This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Zhu, Hongyin; Tiwari, Prayag; Ghoneim, Ahmed; Hossain, M. Shamim

A Collaborative AI-enabled Pretrained Language Model for AIoT Domain Question Answering

Published in:
IEEE Transactions on Industrial Informatics

DOI:
[10.1109/TII.2021.3097183](https://doi.org/10.1109/TII.2021.3097183)

Published: 01/05/2022

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Zhu, H., Tiwari, P., Ghoneim, A., & Hossain, M. S. (2022). A Collaborative AI-enabled Pretrained Language Model for AIoT Domain Question Answering. *IEEE Transactions on Industrial Informatics*, 18(5), 3387-3396. <https://doi.org/10.1109/TII.2021.3097183>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

A Collaborative AI-enabled Pretrained Language Model for AIoT Domain Question Answering

Hongyin Zhu¹, Prayag Tiwari², Ahmed Ghoneim³, and M. Shamim Hossain⁴

Abstract—Large-scale knowledge in the Artificial Intelligence of Things (AIoT) field urgently needs effective models to understand human language and automatically answer questions. Pre-trained language models (PLMs) achieve state-of-the-art performance on some question answering (QA) datasets, but few models can answer questions on AIoT domain knowledge. Currently, AIoT domain lacks sufficient QA datasets and large-scale pre-training corpora. We propose RoBERTa_{AIoT} to address the problem of the lack of high-quality large-scale labeled AIoT QA datasets. We construct an AIoT corpus to further pre-train RoBERTa and BERT. RoBERTa_{AIoT} and BERT_{AIoT} leverage unsupervised pre-training on a large corpus composed of AIoT-oriented Wikipedia webpages to learn more domain-specific context and improve performance on the AIoT QA tasks. To fine-tune and evaluate the model, we construct 3 AIoT QA datasets based on the community QA websites. We evaluate our approach on these datasets and the experimental results demonstrate the significant improvements of our approach.

Index Terms—AIoT, Question answering, RoBERTa, BERT, Domain-specific.

I. INTRODUCTION

Artificial Intelligence of Things (AIoT) has become a promising development trend, which contains many aspects of knowledge, such as big data, blockchain, cloud computing [1], [2], edge computing [3], machine learning [4], [5], 5G network [6], the Internet of things (IoT) [7], [8], [9], and industrial applications [10]. With the rapid development of the Internet, artificial intelligence (AI) and IoT, AIoT is constantly being given new connotations [8], [7], [9]. However, many enthusiasts and developers lack a comprehensive understanding of the content of AIoT and will encounter various problems related to AIoT, which is also a challenge for the development of AIoT. With the passage of time, the scope of AIoT continues to expand, and learning the corresponding knowledge becomes more and more challenging. Allowing machines to automatically answer questions about AIoT knowledge has become an urgent need for the development of AIoT community.

Passage reranking [11], [12] aims to rerank the relevant answering passages [13] for the question and return the top-ranked passages as the final answer, which is an indispensable module in the question answering pipeline. Each passage is a text fragment, containing one to several sentences. Table I gives an example to illustrate the task form. We can see

the first passage “*I don’t accept the premise, as it’s far too broad...*” can best answer the question “*Why are 5G networks so unreliable?*”. Although other passages mention 5G networks, their central idea is not to discuss the reliability of 5G networks. This means that a more accurate understanding of the AIoT-related concepts and context in the passage is a critical problem.

TABLE I: Example of the question answering task

Question: Why are 5G networks so unreliable?
Passages:
1. I don’t accept the premise, as it’s far too broad. Perhaps a better question is “why are current mmWave 5G networks so difficult to maintain a connection with as I move around”? That’s because there aren’t that many base stations yet, and the range of mmWave radios is fairly short. There actually is a part of 5G that is much more reliable of a radio connection than 4G, but using more forward error correction. That functionality probably hasn’t been enabled yet.
2. I recently did some market research in 5G and converted that into a guide which has described the research of some big telecom companies. The guide has more than 10 companies who intensely research on 5G technology. And the company which seemed leading on 5G to me is Ericsson. Why? You asked. Here is why. Ericsson claims to be the only vendor that is currently working in all continents to make 5G a global standard for the next generation of wireless technology...
3. Why is YouTube banning conspiracy theories videos about 5G networks being connected to the Coronavirus? Well most of the actions in the videos are either dangerous, or illegal so to prevent copy cat crimes they are being banned. Also they are either censoring the truth of the connection or the more li Why is YouTube banning conspiracy theories videos about 5G networks being connected to the Coronavirus? ...
...
Answer: 1.

Question answering is one of the most important tasks of natural language understanding (NLU). Large pre-trained language models (PLMs), such as BERT [14], RoBERTa [15], ELECTRA [16], GPT-3 [17], etc., achieve state-of-the-art results on some QA datasets, such as the GLUE benchmark [18]. These models are pre-trained on a large-scale unlabeled corpus and then fine-tuned for specific tasks. The understanding of natural language is inseparable from the pre-training process of large-scale corpora. However, there are relatively few researches on knowledge question answering in the AIoT domain. Existing pre-trained language models lack AIoT knowledge. Besides, the ambiguity of the question and the wide range of content contained in the passage pose great challenges to accurate language understanding.

Expanding PLMs to the AIoT domain is a meaningful problem. We hope to use the AIoT domain large-scale corpus to further pre-train the PLMs, so as to learn the AIoT knowledge and context. However, there is no directly available high-quality and restrictive AIoT knowledge corpus. Meanwhile, there is no manually annotated QA dataset on AIoT domain knowledge. The main challenges lie in the construction of the pre-training corpus and the evaluation of model performance.

