Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7088126, 14 pages
https://doi.org/10.1155/2022/7088126

Research Article

Machine Reading Comprehension-Enabled Public Service Information System: A Large-Scale Dataset and Neural Network Models

Changchang Zeng,[1,2] **Shaobo Li**,[3] **and Bin Chen**[4]

[1]*Chengdu Institute of Computer Application, Chinese Academy of Sciences, Chengdu 610041, China*
[2]*University of Chinese Academy of Sciences, Beijing 100049, China*
[3]*State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China*
[4]*Chongqing Research Institute of Harbin Institute of Technology, Chongqing 401151, China*

Correspondence should be addressed to Shaobo Li; lishaobo@gzu.edu.cn

Received 4 January 2022; Revised 18 January 2022; Accepted 26 January 2022; Published 23 February 2022

Academic Editor: Mohammad Farukh Hashmi

With the rapid development of 5G and machine learning, various innovative information systems continue to emerge. Among them, the public service information system (PSIS) has attracted extensive attention from researchers. However, the existing public service information system can only handle simple tasks and cannot answer complex questions raised by citizens. In this paper, we explore the application of machine reading comprehension to the field of Chinese public service information systems from scratch. Machine reading comprehension is a new research hotspot in the field of machine learning, which has great application potential in PSIS systems. However, work in this area is still lacking, with neither large scale datasets nor neural network models available. To address the above issue, first, we create a large scale machine reading comprehension dataset of Chinese public service information, including public service affair guidelines, policies, etc. Next, we propose several new neural network models which were continually pretrained on the Chinese public service corpus. The experimental results show that the proposed models achieve a significant improvement over several previous SOTA models on the new dataset. However, they are still far below human performance, indicating that the proposed dataset is challenging.

1. Introduction

In recent decades, governments across the globe have been facing numerous challenges in improving the efficiency and equity of public services. The growing pressure of public services has led to the government's increasing attention to novel information technology such as machine learning. One of the main goals of public service information systems (PSIS) is to ensure that citizens can easily and effectively obtain the required public service information. To help bridging the information gap between citizens and governments, many novel information technologies could be applied to the PSIS systems. Machine Reading Comprehension (MRC) is a challenging research hotspot in Artificial Intelligence (AI), which aims to teach machines to understand text and answer questions about the text like humans [1, 2].

Machine reading comprehension can be applied in various PSIS systems, including question answering bot (QABot), search engines, and dialogue systems, and it is possible to change the way of interaction between citizens and the government. As shown in Figure 1, the existing Chinese government QA systems still use the search engine to find the relevant text from the structured knowledge base or semistructured data source, rather than giving the answer to the question directly.

Compared with the existing question answering technology, the machine reading comprehension model can directly extract predictive answers from unstructured text [3]. Therefore, it is obvious that the application of machine

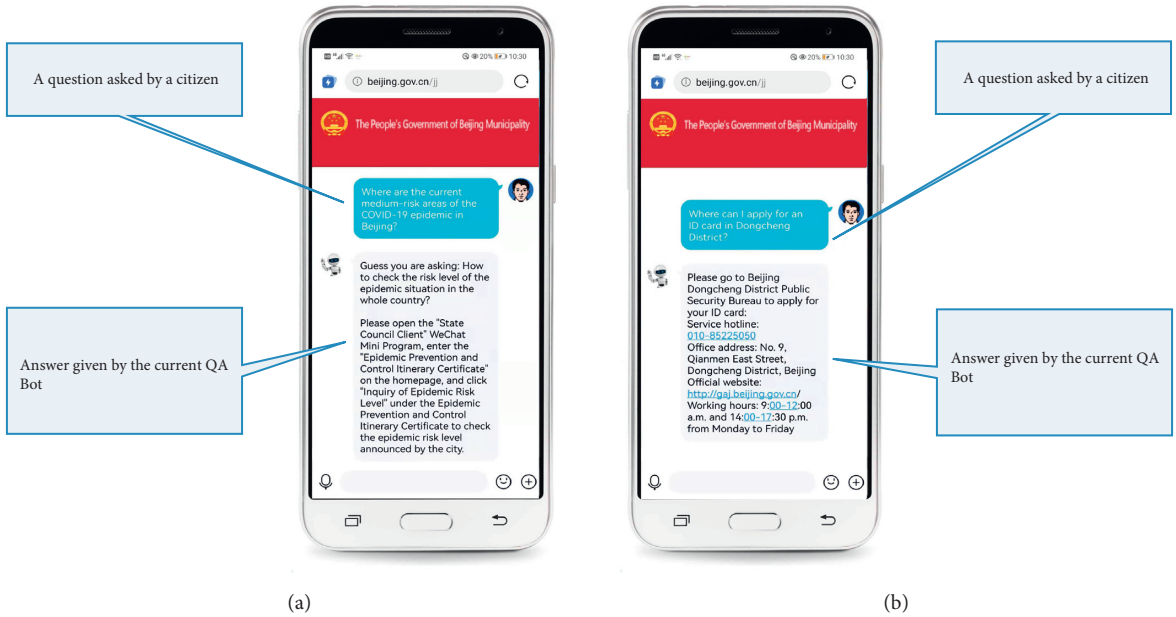

Figure 1: Real examples from two online public service QA bots on the official website of the government of Beijing. (a) An example from the QA bot for epidemic prevention and control. (b) An example from the QA bot for government service consulting. We can see that the above public service QA bot can only give information related to the answer rather than directly giving the correct answer (note that the original text in the figure is Chinese and has been translated into English).

reading comprehension (MRC) technology in PSIS systems will provide many advantages for public services. MRC could greatly improve the efficiency of people to obtain the public service information they need. However, although machine reading comprehension (MRC) technology offers great potential for implementation, work in this area is still lacking. High quality dataset and efficient algorithms are known as the key factors to determine whether the MRC based PSIS can meet the expectations of governments and citizens. But in the public services domain, large-scale MRC datasets are still extremely scarce, and there is no benchmark or leaderboard to evaluate and compare the performance of neural network models.

In this article, we explore the application of machine reading comprehension to the Chinese public service from scratch. Our contributions can be summarized as follow:

(i) We created a new machine reading comprehension dataset of Chinese public service information. To the best of our knowledge, it is the first machine reading comprehension dataset in this field.

(ii) To evaluate the new dataset, we applied several previous SOTA models on the baseline. However, directly applying these models often yields suboptimal results due to the difference of word distribution between general domain corpus and public service corpus, which is especially obvious in Chinese.

(iii) We propose a new position encoding method named exponential joint relative position encoding, which uses both additive position encoding and multiplicative position encoding and also introduces a training weight.

(iv) We proposed the first domain-specific language models for Chinese public service, which were continually pretrained on our own corpus. The experiment results show that the new models consistently outperform the previous SOTA models, but there is still a big gap compared with human performance, indicating that the proposed dataset is challenging.

2. Related Works

Recently, language model pretrained on large unsupervised corpora has greatly improved the performance of many NLP tasks, especially in the field of machine reading comprehension (MRC).

