases in the latent query reformulation (IP-LQR) method to handle the MH-QA problem. IP-LQR first treats phrases in the question as central nodes and connects them with other phrases from relevant contexts based on literal and latent similarities. The graph is then applied to augment the information of central phrases and finally reformulate the query. IP-LQR attains competitive results on HotpotQA. In the future, we aim to design a novel phrase extraction method to improve the construction of the phrase graph and address the challenges of more complicated questions, such as comparison and judgment query.

Author Contributions: Conceptualization, J.T. and S.H.; methodology, Z.C.; software, Z.T.; validation, H.X. and Z.T.; formal analysis, J.T.; investigation, S.H.; resources, Z.T.; data curation, Z.C.; writing—original draft preparation, Z.C.; writing—review and editing, Z.T.; visualization, S.H.; supervision, H.X.; project administration, J.T.; funding acquisition, H.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NSFC grant numbers 61902417, 71971212.

Data Availability Statement: Not applicable.

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

References

1. Chen, D.; Bolton, J.; Manning, C.D. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Berlin, Germany, 7–12 August 2016; Volume 1.
2. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, TX, USA, 1–4 November 2016; pp. 2383–2392.
3. Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; Hovy, E.H. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, 9–11 September 2017; pp. 785–794.
4. Chen, D.; Fisch, A.; Weston, J.; Bordes, A. Reading Wikipedia to Answer Open-Domain Questions. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 1870–1879.
5. Welbl, J.; Stenetorp, P.; Riedel, S. Constructing Datasets for Multi-hop Reading Comprehension Across Documents. *arXiv* **2018**, arXiv:1710.06481.
6. Nishida, K.; Nishida, K.; Nagata, M.; Otsuka, A.; Saito, I.; Asano, H.; Tomita, J. Answering while Summarizing: Multi-task Learning for Multi-hop QA with Evidence Extraction. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019; Volume 1, pp. 2335–2345.
7. Qiu, L.; Xiao, Y.; Qu, Y.; Zhou, H.; Li, L.; Zhang, W.; Yu, Y. Dynamically Fused Graph Network for Multi-hop Reasoning. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019; Volume 1, pp. 6140–6150.
8. Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.W.; Salakhutdinov, R.; Manning, C.D. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2369–2380.
9. Ye, D.; Lin, Y.; Liu, Z.; Liu, Z.; Sun, M. Multi-Paragraph Reasoning with Knowledge-enhanced Graph Neural Network. *arXiv* **2019**, arXiv:1911.02170.
10. Katz, B. *Using English for Indexing and Retrieving*; Computer-Assisted Information Retrieval (Recherche d'Information et ses Applications); RIAO: Cambridge, MA, USA, 1988; pp. 313–333.
11. Ravichandran, D.; Hovy, E.H. Learning surface text patterns for a Question Answering System. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002; pp. 41–47.
12. Wang, B.; Yao, T.; Zhang, Q.; Xu, J.; Tian, Z.; Liu, K.; Zhao, J. Document Gated Reader for Open-Domain Question Answering. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, 21–25 July 2019; pp. 85–94.
13. Zhang, Z.; Wu, Y.; Zhou, J.; Duan, S.; Zhao, H.; Wang, R. SG-Net: Syntax-Guided Machine Reading Comprehension. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 9636–9643.

14. Nishida, K.; Saito, I.; Otsuka, A.; Asano, H.; Tomita, J. Retrieve-and-Read: Multi-task Learning of Information Retrieval and Reading Comprehension. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, 22–26 October 2018; pp. 647–656.
15. Min, S.; Zhong, V.; Socher, R.; Xiong, C. Efficient and Robust Question Answering from Minimal Context over Documents. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 1725–1735.
16. Pirtoaca, G.; Rebedea, T.; Ruseti, S. Answering questions by learning to rank - Learning to rank by answering questions. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019; pp. 2531–2540.
17. Yao, X.; Durme, B.V. Information Extraction over Structured Data: Question Answering with Freebase. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, Baltimore, MD, USA, 22–27 June 2014; Volume 1, pp. 956–966.
18. Zhao, W.; Chung, T.; Goyal, A.K.; Metallinou, A. Simple Question Answering with Subgraph Ranking and Joint-Scoring. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 324–334.
19. Fu, B.; Qiu, Y.; Tang, C.; Li, Y.; Yu, H.; Sun, J. A Survey on Complex Question Answering over Knowledge Base: Recent Advances and Challenges. *arXiv* **2020**, arXiv:2007.13069.
20. Shi, J.; Cao, S.; Hou, L.; Li, J.; Zhang, H. TransferNet: An Effective and Transparent Framework for Multi-hop Question Answering over Relation Graph. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event/Punta Cana, Dominican Republic, 7–11 November 2021; pp. 4149–4158.
21. Saxena, A.; Tripathi, A.; Talukdar, P.P. Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020; pp. 4498–4507.
22. Liang, C.; Berant, J.; Le, Q.V.; Forbus, K.D.; Lao, N. Neural Symbolic Machines: Learning Semantic Parsers on Freebase with Weak Supervision. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 23–33.
23. Guo, D.; Tang, D.; Duan, N.; Zhou, M.; Yin, J. Dialog-to-Action: Conversational Question Answering Over a Large-Scale Knowledge Base. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montréal, ON, Canada, 3–8 December 2018; pp. 2946–2955.
24. Saha, A.; Ansari, G.A.; Laddha, A.; Sankaranarayanan, K.; Chakrabarti, S. Complex Program Induction for Querying Knowledge Bases in the Absence of Gold Programs. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 185–200. [[CrossRef](#)]
25. Ma, Y.; Nguyen, K.L.; Xing, F.Z.; Cambria, E. A Survey on Empathetic Dialogue Systems. *Inf. Fusion* **2020**, *64*, 50–70. [[CrossRef](#)]
26. Choi, E.; He, H.; Iyyer, M.; Yatskar, M.; Yih, W.; Choi, Y.; Liang, P.; Zettlemoyer, L. QuAC: Question Answering in Context. In Proceedings of the EMNLP 2018, Brussels, Belgium, 31 October–4 November 2018; pp. 2174–2184.
27. Huang, H.; Choi, E.; Yih, W. FlowQA: Grasping Flow in History for Conversational Machine Comprehension. In Proceedings of the ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.
28. Reddy, S.; Chen, D.; Manning, C.D. CoQA: A Conversational Question Answering Challenge. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 249–266. [[CrossRef](#)]
29. Yatskar, M. A Qualitative Comparison of CoQA, SQuAD 2.0 and QuAC. In Proceedings of the NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 2318–2323.
30. Chaturvedi, I.; Cambria, E.; Welsch, R.E.; Herrera, F. Distinguishing between facts and opinions for sentiment analysis: Survey and challenges. *Inf. Fusion* **2018**, *44*, 65–77. [[CrossRef](#)]
31. Ma, L.; Zhang, W.; Li, M.; Liu, T. A Survey of Document Grounded Dialogue Systems (DGDS). *arXiv* **2020**, arXiv:2004.13818.
32. Merdivan, E.; Singh, D.; Hanke, S.; Holzinger, A. Dialogue Systems for Intelligent Human Computer Interactions. *Electron. Notes Theor. Comput. Sci.* **2018**, *343*, 57–71. [[CrossRef](#)]
33. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
34. Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; McClosky, D. The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 22–27 June 2014; pp. 55–60.
35. Hu, M.; Peng, Y.; Huang, Z.; Qiu, X.; Wei, F.; Zhou, M. Reinforced Mnemonic Reader for Machine Reading Comprehension. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, 13–19 July 2018; pp. 4099–4106.