In this paper, we propose to construct a pre-training corpus in the AIoT domain. We use this corpus to further pre-train the RoBERTa and BERT to expand the model capacity to the AIoT domain. In the learning process, the model converts

This work was supported by the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia for funding this work through the research group project no. RGP-229. (Corresponding authors: M. Shamim Hossain)

Hongyin Zhu and Prayag Tiwari contributed equally to this work and share the joint first authorship.

H. Zhu is with the Department of Computer Science and Technology, Tsinghua University, China (email: zhuhongyin2020@mail.tsinghua.edu.cn).

P. Tiwari is with the Department of Computer Science, Aalto University, Finland (email: prayag.tiwari@aalto.fi).

A. Ghoneim is with the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia and Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Shebin El-Koom 32511, Egypt (email: ghoneim@ksu.edu.sa).

M. Shamim Hossain is with the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia (email: mshossain@ksu.edu.sa).

learned domain knowledge into hidden parameters. To fine-tune and evaluate the model, we construct 3 QA datasets on AIoT domain knowledge through the community QA websites, such as stackoverflow.com, quora.com and superuser.com. Our method improves the model performance on these 3 AIoT QA datasets. The main contributions of this paper are as follows.

(i) We further pre-train RoBERTa_{AIoT} and BERT_{AIoT} using an AIoT domain corpus to expand the model capacity to the AIoT domain.

(ii) We construct 3 AIoT QA datasets and a pre-training corpus to train and evaluate the QA models.

(iii) Our method improves the model performance on AIoT QA tasks.

II. RELATED WORK

A. Pre-trained Language Models

Peters et al. [19] pre-train a 2-layer LSTM encoder with a bidirectional language model to generate ELMo (Embeddings from Language Models). The contextual representation of the pre-trained ELMo shows significant improvements in various NLP tasks. BERT [14] propose an approach to directly model the representation of the sentence pair. Composed of the bidirectional transformer encoders [20], BERT uses masked language model (MLM) loss and next sentence prediction (NSP) loss for pre-training.

RoBERTa [15] propose to use an enhanced version MLM to further improve BERT by using dynamic masking. GPT-2 [21] propose to pre-train the language model in a large corpus and multi-task learning setting, and perform downstream tasks in a zero-shot setting. To better model inter-sentence coherence, ALBERT [22] replace NSP loss with a sentence order prediction (SOP) loss. ELECTRA [16] improves replaced token detection (RTD) by utilizing a generator to replacing some tokens of the sequence. PLMs have achieved the best results in modeling the representation of sentence pairs. Inspired by the above works, we choose to use the PLMs to model the QA representation. PLMs have been further pre-trained on scientific publications and the biomedical domain, but PLMs have not been expanded to the AIoT domain.

B. Traditional Deep Learning Methods for QA

For traditional deep learning methods, prior works adopt deep learning models (such as CNN, Bi-LSTM. etc.) to enhance the sentence representation and calculate various similarities. Rocktaschel et al. [23] propose a model for textual entailment, which models the word relationship between sentence pair by using word-to-word attention over LSTM-RNNs. Severyn et al. [24] present the CNN_R to model the relational information between a QA pair with the overlapping words. Miller et al. [25] propose three attention schemes on CNN (ABCNN) to integrate the mutual influence between the sentence pair.

Miller et al. [26] propose the key-value memory network (KV-MemNN) to predict the answer by using facts in a key-value structured memory. Wang et al. [27] present the bilateral multi-perspective matching (BiMPM) model that uses the attention mechanism to establish the interactions of the sentence pair from different scales. Although there are relatively few parameters in the aforementioned neural networks compared with pre-trained language models, they provide a lot of insight for the deep learning QA tasks.

III. METHODS

This section explains the mechanism of the RoBERTa_{AIoT} for QA. Suppose we have a question Q with l tokens $\{w_1^q, w_2^q, \dots, w_l^q\}$ and a candidate passage set O including n

passages $\{O_1, O_2, \dots, O_n\}$, where n can vary over a wide range and the superscript ^q denotes the question. Passage O_i consists of m tokens $\{w_1^o, w_2^o, \dots, w_m^o\}$, where the superscript ^o denotes the passage. The label of O_i is $y_i \in \{0, 1\}$ with 1 indicating a positive answer and 0 otherwise. Our goal is to learn a neural network that can assign a score to each passage to reflect how well it matches the question. Passages with higher matching scores are ranked above the ones with lower scores. The top-ranked passages are taken as the final answer.

Our main effort lies in pre-training the PLMs with domain-specific corpus and fine-tuning the domain-adaptive PLMs to the AIoT QA task. We first describe the way we construct the AIoT domain corpus and QA datasets. Then, we describe the model architecture, pre-training and fine-tuning methods.

Expanding PLMs to the AIoT domain is an essential task. If the language cannot be understood automatically, the efficiency will be low in the case of big data. The reason that there is not much relevant research in the topic of AIoT NLP before is that there are no systematic methods and resources to expand PLMs to the AIoT domain. Although this article has already carried out this task in the AIoT domain, this solution can also be applied to other fields. In addition to providing models, tasks, datasets and corpora, the contribution is also that we have introduced a new AIoT NLP solution.