Following the pioneer pretrained models such as OpenAI GPT [4], ELMo [5], and BERT [6], many endeavors have been made for further improvement. To name just a few, ALBERT [7] presents two optimizations to increase the training speed and lower the memory consumption of BERT. SpanBERT [8] proposes a randomly spanned masking strategy instead of word masking. BART [9] is an autoencoder for pretraining seq-to-seq models. Omni-perception Pretrainer (OPT) [10] is the first pretrained model that connects the three modalities of text, vision, and audio, and it is proposed for cross-modal understanding and generation.

However, these models are all pretrained on general-domain corpus such as Wikipedia, thus cannot well adapt the word distribution in specific fields. Therefore, several domain-specific pretrained models have been proposed recently, such as E-BERT [11], FinBERT [12], BioBERT [13],

and SciBERT [14]. They pretrain the model on the domain-specific corpus to obtain specific language knowledge and improve the performance of many NLP tasks.

Obviously, the word distribution of the Chinese public service corpus is very different from that of the general corpus, so it is necessary to pretrain the model in this specific field. However, to the best of our knowledge, there is no pretrained model in the field of machine reading comprehension for public services information, which limits the performance of NLP tasks in this field. To this end, we propose six pretrained models in this field. In addition, different from the above pretraining method, we have adopted a three-stage pretraining strategy [15]. Experiment shows that using a simple continuous pretraining strategy is an effective alternative in the MRC task for Chinese public services information.

3. Task Definition and Evaluation Metrics

3.1. Definition of C-Pulse Task. In recent years, with the rapid development of the machine reading comprehension (MRC) field, a variety of MRC tasks have been proposed. To make up for the lack of MRC dataset in the public service field, we created a new machine reading comprehension dataset of Chinese public service information. Figure 2 shows an example of the proposed dataset.

3.2. Evaluation Metrics. The evaluation metrics we used are Exact Match and $F1$-Score.

3.2.1. Exact Match. The Exact Match can then be calculated as follows:

$$\text{Exact Match} = \frac{M}{N}, \tag{1}$$

where N represents the number of questions, and M represents the number of questions answered correctly.

3.2.2. F1. $F1$ is a commonly used MRC task evaluation metrics. The equation of $F1$ for a single question is as follows:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{2}$$

where Precision denotes the token-level Precision for a single question and Recall denotes the Recall for a single question.

The Precision and Recall are composed of four different variables, including true positive (TP), false positive (FP), true negative (TN), and false negative (FN), as shown in Figure 3.

The Precision for a single question is computed as follows:

$$\text{Precision} = \frac{\text{Num}(\text{TP})}{\text{Num}(\text{TP}) + \text{Num}(\text{FP})}. \tag{3}$$

The Recall for a single answer is computed as follows:

$$\text{Recall} = \frac{\text{Num}(\text{TP})}{\text{Num}(\text{TP}) + \text{Num}(\text{FN})}. \tag{4}$$

4. Dataset Construction and Description

In this section, we describe how we construct C-Pulse and provide detailed statistics and analysis of this dataset. Figure 4 shows the construction process of our dataset.

4.1. Data Collection and Cleaning. As shown in Figure 4, we used a python crawler to collect the original data from a variety of government website in China, such as the Chinese central government website, the State Council's public affairs question answering platform, the provincial governments' public service websites, and the National Health Commission website. The data cover a wide range of forms, including policies, notices, regulations, news, and guidelines written by government officials or official answers to questions raised by citizens. The data types are also very diverse, such as economy, education, transportation, medical care, industry, commerce, taxation, environment, resources, etc.

Then, we perform the following steps to clean the original data. First, we removed the HTML template from the crawl report. Next, we filtered out articles and questions that contained images or tables. Third, the special punctuation and other symbols have been deleted. Fourth, we carefully removed that cumbersome and complicated legal information from the dataset. Finally, we selected these samples with moderate paragraph length. We use Python to remove those too long and too short samples. After removing, the shortest sample contains 115 tokens and the longest sample contains 900 tokens. Finally, these samples will be used as the original data of the C-Pulse dataset we are about to build. In order to prevent the repeated use of these selected data in the process of model pretraining, we delete these data from the corpus.

4.2. Real Question and Answering Records. The quality of the dataset is the key to effectively pretraining and fine-tuning of the language model. There are three prevalent methods to create new datasets: crowdsourcing, expert assistance, and automatic generation. Among them, crowdsourcing usually refers to hiring annotators on the website such as Amazon Mechanical Turk to perform text generation or text annotation. The expert method usually refers to hiring experts in a specific field to complete the generation of the dataset. The automatic method refers to the use of various NLP tools or programs to automatically generate datasets.

Although the two methods of crowdsourcing and experts assistance can get more accurate and natural datasets, they both require a lot of funds to hire annotators. Moreover, in the field of public services studied in this article, because ordinary crowd workers are not really needed to obtain the public service, their questions or answers raised are often not able to hit the point.

Fortunately, we found some real question and answering (QA) records on the government service websites. These

Title:
The Latest Entry-and-Exit Control Measures of Beijing
Context:
People entering (returning to) Beijing must present a negative nucleic acid test result obtained within 48 hours prior to the entrance and show the green code status on the "Beijing Health Kit". People with a travel or residence history in a county (county-level city, district, banner) where one or more confirmed local cases are found within 14 days, are strictly restricted from entering (returning to) Beijing. Commuters from neighboring areas who enter (return to) Beijing for the first time after the new measure takes effect, are required to present a negative nucleic acid test result obtained within 48 hours prior to the entry, and for each subsequent entry, a negative result of nucleic acid test taken within 14 days is sufficient.
Question:
When entering (returning) to Beijing for the first time, commuters in the suburbs of Beijing are required to present a negative nucleic acid test result obtained within how many hours?
Answer:
48 hours

Figure 2: An example of the proposed machine reading comprehension dataset (note that the original text in the figure is Chinese and has been translated into English).

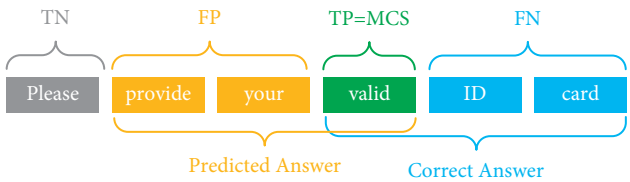

Figure 3: The true positive (TP), false positive (FP), true negative (TN), and false negative (FN).

questions are asked by those citizens who really need access to the public service, and they are answered by the officials in charge of the matter and released publicly.

Therefore, the data quality of these real question and answer records is better than the quality of the text generated by the crowd worker experts.

In addition, since this government service website is an official website, government officials will also review the QA records. As the website has normally been operating for many years and has served many citizens, we have reason to believe that the information on this website has been checked by many officials.

Moreover, in the process of constructing our MRC dataset, we also conducted checks on the answers to the questions, and the results of our checks also confirmed that these answers are reliable.

4.3. Dataset Construction. Our original data comes from the public service information on the official website of the Chinese government.

Generally, we save every public service item in a paragraph with multiple short passages.

Since not every public service item has a corresponding real QA record, we divide the collected paragraphs into two parts, one is paragraphs that contain real QA records, and the second is paragraphs that do not contain real QA records. For those paragraphs that do not contain real QA records, we generated questions and answers for them. To save time, we first manually edit some questions and answers and then generate synonymous sentences.

Then, we reorganized the paragraphs. We put the question at the beginning of each paragraph and the answer at the end of the paragraph.

Third, in order to prevent the paragraph structure from being too singular, we randomly deleted some passages from the paragraph and inserted the answers into the paragraph randomly. Since our answer is a complete sentence or paragraph, and our passages are independent of each other, inserting answers randomly between passages will not cause grammatical errors.

Fourth, we check the grammar of these paragraphs. If a grammatical error is found in a paragraph, we delete the paragraph.

Finally, we get about 20,000 question-answer pairs and 11,346 Chinese public service paragraphs, and we randomly divide them into a training set, development set, and test set according to the ratio of 8 : 1 : 1. The dataset is saved in the form of JSON files.

4.4. Dataset Analysis. To understand the properties of the C-Pulse dataset, we analyze the questions and answers in this dataset. Specifically, we explore (1) the statistics of the size of final data, (2) types of questions, (3) the types of answers, (4) the classification of public service information, and (5) the generation methods of C-Pulse. Table 1 shows the statistics of the data size and the number of tokens in context, question, and answer.

Next, we analyze the types of questions and types of answers of the dataset.

4.4.1. Question Types. The interrogative words in modern Chinese are diversified. There are interrogative pronouns and interrogative modal particles in Chinese.

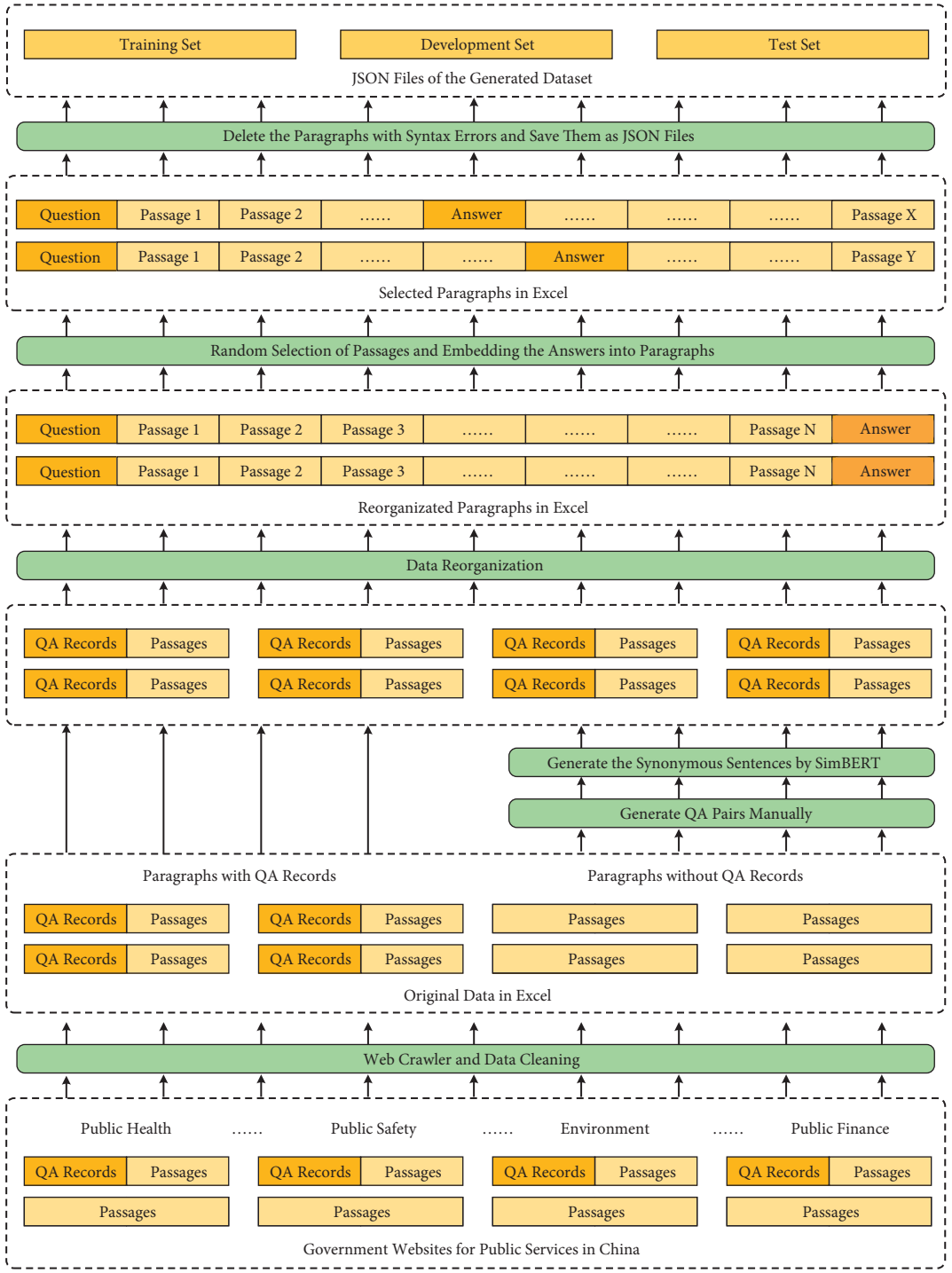

Figure 4: Construction process of the C-Pulse dataset.

To identify the question type, first, we follow the concept of the central question word (CQW) [16] and divide our questions into the following categories: "Which," "Who," "Where," "Why," "When," "Whether," "What" (include what time, what date, what for, what color, what about), and "How" (include How does/do, how many, how much, how far, how often), etc.

Then, we randomly select 200 questions from the dataset for manual classification and provide the distribution of question types in Figure 5.

Table 1: The statistics of the C-Pulse dataset.

	Train set	Dev set	Test set
Context #	8,845	1,249	1,249
Question #	16,000	2,000	2,000
Max context tokens #	897	900	899
Avg context tokens #	373	407	415
Min context tokens #	130	115	123
Max question tokens #	97	109	77
Avg question tokens #	24	23	23
Min question tokens #	5	7	7
Max answer tokens #	487	401	682
Avg answer tokens #	30	31	31
Min answer tokens #	2	2	2

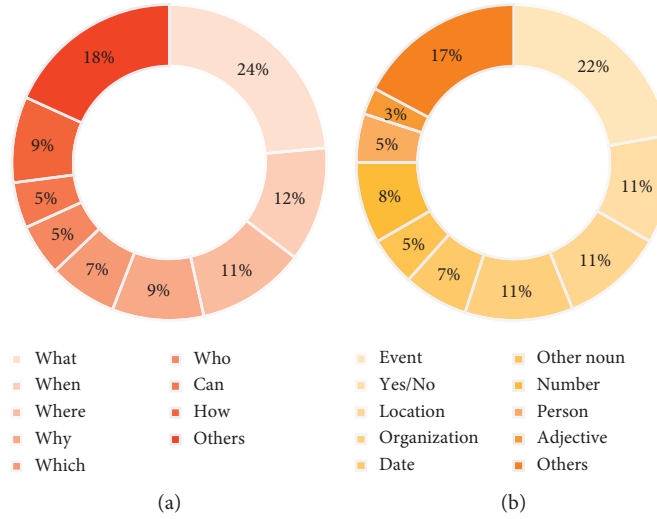

Figure 5: The distributions of question types and answer types in the C-Pulse dataset. (a) The question types. (b) The answer types.