A. Constructing Domain-specific Corpus

To obtain pre-training data, we use Wikipedia webpages as the data source. We selected representative terms of AIoT and obtained their Wikipedia webpages, and then extracted all the anchor text on these webpages. Based on these anchor text, we can get the corresponding Wikipedia webpages. In this iteration, we filter out some irrelevant anchor text through manual reading. Since the number of webpages obtained in the first iteration is not enough, the above process is repeated based on the obtained webpages, and all webpages corresponding to the anchor text are obtained again. Using 8 initial seeds (5G, Amazon Echo, Blockchain, Clouding computing, Edge computing, Google Nest, HomePod, Internet of things), in the first iteration we obtained 2,195 webpages, and in the second iteration we obtained 73,663 webpages. Many other terms do not have a corresponding Wikipedia webpage.

We use the most representative terms in the AIoT field as a starting point. Non-core AIoT terms (e.g. embedded systems, smartphones, Microsoft) will be found in the iterative search process on Wikipedia. Although the Wikipedia webpages contain some non-core AIoT terms, it is not accidental that these terms appear on these Wikipedia webpages. These non-core terms are related to AIoT at different levels. The automatic construction process inevitably introduces noise but our automatic construction method greatly improves efficiency. In addition, the language model learns contextual expression, and all Wikipedia webpages are high-quality corpora. Even Wikipedia articles of non-core AIoT terms also help the learning process.

B. Constructing QA Datasets Using Coarse Rankers

1) *Data Acquisition:* To build QA datasets, we collect questions and answers related to AIoT from quora.com, superuser.com and stackoverflow.com websites. Specifically, we use representative AIoT keywords (such as 5G network, blockchain, edge computing, clouding computing, Amazon echo, etc.) to search on the website and collect all retrieved questions. Since people pay different attention to different keywords, and some keywords do not have many questions. We did not change the distribution of these problems, and directly adopted all related questions. Then, we extract questions and passages through DOM parsing.

2) *Coarse Rankers*: This section describes the process of selecting some candidate passages for each question. Training a QA model requires both positive answers and negative answers. As there are no available datasets for this AIoT QA task, we use symbolic heuristic methods to get relevant passages.

Since the community Q&A website is open to all users on the Internet, it contains high-quality and low-quality replies. For the question Q , we have a list of replies $[O_1^Q, O_2^Q, \dots, O_n^Q]$ in the webpage. Moreover, some replies are colloquial, often lacking thorough and precise expression. Therefore, this poses a challenge for automatically selecting answers to questions. In order to reduce the influence of noise, we limit each question to only one correct answer O_i^Q and consider choosing the passage with higher upvotes and longer content as the correct answer. We observed that longer answers usually describe more complete and thorough content, which helps to improve the quality of the dataset. We also test the randomly selected answer, and the model performance is almost the same. The description of some short answers is abstract and arbitrary. This research can be extended to the corpus of scientific literature in the future. Using passages obtained in the literature or other high-quality paragraphs such as Wikipedia articles to build the dataset will help the model to return more accurate answers.

For the negative answers of a question, we use BM25 to score and sort the set of all answers to other questions $\hat{q}$. BM25 is used to generate high-quality negative candidates from all the passages. We use BM25 to filter out irrelevant candidates. In doing so, it improves the quality of the dataset. At the same time, the difficulty of the problem is increased, because the remaining candidates are sort of confusing. In addition, we select the 30 candidates with the highest scores.

$$score(Q, O) = \sum_{q_i \in Q} IDF(q_i) \cdot \frac{f_{q_i, O} \cdot (k_1 + 1)}{f_{q_i, O} + k_1 \cdot (1 - b + b \cdot \frac{|O|}{avgdl})} \quad (1)$$

where $f_{q_i, O}$ is the term frequency of q_i in the passage O . $avgdl$ is the average passage length. k_1 and b are hyper-parameters.

$$IDF(q_i) = \ln\left(\frac{N - n_{q_i} + 0.5}{n_{q_i} + 0.5} + 1\right) \quad (2)$$

where N is the number of passages. n_{q_i} is the number of documents that contains q_i .

However, sometimes different questions express the same meaning, so the answers to other questions can answer the target question (false negative answers).

$$\hat{q} \in \{\hat{q} | Q \setminus \{q\}\} \quad (3)$$

where q and $\hat{q}$ are the target question and other questions respectively. As shown in formula (3), to prevent other questions $\hat{q}$ from expressing the same meaning as the target question q , we use TF-IDF to score the similarity between the questions and filter out similar questions, calculated by the following equation.

$$similarity = \cos(\psi(q), \psi(\hat{q})) \quad (4)$$

where $\psi(\cdot)$ is the vectorization (mapping) function to convert text to TF-IDF vector. q and $\hat{q}$ are seen as different documents. Then we calculate the cosine similarity. Specifically, we calculate the TF-IDF of the corpus composed of all questions, as shown below.

$$\psi(q)_t = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \times \log \frac{N}{n_t} \quad (5)$$

where d is the document that contains term t . $f_{t,d}$ is the term frequency of t in document d . N is the number of documents of the corpus. n_t is the number of documents that contains t .

Here we apply coarse rankers to improve the quality of the datasets. Coarse rankers are indispensable in the process of building datasets because each question has thousands of passages in total. The first step is to perform a rough selection from all the candidates, and then to finely select the answers, otherwise the computational cost is high. Selecting the most relevant candidates creates a more challenging dataset. Especially when sampling negative samples, we should remove similar questions, because the answers to these questions are seen as false negative if they are also treated as negative samples. These careful steps are critical to the creation of the dataset and model training.

C. Reranking Passages Using Domain-adaptive Fine Rankers

Fig. 1: Neural architecture overview of the RoBERTa_{AIoT}

1) *Fine-tuning Domain-Adaptive PLMs for AIoT QA*: The neural architecture of our RoBERTa_{AIoT} is shown in Figure 1. In the input layer, for passage O_i , it is concatenated with the question Q and 2 special symbols, classification [CLS] and separator [SEP], denoted as $\langle [CLS]; Q; [SEP]; O_i; [SEP] \rangle$.