4.4.2. Answer Types. As shown in Figure 5, we randomly take out 200 answers from the dataset and then manually classify them according to the answer type. We find that C-Pulse covers a wide range of answer types.

4.4.3. Data Diversity and Generation Methods. Finally, the data diversity and the generation methods of our dataset are analyzed. We further sample 200 examples and visualize their distributions in Figure 6.

4.5. Dataset Comparison. The statistics of the proposed dataset have been given in the previous section. In this section, we compare the C-Pulse dataset with the other MRC datasets.

The comparison of the number of questions is shown in Table 2. In contrast to prior MRC datasets, the question size of the C-Pulse dataset is at a medium level.

Next, the statistics of the context size are given in Table 3. As we can see, the C-Pulse dataset contains 11,346 contexts, including 8,845 passages in the train set, 1,250 passages in the development set, and 1,250 passages in the test set. In contrast to prior MRC datasets, the context size of the C-Pulse dataset is also at a medium level.

We also compared the question style, answer style, source of corpora, and generation method of each dataset. Before the comparison, we have summarized the four dimensions of the MRC tasks, including "Answer Source," "Question Type," "Answer Type," and "Type of Corpus." Each of these dimensions is divided into several different categories.

According to our classification method, the "Answer Source" of the C-Pulse task belongs to the "Span Extraction" type. That is, the answer should be a textual span that is directly extracted from the passage. The "Type of Questions" of C-Pulse belongs to "Natural Form," which is much more natural than the cloze-style MRC datasets, as shown in Table 4.

5. Proposed Models

5.1. The Three-Stage Training Method. In recent years, pretrained language models such as ELMo [5], GPT [4], BERT [6], and ALBERT [7] have achieved significant success in many NLP tasks. These models are often pretrained on general domain corpus such as Wikipedia or news articles. However, directly fine-tuning these pretrained models on domain-specific datasets often get

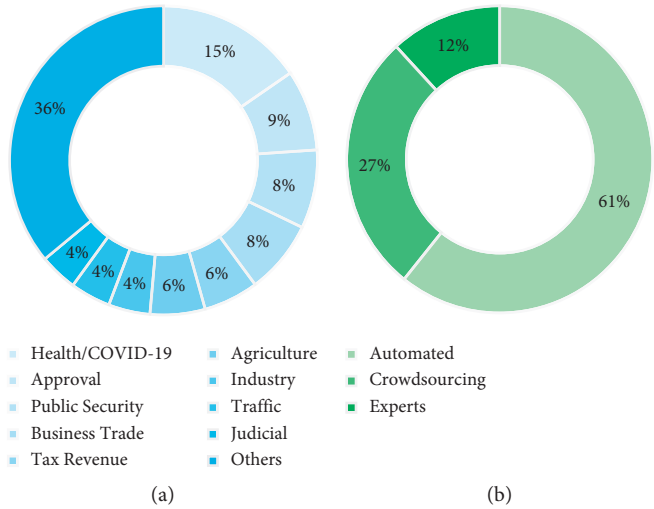

Figure 6: The data diversity and generation methods of the C-Pulse dataset. (a) The data diversity. (b) The generation methods.

Table 2: The number of questions of each MRC dataset.

Dataset	Question #	Train Qu. #	Dev Qu. #	Test Qu. #	Percentage of the train set
SQuAD2.0	151,054	130,319	11,873	8,862	86.27
MCScript	13,939	9,731	1,411	2,797	69.81
TQA	26,260	15,154	5,309	5,797	57.71
SQuAD1.1	107,702	87,599	10,570	9,533	81.33
MovieQA	21,406	14,166	2,844	4,396	66.18
OpenBookQA	5,957	4,957	500	500	83.21
DREAM	10,197	6,116	2,040	2,041	59.98
WikiQA	3,047	2,118	296	633	69.51
ARC-E	5,197	2,251	570	2,376	43.31
ProPara	488	391	54	43	80.12
ARC-C	2,590	1,119	299	1,172	43.20
C-Pulse	20,000	16,000	2,000	2,000	80.00

Table 3: The number of contexts of each MRC dataset.

Dataset	Context #	Train co. #	Dev co. #	Test co. #	Unit of context
CLOTH	7,131	5,513	805	813	Passages
CoQA	8,399	7,199	500	700	Passages
Qangaroo-W	51,318	43,738	5,129	2,451	Passages
Qangaroo-M	2,508	1,620	342	546	Passages
MovieQA	548	362	77	109	Movies
TQA	1,076	666	200	210	Lessons
DREAM	6,444	3,869	1,288	1,287	Dialogues
SQuAD2.0	505	442	35	28	Articles
SQuAD1.1	536	442	48	46	Articles
C-Pulse	11,346	8,846	1,250	1,250	Passages

unsatisfactory results due to the difference of word distribution between general domain corpus and public service corpus, which is especially obvious in the Chinese public service domain.

In addition, most of the current pretrained models follow the two-stage training strategy of "pretraining + fine-tuning." But recent studies have shown the benefit of continuous pretraining on domain-specific unlabeled data [11, 12, 15].

Therefore, based on previous works [11–13, 17–19], we implement the three-stage training method in our baseline. As shown in Figure 7, the three-stage training method includes general domain corpus pretraining, domain-specific continuous pretraining, and fine-tuning.

However, pretraining the language model from scratch requires humongous amounts of data, computing power, or time. Due to our limited computational resources, in the first stage, the pretrained model from the general-domain corpus is still used. In the second stage, we continue to train these pretrained models using unlabeled corpus in the public service domain. In the third stage, the model is fine-tuned on our C-Pulse dataset. Before the fine-tuning, we adopt a data

Table 4: Analysis of each MRC dataset.

Dataset	Question style	Answer style	Corpora source	Dataset generation method
Facebook CBT	Cloze	Free-form	Children's books	Automated
MovieQA	Natural	Free-form	Movie plot	Crowdsourcing
DREAM	Natural	Free-form	Exam	Crowdsourcing
CliCR	Cloze	Free-form	BMJ case reports	Automated
MCScript	Natural	Free-form	Narrative texts	Crowdsourcing
ARC-C	Natural	Free-form	Science questions	Experts
ARC-E	Natural	Free-form	Science questions	Experts
WikiQA	Natural	Free-form	Wikipedia	Crowdsourcing
TQA	Natural	Free-form	Science curricula	Experts
ProPara	Natural	Spans	Process paragraph	Crowdsourcing
SQUAD2.0	Natural	Spans	Wikipedia	Crowdsourcing
SQuAD1.1	Natural	Spans	Wikipedia	Crowdsourcing
Qangaroo-W	Synthesis	Spans	Paper abstracts	Crowdsourcing
Qangaroo-M	Synthesis	Spans	Wikipedia	Crowdsourcing
OpenBookQA	Natural	Free-form	Elementary level science facts	Crowdsourcing
C-Pulse	Natural	Spans	Public service information	All three methods

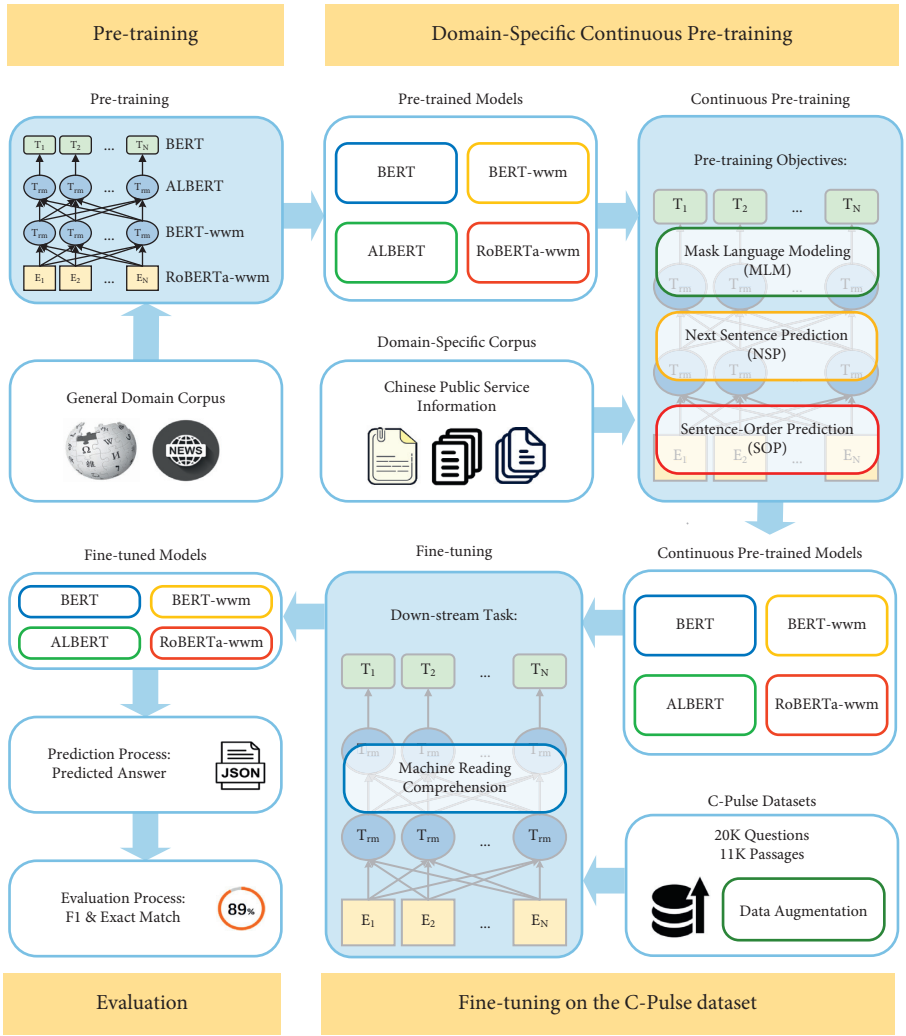

Figure 7: The architecture of the three-stage training method and evaluation.

augmentation method to increase the training data. In the final stage, the fine-tuned models are used to predict the answers to the questions in the test set of C-Pulse. Then, we conduct the evaluation process by comparing the predicted answer with the correct answer. We use Exact Match and *F*1-Score as the evaluation metrics.

5.2. Overview of Proposed Models. Then, in order to learn domain-specific knowledge, we continue the pretraining of the above models on the Chinese public service corpus and get the following domain-specific pretrained models:

(i) CP-BERT: the domain-specific BERT model pretrained on Chinese public service corpus

(ii) CP-BERT-wwm: the domain-specific BERT-wwm model pretrained on Chinese public service corpus

(iii) CP-ALBERT (base) and CP-ALBERT (large): the domain-specific ALBERT models pretrained on Chinese public service corpus

(iv) CP-RoBERTa-wwm (base) and CP-RoBERTa-wwm (large): the domain-specific RoBERTa-wwm models pretrained on Chinese public service corpus

The "CP" stands for the continually pretrained models on the Chinese public service corpus. The details of the continuous pretraining will be discussed in the next subsections.

The differences between the pretrained model in this article and the existing pretrained models are as follows:

(1) Most existing pretrained language models are trained on general domain corpus such as Wikipedia. However, for machine reading comprehension of Chinese public service information, directly applying these general pretrained models often yields suboptimal results due to the difference of word distribution between general domain corpus and public service corpus, which is especially obvious in Chinese. Therefore, in this paper, we use the Chinese public service corpus to train the domain-specific models.

(2) Most of the current pretrained models follow the two-stage training strategy of "pretraining + fine-tuning." But recent studies have shown the benefit of continuous pretraining on domain-specific unlabeled data. Therefore, based on previous works, we implement the three-stage training method in our baseline. The three-stage training method includes general domain corpus pretraining, domain-specific continuous pretraining, and fine-tuning. We use different versions of BERT, ALBERT, and RoBERTa to initialize the weight parameters.

(3) Most domain-specific pretrained models just directly use the structure and pretraining objectives of the BERT. In this article, we have trained a variety of advanced domain-specific pretrained models, including CP-BERT, CP-ALBERT, CP-RoBERTa-wwm, and CP-BERT-wwm. Therefore, our model structure includes some components that are different from BERT, such as the whole word masking used in RoBERTa-wwm and BERT-wwm, and the parameter sharing used in ALBERT. At the same time, our pretraining objectives are also diversified. Among them, the pretraining objective of ALBERT are SOP and MLM, BERT and BERT-wwm are MLM and NSP, and the pretraining objective of RoBERTa-wwm is MLM.

(4) Position encoding represents the relationship between elements at different positions in the input sequence. Different from the existing position encoding method [20–24], we propose a new position encoding method named exponential joint relative position encoding, which uses both additive position encoding and multiplicative position encoding and also introduces a training weight. After using our exponential relative position encoding, the inner product result of Q and K in the Attention operation and the relative position show a negative exponential relationship. That is, as the position distance increases, the mutual influence is smaller. That is, the longer the distance, the weaker the connection between words.

5.3. Details of Proposed Models

5.3.1. Domain-Specific Corpus. The pretraining corpus we use mainly includes the following five types of texts in public service fields:

(i) QA pairs: question and answering records on public service websites

(ii) Policies: public policies on the websites of the central government

(iii) Official news: public service-related news from official media

(iv) Statistical information: statistical information on public service matters of government departments

(v) Health information: public health and health service information during the COVID-19 pandemic

For the above five types of corpus, under the guidance of professionals in the public service field, we screen and preprocess the important parts of various corpora to obtain the final corpus.