We use this network to output the representation of QA pairs for subsequent scoring of QA pairs.

$$h_i^{[CLS]}, h_i^w = \mathcal{F}_{PLM}^{(AIoT)}(\langle [CLS]; Q; [SEP]; O_i; [SEP] \rangle) \quad (6)$$

where $h_i^{[CLS]} \in \mathbb{R}^d$ is the representation of [CLS]. h_i^w denote the representation of every tokens in the text sequence. $\mathcal{F}_{PLM}^{(AIoT)}(\cdot)$ denotes the PLM after domain-specific pre-training. Then we use a single layer neural network to calculate the matching score.

$$p_i = \text{softmax}(Wh_i + b) \quad (7)$$

where $W \in \mathbb{R}^{d \times 2}$ and b are the weight and bias parameters. $\text{softmax}(\cdot)$ is the softmax function that outputs the probability of being a correct answer or not so that we can treat each QA pair as a classification task. The input to this layer is a list of QA pairs, $[(Q, O_1), (Q, O_2), \dots, (Q, O_n)]$, so this model calculates a matching score for each passage. Then we take the probability of being the correct answer to rank the passages.

The training loss is to sum the deviation of classification task. We adopt the negative log-likelihood as the objective, as shown below.

$$\mathcal{L} = - \sum_i^D \log p_i^{(t)}(y_i^{(t)} | \mathbf{h}_i; \Theta) \quad (8)$$

where $p_i^{(t)}(\cdot)$ represents the probabilities of true class of title matching task. D is the number of QA pairs.

2) *Pre-training with Domain-specific Corpus*: Howard et al. [28] show that further pre-training a language model on a target domain corpus improves the eventual classification performance. This part aims to use unstructured text in the AIoT domain to further pre-train PLMs.

In this paper, we choose to use RoBERTa and BERT as the benchmark for further pre-training because BERT and RoBERTa have achieved state-of-the-art results in this task. Each text sequence is concatenated with special symbols, classification [CLS] and separator [SEP], denoted as $\langle [CLS]; s; [SEP] \rangle$. We use this network to output the representation of each tokens in the input text for the subsequent Cloze task.

$$h^{[CLS]}, \mathbf{h}^w = \mathcal{F}_{PLM}^{(AIoT)}(\langle [CLS]; s; [SEP] \rangle) \quad (9)$$

where $\mathbf{h} \in \mathbb{R}^{d \times l}$ is the representation of each token. s is the unstructured text. d and l are dimension and sequence length. $\mathcal{F}_{PLM}^{(AIoT)}$ denotes the network defined in [14]. Then we use the masked language model objective to train the model, as calculated in equation (10). MLM objective is based on the Cloze task which aims to predict the masked words.

$$\mathcal{L}^{(mlm)} = - \sum_i \log p^{(mlm)}(y_i^{(mlm)} | \mathbf{h}; \Theta) \quad (10)$$

where $p^{(mlm)}(\cdot)$ represents the probabilities of true class of Cloze task. i denotes the indices of token.

Both BERT and RoBERTa are composed of transformer encoder blocks, and the architecture of the transformer encoder [20] is shown in Figure 1. In the following, we will briefly introduce the architecture of the transformer encoder. In the lower part of the transformer encoder, the multi-head self-attention mechanism can learn features from different perspectives, as calculated in Formula (11).

$$\vec{v}^l = (\overrightarrow{head}_1^l \oplus \dots \oplus \overrightarrow{head}_m^l) W^O \quad (11)$$

where $\oplus$ denotes the concatenate operation. The superscript l means the l -th network layer. W^O is the parameter matrix.

$$\overrightarrow{head}_i^l = softmax(\frac{\vec{h}^{l-1} W_i^Q (\vec{h}^{l-1} W_i^K)^T}{\sqrt{d_k}}) \vec{h}^{l-1} W_i^V \quad (12)$$

where the projections W_i^Q, W_i^K, W_i^V are parameter matrices. $\vec{h}^l$ is the final output of the l -th transformer encoder layer. $\vec{h}^0$ means the representation of the input layer.

The residual connection and layer normalization in the lower part of the block is calculated in Formula (13).

$$\vec{u}^l = \frac{g}{\sigma_1^l} \odot (\vec{h}^{l-1} + \vec{v}^l - \mu_1^l) + b^l \quad (13)$$

where $\vec{u}^l$ is the vector after the residual connection and layer normalization. $\vec{v}^l$ is the output of multi-head self attention in the l -th layer. g is a gain parameter for scaling the normalized activation. μ_1^l, σ_1^l, b^l are the mean, standard deviation and bias. $\odot$ denotes the element-wise multiplication.

The upper part of the transformer block, can be computed as follows.

$$\vec{h}^l = \frac{g}{\sigma_2^l} \odot (\vec{u}^l + (\max(0, W_1 \vec{u}^l + b_1^l) W_2^l + b_2^l) - \mu_2^l) + b_3^l \quad (14)$$

where W_1, W_2 are parameter matrices. μ_2^l, σ_2^l, b^l are the mean, standard deviation and bias. The μ and σ are computed as follows.

$$\mu_1^l = \frac{1}{H} \sum_{i=1}^H (\vec{h}_i^{l-1} + \vec{v}_i^l) \quad (15)$$

where H is the dimension of the vector.