5.3.2. Exponential Joint Relative Position Encoding. Our exponential joint relative position encoding uses both additive position encoding and multiplicative position encoding and also introduces training weight, where the additive position encoding part adopts the T5 style [22]:

$$Q_m K_n^{\intercal} = x_m W_Q W_K^{\intercal} x_n^{\intercal} + \beta_{m,n}. \tag{5}$$

Among them, $\boldsymbol{\beta}_{m,n}$ is a trainable weight parameter, and $\mathbf{x}_m \mathbf{W}_Q \mathbf{W}_K^T \mathbf{x}_n^T$ is the inner product between K_n and Q_m in the Attention operation. K_n and Q_m are the components of the K and Q vectors at the m and n positions after the proposed relative position encoding is added. Therefore, the additive position encoding of the T5 style used in this paper is essentially adding a trainable bias term $\boldsymbol{\beta}_{m,n}$ on the basis of the Attention matrix.

We also use multiplicative relative position encoding, which is different from the existing multiplicative position encoding. The method we use is to multiply the word vectors K_n and Q_m by exponential coefficients. First of all, the K_n and Q_m are, respectively,

$$Q_m = x_m W_Q = (q_0, \ldots, q_l),$$
$$K_n = x_n W_k = (k_0, \ldots, k_l). \tag{6}$$

Among them, l is the length of the vector. In this article, the multiplicative exponential-level relative position encoding we use is essential to multiply the components at different positions in the vectors K_n and Q_m (just take the positions m and n as examples) and multiply them by the exponential weight. After weighting, the result value is an exponential function of $m - n$.

The formulas for the components K_n and Q_m after the exponentially weighted are as follows:

$$f(Q_m) = R_m Q_m = \begin{pmatrix} e^m & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^m \end{pmatrix} \begin{pmatrix} q_0 \\ \vdots \\ q_l \end{pmatrix} = \begin{pmatrix} q_0 e^m \\ \vdots \\ q_l e^m \end{pmatrix},$$
$$f(K_n) = R_n K_n = \begin{pmatrix} e^{-n} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{-n} \end{pmatrix} \begin{pmatrix} k_0 \\ \vdots \\ k_l \end{pmatrix} = \begin{pmatrix} k_0 e^{-n} \\ \vdots \\ k_l e^{-n} \end{pmatrix}. \tag{7}$$

Then, we do the Attention inner product of K_n and Q_m after exponentially weighting and then find the negative exponent of the absolute value of the position difference to obtain the relative position encoding function of the exponential order. The result of the calculation is as follows:

$$g(f(Q_m), f(K_n)) = \sum_{i=0}^{l} k_i q_i e^{-|m-n|} = Q_m K_n^{\intercal} e^{-|m-n|}. \tag{8}$$

It can be seen from the above formula that the result of the Attention inner product of the weighted K_n and Q_m we calculated is an exponential function of $m - n$.

Therefore, using exponential relative position encoding, the result of the Attention operation and the relative position $m - n$ present a negative exponential relationship. That is, as the position distance increases, the mutual influence becomes smaller. This also conforms to our intuitive conjecture that the further the distance, the weaker the connection between words.

Finally, we combine the above-mentioned additive relative position encoding and multiplicative relative position encoding, that is to say, on the basis of the multiplicative exponential relative position encoding, add a trainable bias term $\boldsymbol{\beta}_{m,n}$. Therefore, the final Attention calculation formula of our relative position encoding function is as follows:

$$p(Q_m, K_n) = g(f(Q_m), f(K_n)) + \beta_{m,n}$$
$$= x_m W_Q W_K^{\intercal} x_n^{\intercal} e^{-|m-n|} + \beta_{m,n}. \tag{9}$$

Among them, $p(Q_m, K_n)$ represents the result of the Attention inner product operation of K_n and Q_m at positions m and n after the exponential joint relative position encoding is added.

5.3.3. Transformer Encoders. This layer is composed of 12 transformer based encoder layers, each having multiple attention heads. Each vector representation is then transformed through the following transformer encoding layers [25]: (1) First, through linear layers, every embedding vector produces three vectors, called the key, query, and value vectors. (2) These three vectors pass through a self-attention head, then output a single vector that represents the whole input embedding, which makes the model context-aware. (3) With multiple attention heads, Each embedding vector go through different linear layers to produce multiple unique triplets of key, query, and value vectors and then go through different self-attention head. (4) The output from multiple attention heads goes through another linear layer and a feed-forward layer. The output is the transformed embedding vectors. After the several transformer encoding layers, the embedding vectors have been transformed to contain enough information about the input sequence.

5.3.4. Pooling Layer. The pooling layer is essentially a fully connected layer that is connected to the last transformer encoding layer. The output of the pooling layer is sent to the pretraining objectives layer for final pretraining. The pretraining objectives are described in detail in the next subsection.

5.3.5. Pretraining Objectives. The pretraining objective plays an essential role in the pretraining stage. As shown in Figure 7, in order to help the language model to learn different levels of semantic information and knowledge from the training corpus, we adopt two pretraining objectives:

(i) Mask Language Modeling (MLM): Mask Language Modeling (MLM) is a fill-in-the-blank task that is widely used in Transformer [25] based pretraining such as BERT [6], where a model receives a passage which is masked partly and learns to predict the masked token(s). MLM is often used to train the models to learn textual patterns from unlabeled text.

(ii) Next Sentence Prediction (NSP): in the NSP pretraining task, the model (such as BERT) is given two sentences as input and then required to predict if the second one is the subsequent sentence to the first sentence [6]. However, some studies have pointed out that NSP tasks can actually be removed. For example, NSP is not used in the pretraining of the RoBERTa and ALBERT.

5.3.6. Weights Initialization. Generally, there are two initialization strategies for the weight parameters of the pretrained model: (1) random initialization; (2) loading the pretraining model. In this paper, in order to make use of the existing pretrained models, we adopt the second method. We use different versions of BERT, ALBERT, and RoBERTa to initialize the weight parameters.

5.4. Data Augmentation. Before fine-tuning, the data augmentation is conducted to enhance the performance. The data augmentation method in this paper is derived from

existing methods [17, 19]. Figure 8 shows the process of data augmentation.

As seen in Figure 8, the training paragraphs are divided into multiple smaller paragraphs, some of which contain correct answers, and there are no answers in the remaining paragraphs. For example, as shown in the figure, the original paragraph is divided into 5 small paragraphs, and the answer only exists in the fourth paragraph, and there are no answers in the remaining paragraphs. The purpose of data augmentation is to create more negative examples. The model needs to learn to judge whether there is an answer in the paragraph. If not, it needs to refuse to answer the question and the given answer should be −1.

5.5. Fine-Tuning. After data augmentation, the models are fine-tuned on the machine reading comprehension task with the C-Pulse dataset.

During fine-tuning, each model was fine-tuned on the C-Pulse train set for 5 epochs with a sequence length of 512 and a batch size of 8, while the learning rate was fixed to $1e-5$. The fine-tuned models of each epoch were all saved until the end of 5 epochs, and then the model with the best performance on the development set was selected as the final fine-tuned model.

The fine-tuning is based on transformers, and two linear layers are added on top of the transformer encoder to compute the probabilities of the start position and end position of the answer span in the context. The training objective is to minimize the softmax cross-entropy between the ground truth positions and the predicted position. We can choose if we want to use all the output of the encoder layer or only the first one (corresponding to the [CLS] token). In this baseline, we "pool" the model by simply taking the hidden state corresponding to the first token (the [CLS] token), which is often sufficient for start and end token classification. The pooled representation of the first token [CLS] is then fed into two full connection layers to predict the answer start and end probability distributions independently with softmax normalization.

Specifically, for start position prediction, after taking the dot product between the pooled representation and the "start" weights in the full connection layer, a softmax is applied to produce a probability distribution over all positions in the context. We take the position with the highest probability as the prediction for the start position of the answer span. For end position prediction, we repeat the same process to predict the end position of the answer span.

Let P denotes the pooled representation output by the final pooler layer, and $\mathbf{W}_s$, $\mathbf{W}_e$ are "start" and "end" weights in the full connection layer with learnable parameters. The probability distributions of start positions for the answer span is as follows:

$$\mathbf{p}_s = \text{softmax}\left(\mathbf{W}_\mathbf{s} \cdot \mathbf{P}\right). \tag{10}$$

The probability distributions of end positions for the answer span is as follows:

$$\mathbf{p}_e = \text{softmax}\left(\mathbf{W}_\mathbf{e} \cdot \mathbf{P}\right). \tag{11}$$

The loss function is the sum of cross-entropy loss for start and end position probabilities. Let T_s and T_e denote the true start position and end position of the answer span.

$$\mathbf{loss} = -\log\left(\mathbf{p}_s\left(\mathbf{T}_s\right)\right) - \log\left(\mathbf{p}_e\left(\mathbf{T}_e\right)\right). \tag{12}$$

5.6. Prediction and Evaluation. Finally, the fine-tuned models are used to predict the answers to the questions in the test set of C-Pulse. The prediction answers are saved in a JSON file. Then, we compared the predicted answers with the correct answers.

6. Experimental Results

6.1. Human Performance. To evaluate human performance, we invited 20 college students to answer questions in the dataset. Each student was assigned 120 questions. Finally, after removing the invalid answers, we got 1,172 and 1,184 answers in the development and test set, respectively. Then we calculated $F1$ and EM to roughly evaluate the human performance on the C-Pulse dataset.

6.2. Existing Models. In order to compare our proposed pretrained models with the existing models, we evaluate the following existing pretrained models on the C-Pulse dataset:

(i) BERT(base): the vanilla Chinese BERT-base model proposed by Devlin et al., and open-sourced by Google, which is pretrained on large-scale general domain corpus such as Chinese Wikipedia [6].

(ii) BERT-wwm (base): the Chinese model published by Cui et al., which has the same configuration of BERT-base except using whole word masking, and is also trained on Chinese general domain corpus [26].

(iii) ALBERT (base) and ALBERT (large): a Lite BERT (ALBERT) model was proposed by Lan et al. It uses 89% fewer parameters than the BERT model and has little accuracy loss [7]. The Chinese ALBERT models are open-sourced by Google and CLUE team. We use the base version with 12 layers and a hidden size of 768, and the large version with 24 layers and a hidden size of 1024. Both are pretrained on the CLUE corpus, which is a Chinese general domain corpus [18].

(iv) RoBERTa-wwm (base) and RoBERTa-wwm (large): the RoBERTa model was proposed by Liu et al. It is based on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger minibatches and learning rates, while dynamically changing the masking pattern [27]. The RoBERTa-wwm models are published by Cui et al., which were trained on Chinese general domain corpus using whole word masking [26].

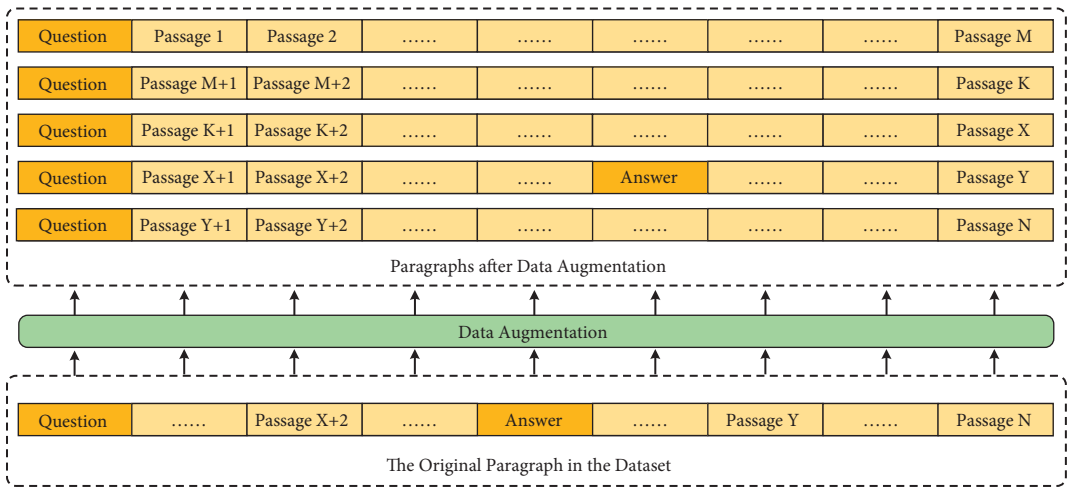

Figure 8: An illustration of the data augmentation.

Table 5: The evaluation results of pretrained models and human performance.

Model	Development		Test	
	*F*1	EM	*F*1	EM
Human performance	92.45	96.13	91.85	96.33
Pretrained on general-domain corpus such as Chinese Wikipedia by Google, etc.				
BERT (base)	76.30	85.34	76.40	85.04
BERT-wwm (base)	76.95	86.42	76.75	85.93
ALBERT (base)	75.85	85.39	76.20	85.26
ALBERT (large)	79.05	88.76	77.70	87.79
RoBERTa-wwm (base)	78.75	86.66	78.30	85.79
RoBERTa-wwm (large)	79.35	88.38	78.60	87.43
Continually pretrained on Chinese public service corpus by us				
Our BERT (base)	81.80	89.44	79.30	87.70
Our BERT-wwm (base)	81.60	89.25	79.65	88.20
Our ALBERT (base)	81.00	89.46	78.40	87.45
Our ALBERT (large)	80.90	89.90	79.60	88.10
Our RoBERTa-wwm (base)	82.65	90.05	79.85	88.30
Our RoBERTa-wwm (large)	83.10	90.79	80.45	88.51

6.3. Results. In this section, we conduct ablation studies and compare the proposed continually pretrained models with existing models. The evaluation metrics are Exact Match and *F*1 score. The performance of different models on the development set and test set are evaluated separately. Table 5 shows our results on the C-Pulse dataset. We compare two kinds of models: (1) No further pretrained models (denoted by BERT, BERT-wwm, ALBERT (base), ALBERT (large), RoBERTa-wwm (base), RoBERTa-wwm (large)), and (2) continually pretrained models (denoted by Our BERT, Our BERT-wwm, Our ALBERT (base), Our ALBERT (large), Our RoBERTa-wwm (base), and Our RoBERTa-wwm (large)).