$$\mu_2^l = \frac{1}{H} \sum_{i=1}^H (\vec{u}_i^l + (\max(0, W_1 \vec{u}_i^l + b_1^l) W_2^l + b_2^l)_i) \quad (16)$$

$$\sigma_1^l = \sqrt{\frac{1}{H} \sum_{i=1}^H ((\vec{h}_i^{l-1} + \vec{v}_i^l) - \mu_1^l)^2} \quad (17)$$

$$\sigma_2^l = \sqrt{\frac{1}{H} \sum_{i=1}^H ((\vec{u}_i^l + (\max(0, W_1 \vec{u}_i^l + b_1^l) W_2^l + b_2^l)_i) - \mu_2^l)^2} \quad (18)$$

IV. EXPERIMENTS

A. Data and Setup

QuoraQA is an AIoT QA dataset that is constructed based on the quora.com website. The questions are about aspects of AIoT, such as, 5G network, blockchain, edge computing, clouding computing, Amazon echo, etc. The dataset comes from quora.com users' questions and other users' answers. Each question averagely contains 30 candidate answers, only one is the best answer, and the negative candidate answers are retrieved from the answers of all other questions based on BM25. This dataset contains 755 questions, 22,835 QA pairs, and the training/development/test set is divided at a ratio of 8:1:1.

StackOverflowQA is an AIoT QA dataset that is constructed based on the stackoverflow.com website. The questions are about different aspects of AIoT, such as, 5G network, blockchain, edge computing, cloud computing, Amazon echo, etc. The dataset comes from stackoverflow.com users' questions and other users' answers. Each question averagely contains 30 candidate answers, only one is the best answer, and the negative candidate answers are retrieved from the answers of all other questions based on BM25. This dataset contains 876 questions, 26,322 QA pairs, and the training/development/test set is divided at a ratio of 8:1:1.

SuperUserQA is an AIoT QA dataset that is constructed based on the superuser.com website. The questions are about different topics of AIoT, such as, 5G network, blockchain, edge computing, Amazon echo, etc. The dataset comes from superuser.com users' questions and other users' answers. Each question averagely contains 30 candidate answers, only one is the best answer, and the negative candidate answers are retrieved from the answers of all other questions based on BM25. This dataset contains 1,083 questions, 20,153 QA pairs, and the training/development/test set is divided at a ratio of 8:1:1.

B. Evaluation

Our objective is to rank the candidate passages based on their relatedness to the question. The two metrics used to evaluate the quality of our model are Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) which are commonly used. We calculate the MRR@10, MRR@5 and MRR@1 which considers only the top-10, top-5 and top-1 candidates respectively.

C. Hyper-parameters

This paper uses the roberta-base and bert-base-uncased models as the baselines to output the sentence pair representations. We use the “[CLS]” vector of the last layer as the representation of the QA pair. Due to the computational limitations, we set 400 tokens as the maximum QA pair length. To avoid the influence of random initialization, we run each experiment 3 times and take the median value as the final result.

We use the AdamW [29] optimization algorithm to update the model parameters. We fine-tune the PLMs to adapt the model to a new domain. The learning rate is $1 * 10^{-5}$. These experiments are run on the Intel(R) Xeon(R) CPU @ 2.00GHz (Mem: 12G) and 4 Tesla T4 (16G) GPUs. We pre-train the language model for two weeks. For the AIoT QA task, the running time of each experiment is averagely 0.3 h/epoch.

D. Results on the AIoT QA datasets

TABLE II: Results on the QuoraQA dataset

Method	MAP	MRR@10	MRR@5	MRR@1
BERT	0.4794	0.4572	0.4504	0.4189
RoBERTa	0.6217	0.6158	0.5975	0.5
ALBERT	0.5998	0.5891	0.5783	0.5
ELECTRA	0.6086	0.5985	0.5881	0.5
GPT-2	0.1821	0.1524	0.1259	0.054
BERT_{AIoT}	0.4914	0.4796	0.4559	0.3649
RoBERTa_{AIoT}	0.6420	0.6315	0.6232	0.5270

1) *Results on the QuoraQA Dataset:* Table II lists the experimental results. BERT_{AIoT} and RoBERTa_{AIoT} denote that we use AIoT corpus to further pre-train the corresponding PLMs. We perform task-specific fine-tuning on multiple PLMs, such as BERT, RoBERTa, ALBERT, ELECTRA, GPT-2. Although GPT-2 claims to implement a zero-shot learner, we found that the performance of using it directly for AIoT knowledge QA task is low, and the MAP score is about 0.1. Therefore, we performed task-specific fine-tuning on GPT-2, which improved the MAP score by 8%. This means that AIoT knowledge QA is a challenging problem.

We observe that our RoBERTa_{AIoT} achieves the best results in answering AIoT questions on the quora.com website. The MAP score of RoBERTa_{AIoT} is 2.03% higher than the MAP score of RoBERTa. The MAP score of BERT_{AIoT} is 1.2% higher than the MAP score of BERT. This shows that the AIoT corpus we built is effective for injecting domain knowledge.

Figure 2 shows the details of the experimental results. RoBERTa and RoBAIoT stand for the baseline model and RoBERTa_{AIoT} respectively. It can be seen that our method increases the proportion of correct answers in the first, sixth and some after places.

TABLE III: Results on the StackOverflowQA dataset

Method	MAP	MRR@10	MRR@5	MRR@1
BERT	0.5752	0.5681	0.5453	0.4471
RoBERTa	0.4918	0.4783	0.4654	0.3529
ALBERT	0.4931	0.4793	0.4533	0.3882
ELECTRA	0.4770	0.4639	0.4520	0.3059
GPT-2	0.1943	0.1672	0.1315	0.0823
BERT_{AIoT}	0.5486	0.5370	0.5178	0.4353
RoBERTa_{AIoT}	0.5377	0.5284	0.51	0.3882

2) *Results on the StackOverflowQA Dataset:* Table III lists the experimental results. We observe that BERT achieves the best performance. The MAP score of RoBERTa_{AIoT} is 4.59%

Fig. 2: Results analysis on the QuoraQA dataset

higher than the MAP score of RoBERTa. This demonstrates that further pre-training of RoBERTa on the AIoT corpus will improve its ability in answering AIoT questions on the stackoverflow.com website. The MAP score of BERT_{AIoT} degrades but still higher than most of the state-of-the-art models. This means that BERT has a natural advantage in answering questions on the stackoverflow.com website.

Figure 3 shows the details of the experimental results. RoBERTa and RoBAIoT stand for the baseline model and RoBERTa_{AIoT} respectively. It can be seen that our method increases the proportion of correct answers in the first, second and forth places. We can see that RoBERTa_{AIoT} achieves

Fig. 3: Results analysis on the StackOverflowQA dataset

higher results than the corresponding prototype, which proves the effectiveness of the method. BERT achieved the highest results on this dataset because different backbones have their own strengths, and have different results on different QA datasets. We also observe that BERT is prone to catastrophic forgetting [30], while other models have relatively small limitations on this problem.

3) *Results on the SuperUserQA Dataset:* Table IV lists the experimental results. We observe that the RoBERTa_{AIoT} achieves the best performance. The MAP score of

TABLE IV: Results on the SuperUserQA dataset

Method	MAP	MRR@10	MRR@5	MRR@1
BERT	0.3565	0.3321	0.3205	0.2462
RoBERTa	0.4332	0.4232	0.3931	0.2769
ALBERT	0.3711	0.3521	0.3359	0.2615
ELECTRA	0.3975	0.3814	0.3566	0.2461
GPT-2	0.1732	0.1432	0.1171	0.0769
BERT_{AIoT}	0.3476	0.3308	0.3015	0.2308
RoBERTa_{AIoT}	0.4929	0.4809	0.4623	0.3538

RoBERTa_{AIoT} is 5.97% higher than the MAP score of RoBERTa. This demonstrates that further pre-training of RoBERTa on the AIoT corpus will improve its ability in answering AIoT questions on the superuser.com website. The MAP score of BERT_{AIoT} is slightly lower than the MAP score of BERT. This means that RoBERTa is more effective than BERT in terms of learning domain adaptive models.

Figure 4 shows the details of the experimental results. RoBERTa and RoBERTa_{AIoT} stand for the baseline model and RoBERTa_{AIoT} respectively. It can be seen that our method increases the proportion of correct answers in the first, forth and fifth places.

Fig. 4: Results analysis on the SuperUserQA dataset

E. Case Study

Analyzing the experimental results on different datasets can provide an interpretable basis for the model. We select some questions on the three experimental datasets to analyze the problems of the model.

For example, Table VIII in the Appendix¹ lists the ranking of the passages predicted by the RoBERTa_{AIoT}. The first passage represents the best answer predicted by the model, and the ground truth is in the “Answer” row. Through some samples of the QuoraQA dataset (such as Table VIII, IX, XI, etc. in the Appendix), we found that the model is good at answering the questions about blockchain, AI, but not good at answering questions about 5G and agriculture IoT.

Through the samples of the StackOverflowQA dataset (such as Table XVI, XIII, XIV, XV, etc. in the Appendix), we found that the model is good at answering the questions about IBM Bluemix, database, AWS IoT but not good at answering questions about UWP, clouding IoT.

Through the samples of the SuperUserQA dataset, we found that the model is good at answering the questions about DDNS, Internet speed but not good at answering questions about Windows 10, WPA2-Enterprise.

When certain concepts appear less in the dataset, it may be difficult for the model to understand them. This also reflects that there are still few applications of the Internet of Things in the agriculture domain. On the Internet, there are relatively few questions related to AIoT in the agricultural sector.

In summary, we found that models trained on different datasets are good at answering different topics. The performance of the model is also related to the vagueness of the question expression and the similarity of passages. Different community QA websites focus on different aspects of AIoT questions. Combining models of different QA communities to answer questions raised by users is very helpful for improving the robustness of the system.

F. Question Answering without Task-specific Fine-tuning

Inspired by the knowledge probing task [31], without the task-specific fine-tuning process, we can directly use the PLMs to probe the effectiveness of the background knowledge in the AIoT QA task. We conduct experiments on three datasets. Table V, VI, VII list the results. We found that domain-adaptive RoBERTa_{AIoT} and BERT_{AIoT} achieve better results, which demonstrates that our approach injects more AIoT domain knowledge into PLMs.

Probing the model can explore the knowledge learned by the model itself after pre-training so as to better observe how well the PLM is expanded to the AIoT domain. Because if we fine-tune the language model, it will inevitably lead to changes in the hidden parameters of PLM, and some knowledge learned during the pre-training process may be diluted. Therefore, direct experiments without fine-tuning can observe the pure result of that how well the PLM has learned the knowledge of AIoT.

The hidden parameters of these models are unchanged, and this can better reflect whether the model incorporates the AIoT knowledge after pre-training. We can see that under the same model architecture, the proposed method improves the results. For example, RoBERTa_{AIoT} improves the MAP score by 23.67% on the QuoraQA dataset. Under different model architectures, we found that the hidden parameters of the ALBERT model are the strongest.