6.4. Analysis. The first thing we notice is that the best results of the machines are still far below the human performance, which is 7.82 and 11.40 lower on the *F*1 and EM scores in the test set, respectively. This shows that the C-Pulse dataset is challenging and leaves much room for further improvements.

Next, we can clearly see that all the proposed continually pretrained models consistently outperformed their corresponding baseline models. Considering those baseline models used to be the SOTA models on large-scale MRC datasets, the experimental results are encouraging.

Third, we find that the results are better when the model is larger. Specifically, the largest model CP-RoBERTa-wwm (large) achieves better performance than that of all other models.

Finally, we also realize that if the model is larger, the improvement of model performance caused by continuous pretraining will be smaller. For instance, CP-BERT (base) outperformed BERT (base) by 2.66 in terms of *F*1 score and 2.90 in terms of EM. But for RoBERTa-wwm (large), the continually pretrained model only outperformed baseline model by 1.08 in terms of *F*1 score and 1.85 in terms of EM. If we compare the improvement of the base and large versions of ALBERT and RoBERTa-wwm, respectively, we will get the same conclusion. It suggests that although large-scale models have stronger representation and generalization ability, they are more difficult to be fully pretrained on domain-specific corpora.

Overall, this experimental result shows that proposed models can provide significant and stable enhancement and also shows that continually pretrained models have learned domain-specific knowledge from the Chinese public service corpus during the pretraining phase.

Research Limitations: Due to the limited computational resources, we could not conduct exhaustive experiments on more baseline models. Moreover, initial hyperparameters (such as learning rate, training steps, batch size, etc.) are very important to the performance of pretrained models. As there are so many possible combinations of these parameters, our pretrained models may not be optimal.

7. Conclusions

In this paper, we explore the application of machine reading comprehension to the Chinese public service from scratch. First, we propose the C-Pulse dataset, a new benchmark dataset that contains 20,000 question-answer pairs collected from 11,346 paragraphs, including China's public service affairs guidelines, regulations, policies, and government news, covering health/COVID-19, commerce, public security, agriculture, industry, and other fields.

Next, to evaluate the C-Pulse dataset, we applied several previous state-of-the-art pretrained models on the baseline. However, directly applying these models often yields suboptimal results due to the difference of word distribution between general domain corpus and public service corpus, which is especially obvious in Chinese. To this end, we implement the three-stage training method and continue to pretrain the above models on the Chinese public service corpus, and we obtain the CP-BERT, CP-BERT-wwm, CP-RoBERTa-wwm, and CP-ALBERT.

Experiments on the C-Pulse dataset show that the proposed models consistently outperformed their corresponding baseline models, which are pretrained on general domain corpus, and we also find that the performance of the state-of-the-art model is still far behind human beings, indicating that the proposed dataset is challenging and there is still much room for improvement in the future.

Data Availability

The data used to support the findings of this study are available from the first author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] C. Zeng and S. Li, "Analyzing the effect of masking length distribution of mlm: an evaluation framework and case study on Chinese mrc datasets," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 5375334, 17 pages, 2021.

[2] S. Liu, X. Zhang, S. Zhang, H. Wang, and W. Zhang, "Neural machine reading comprehension: methods and trends," *Applied Sciences*, vol. 9, no. 18, p. 3698, 2019.

[3] M. Hu, *Research on machine reading comprehension and textual question answering*, PhD thesis, National University of Defense Technology of China, Changsha, China, 2019.

[4] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pretraining," 2018, https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[5] M. E. Peters, M. Neumann, M. Iyyer et al., "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2227–2237, New Orleans, LA, USA, June 2018.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: pretraining 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*, pp. 4171–4186, Minneapolis, MN, USA, June 2019.

[7] 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 the International Conference on Learning Representations*, Addis Ababa, Ethiopia, April 2020.

[8] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: improving pretraining by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.

[9] M. Lewis, Y. Liu, N. Goyal et al., "Bart: denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, July 2020.

[10] J. Liu, X. Zhu, F. Liu et al., "Opt: omni-perception pre-trainer for cross-modal understanding and generation," 2021, https://arxiv.org/abs/2107.00249.

[11] D. Zhang, Z. Yuan, Y. Liu et al., "E-bert: a phrase and product knowledge enhanced language model for e-commerce," 2020, https://arxiv.org/abs/2009.02835.

[12] Z. Liu, D. Huang, K. Huang, L. Zhuang, and J. Zhao, "Finbert: a pre-trained financial language representation model for financial text mining," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 4513–4519, Yokohama, Japan, July 2020.

[13] J. Lee, W. Yoon, S. Kim et al., "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics (Oxford, England)*, vol. 36, no. 4, pp. 1234–1240, 2020.

[14] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: a pretrained language model for scientific text," 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)*, pp. 3615–3620, Hong Kong, China, November 2019.

[15] S. Gururangan, A. Marasović, S. Swayamdipta et al., "Don't stop pretraining: adapt language models to domains and tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, July 2020.

[16] W. Chen, H. Zha, Z. Chen, W. Xiong, H. Wang, and W. Wang, "HybridQA: a dataset of multi-hop question answering over tabular and textual data," in *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1026–1036, November 2020.

[17] Z. Zhao, H. Chen, J. Zhang et al., "UER: an open-source toolkit for pretraining models," 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): System Demonstrations*, pp. 241–246, Hong Kong, China, November 2019.

[18] L. Xu, H. Hu, X. Zhang et al., "CLUE: a Chinese language understanding evaluation benchmark," in *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 4762–4772, Barcelona, Spain, December 2020.

[19] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, TX, USA, November 2016.

[20] J. Su, Y. Lu, S. Pan, B. Wen, and Y. Liu, "Roformer: enhanced transformer with rotary position embedding," 2021, https://arxiv.org/abs/2104.09864.

[21] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 464–468, New Orleans, LA, USA, June 2018.

[22] C. Raffel, N. Shazeer, A. Roberts et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2020.

[23] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: attentive language models beyond a fixed- length context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988, Florence, Italy, July 2019.

[24] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: decoding-enhanced bert with disentangled attention," in *Proceedings of the International Conference on Learning Representations*, Addis Ababa, Ethiopia, April 2020.

[25] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, pp. 5998–6008, MIT Press, Cambridge, MA, USA, 2017.

[26] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, "Pre-training with whole word masking for Chinese bert," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3504–3514, 2021.

[27] Z. Liu, W. Lin, Y. Shi, and J. Zhao, "A robustly optimized BERT pretraining approach with post-training," in *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pp. 1218–1227, Hohhot, China, August 2021.