TABLE V: Probing results on the QuoraQA dataset

Method	MAP	MRR@10	MRR@5	MRR@1
BERT	0.3325	0.3140	0.2921	0.1892
RoBERTa	0.2206	0.1883	0.1642	0.0946
ALBERT	0.5232	0.5129	0.4682	0.3649
ELECTRA	0.1522	0.1192	0.0928	0.0270
GPT-2	0.2026	0.1777	0.1538	0.0676
BERT_{AIoT}	0.2641	0.2442	0.2176	0.1216
RoBERTa_{AIoT}	0.4573	0.4441	0.4266	0.2973

As shown in Table V, compared with the corresponding prototype, the MAP score of the domain-adaptive RoBERTa_{AIoT} is 23.67% higher than the baseline. As can be seen from Figure 5, the proportion of correct answers in the top-ranked positions is improved by RoBERTa_{AIoT}.

As shown in Table VI, the MAP score of the domain-adaptive RoBERTa_{AIoT} is 3.37% higher than the baseline. The MAP score of the domain-adaptive BERT_{AIoT} is similar to the corresponding prototype. As can be seen from Figure 6, the proportion of correct answers in the top-ranked positions is improved by RoBERTa_{AIoT}.

¹<https://docs.qq.com/pdf/DZFZaRFNncIFSaUVo>

Fig. 5: Probing ranking on the QuoraQA dataset

Fig. 7: Probing ranking on the SuperUserQA dataset

TABLE VI: Probing results on the StackOverflowQA dataset

Method	MAP	MRR@10	MRR@5	MRR@1
BERT	0.3096	0.2852	0.2688	0.1882
RoBERTa	0.3068	0.2814	0.2657	0.1882
ALBERT	0.3759	0.3625	0.3365	0.2235
ELECTRA	0.2369	0.2094	0.1867	0.1176
GPT-2	0.1622	0.1238	0.1080	0.0588
BERT_{AIoT}	0.2989	0.2755	0.2469	0.1882
RoBERTa_{AIoT}	0.3405	0.3198	0.2890	0.2118

Fig. 6: Probing ranking on the StackOverflowQA dataset

TABLE VII: Probing results on the SuperUserQA dataset

Method	MAP	MRR@10	MRR@5	MRR@1
BERT	0.4418	0.4257	0.4141	0.3077
RoBERTa	0.1632	0.1252	0.1056	0.0615
ALBERT	0.4126	0.3908	0.3792	0.2923
ELECTRA	0.166	0.1327	0.1174	0.0308
GPT-2	0.1602	0.1256	0.0953	0.0615
BERT_{AIoT}	0.2377	0.2115	0.1867	0.1077
RoBERTa_{AIoT}	0.3598	0.3427	0.3169	0.2154

As shown in Table VII, ALBERT achieves the best result. Compared with the corresponding prototype, the domain-adaptive RoBERTa_{AIoT} also improves the MAP score by 19.66%. As can be seen from Figure 7, the proportion of correct answers in the top-ranked positions is improved by RoBERTa_{AIoT}. The above results show that our method injects more AIoT domain knowledge into the model, so that the model can better understand the AIoT context before task-specific fine-tuning.

G. Discussion

The connotation of knowledge includes a wide range of content, and both plain text and knowledge base contain knowledge. This paper mainly studies the knowledge derived from the text. The theoretical contribution of this paper is to prove that it is feasible to expand RoBERTa by pre-training on the AIoT corpus. This has important theoretical significance for the language understanding of AIoT domain.

To the best of our knowledge, there is no research on expanding the language model to AIoT domain before. The advantage of using this model is that compared with the traditional deep learning method, the PLM-based method has been proven to improve the results. Because the PLM has learned extensive knowledge on a large-scale corpus in advance, and transformed this knowledge into hidden parameters so that the model can get a good initial state. These models only need to be fine-tuned on downstream tasks to obtain good results. A model that has not been pre-trained on the AIoT corpus can only judge whether it is the correct answer based on the context learned from the general domain.

The limitation of the QA model is that if there are relatively few or no questions about a specific topic, then only context matching may not be able to obtain a satisfactory answer. For example, some questions ask about the application of AIoT in the agricultural field, and the correct answer to these questions do not score very high.

Compared with the methods using the knowledge base, our method is more versatile. The AIoT domain lacks a high-quality AIoT domain knowledge base, so this paper directly learns knowledge from the domain text. We do not need to build a knowledge graph for this task. The disadvantage is that the understanding of concepts and entities is still insufficient. The method of using the knowledge base can inject more knowledge, but the construction of the domain knowledge base is a challenging problem.

Applying this method to the passages of books and scientific works is a promising direction. This can further improve the quality of the dataset, but labeling this type of dataset is labor-intensive and time-consuming. Therefore, in this paper, we use web text. We can further explore the labeling of the literature corpus in future work.

Using PLMs allows the model to learn a lot of task-agnostic features from the corpus to initialize the model to better hidden parameters, and internalize the knowledge in the hidden parameters. This is like a human learning process, first learn some basic subjects, and then learn specific professional topics based on this. Let the model learn some general knowledge first, and then fine-tune it for specific tasks, which has better performance than a model without general knowledge.

Based on the transformer's self-attention mechanism, each word in the question interacts with the words in the passage to determine the result of semantic matching. This kind of matching is a multi-view matching, based on the multi-head attention, which helps to find the connection between words from multiple perspectives. Each word also attends to the words in the context to obtain the contextual representation.

We ran the program 3 times and take the median, which can show that the experimental results are improved and representative. Due to the limitation of computing resources, we did not run too many experiments. We trained a total of 126 large PLMs (including BERT, RoBERTa, ELECTRA, ALBERT, GPT-2, BERT_{Alot}, RoBERTa_{Alot}) on AIoT QA datasets. First, we need to train 7 (architectures) $\times$ 3 (times) large PLMs on each dataset. Then, for probing experiments, we also train another 63 large PLMs. In addition, there are pre-training experiments on BERT and RoBERTa.

V. CONCLUSION AND FUTURE WORK

This paper proposes a pre-training method that can expand PLMs to the AIoT domain to perform QA tasks. Specifically, we build an AIoT domain corpus to further pre-train RoBERTa and BERT. Then, we construct 3 AIoT QA datasets to fine-tune and evaluate RoBERTa_{Alot} and BERT_{Alot}. Experimental results show that further pre-training of RoBERTa on the AIoT corpus will improve its ability in answering AIoT questions on the quora.com, superuser.com and stackoverflow.com websites.

This article mainly solves the problem of how to use the unstructured corpus to expand PLMs to the AIoT domain. The innovation of the proposed method lies in the way we use the domain-specific corpus to further pre-train RoBERTa_{Alot} and BERT_{Alot} to incorporate more AIoT domain knowledge. In the future, we plan to build a large-scale QA dataset in the AIoT domain.

REFERENCES

- [1] I. Ahmad, S. Shahabuddin, H. Malik, E. Harjula, T. Leppänen, L. Lovén, A. Anttonen, A. H. Sodhro, M. M. Alam, M. Juntti *et al.*, "Machine Learning Meets Communication Networks: Current Trends and Future Challenges," *IEEE Access*, vol. 8, pp. 223 418–223 460, 2020.
- [2] Y. Lin, X. Jin, J. Chen, A. H. Sodhro, and Z. Pan, "An analytic computation-driven algorithm for Decentralized Multicore Systems," *Future Generation Computer Systems*, vol. 96, pp. 101–110, 2019.
- [3] A. H. Sodhro, S. Pirbhulal, and V. H. C. De Albuquerque, "Artificial intelligence-driven mechanism for edge computing-based industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4235–4243, 2019.
- [4] A. K. Sangaiah *et al.*, "Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4189–4196, 2019.
- [5] T. Zhang, A. H. Sodhro, Z. Luo, N. Zahid, M. W. Nawaz, S. Pirbhulal, and M. Muzammal, "A joint deep learning and internet of medical things driven framework for elderly patients," *IEEE Access*, vol. 8, pp. 75 822–75 832, 2020.
- [6] A. H. Sodhro, N. Zahid, L. Wang, S. Pirbhulal, Y. O. Ouzrout, A. Sekhari, A. V. L. Neto, A. R. L. De Macedo, and V. H. C. De Albuquerque, "Towards ML-based Energy-Efficient Mechanism for 6G Enabled Industrial Network in Box Systems," *IEEE Transactions on Industrial Informatics*, 2020.
- [7] Y. J. Fan, Y. H. Yin, L. Da Xu, Y. Zeng, and F. Wu, "Iot-based smart rehabilitation system," *IEEE transactions on industrial informatics*, vol. 10, no. 2, pp. 1568–1577, 2014.
- [8] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.
- [9] S. Li, G. Oikonomou, T. Tryfonas, T. M. Chen, and L. Da Xu, "A distributed consensus algorithm for decision making in service-oriented internet of things," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1461–1468, 2014.
- [10] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, "Automatic fruit classification using deep learning for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1027–1034, 2019.
- [11] R. Nogueira and K. Cho, "Passage Re-ranking with BERT," *arXiv preprint arXiv:1901.04085*, 2019.
- [12] E. Aktolga, J. Allan, and D. A. Smith, "Passage reranking for question answering using syntactic structures and answer types," in *European Conference on Information Retrieval*. Springer, 2011, pp. 617–628.
- [13] H. Cui, R. Sun, K. Li, M.-Y. Kan, and T.-S. Chua, "Question answering passage retrieval using dependency relations," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005, pp. 400–407.
- [14] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of NAACL-HLT*. Association for Computational Linguistics, 2019, pp. 4171–4186.
- [15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *CoRR*, vol. abs/1907.11692, 2019.
- [16] K. Clark, M. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators," in *Proceedings of ICLR*. OpenReview.net, 2020.
- [17] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Proceedings of NIPS*, 2020.
- [18] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," in *Proceedings of ICLR*, 2019.
- [19] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep Contextualized Word Representations," in *Proceedings of NAACL-HLT*. Association for Computational Linguistics, 2018, pp. 2227–2237.
- [20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of NIPS*, 2017.
- [21] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [22] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," in *Proceedings of ICLR*. OpenReview.net, 2020.
- [23] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kociský, and P. Blunsom, "Reasoning about entailment with neural attention," in *Proceedings of ICLR*, 2016.
- [24] A. Severn and A. Moschitti, "Modeling relational information in question-answer pairs with convolutional neural networks," *arXiv preprint arXiv:1604.01178*, 2016.
- [25] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "ABCNN: Attention-based convolutional neural network for modeling sentence pairs," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 259–272, 2016.
- [26] A. H. Miller, A. Fisch, J. Dodge, A. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," pp. 1400–1409, 2016.
- [27] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," in *Proceedings of IJCAI*. ijcai.org, 2017, pp. 4144–4150.
- [28] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," in *Proceedings of ACL*, 2018.
- [29] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of ICLR*. OpenReview.net, 2019.
- [30] S. Rongali, A. Jagannatha, B. P. S. Rawat, and H. Yu, "Improved Pretraining for Domain-specific Contextual Embedding Models," *CoRR*, vol. abs/2004.02288, 2020.
- [31] I. Vulic, E. M. Ponti, R. Litschko, G. Glavas, and A. Korhonen, "Probing pretrained language models for lexical semantics," in *Proceedings of EMNLP*, 2020, pp. 7222–7240.