IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 September 2024, accepted 25 October 2024, date of publication 29 October 2024, date of current version 7 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3487581

RESEARCH ARTICLE

QARR-FSQA: Question-Answer Replacement and Removal Pretraining Framework for Few-Shot Question Answering

SIAO WAH TAN[1], CHIN POO LEE[2], (Senior Member, IEEE), KIAN MING LIM[3], (Senior Member, IEEE), CONNIE TEE[1], (Senior Member, IEEE), AND ALI ALQAHTANI[4,5]

[1]Faculty of Information Science and Technology, Multimedia University, Malacca 75450, Malaysia
[2]School of AI and Advanced Computing, Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu 215412, China
[3]School of Computer Science, University of Nottingham Ningbo China, Yinzhou, Ningbo, Zhejiang 315100, China
[4]Department of Computer Science, King Khalid University, Abha 61421, Saudi Arabia
[5]Center for Artificial Intelligence (CAI), King Khalid University, Abha 61421, Saudi Arabia

Corresponding authors: Chin Poo Lee (ChinPoo.Lee@xjtlu.edu.cn) and Kian Ming Lim (Kian-Ming.Lim@nottingham.edu.cn)

This work was supported in part by the Telekom Malaysia Research and Development under Grant RDTC/231075 and Grant RDTC/231084; and in part by the Deanship of Scientific Research at King Khalid University, Saudi Arabia, under Grant RGP2/332/44.

ABSTRACT In Natural Language Processing, creating training data for question answering (QA) systems typically requires significant effort and expertise. This challenge is amplified in few-shot scenarios where only a limited number of training samples are available. This paper proposes a novel pretraining framework to enhance few-shot question answering (FSQA) capabilities. It begins with the selection of the Discrete Reasoning Over the Content of Paragraphs (DROP) dataset, designed for English reading comprehension tasks involving various reasoning types. Data preprocessing converts question-answer pairs into a predefined template, consisting of a concatenated sequence of the question, a mask token with a prefix, and the context, forming the input sequence, while the target sequence includes the question and answer. The Question-Answer Replacement and Removal (QARR) technique augments the dataset by integrating the answer into the question and selectively removing words. Various templates for question-answer pairs are introduced. Models like BART, T5, and LED are then used to evaluate the framework's performance, undergoing further pretraining on the augmented dataset with their respective architectures and optimization objectives. The study also investigates the impact of different templates on model performance in few-shot QA tasks. Evaluated on three datasets in few-shot scenarios, the QARR-T5 method outperforms state-of-the-art FSQA techniques, achieving the highest F1 scores of 81.7% in 16-shot and 32-shot, 82.7% in 64-shot, and 84.5% in 128-shot on the SQuAD dataset. This demonstrates the framework's effectiveness in improving models' generalization and performance on new datasets with limited samples, advancing few-shot QA.

INDEX TERMS Natural language processing, few-shot question answering, pretraining framework, generative question answering models.

I. INTRODUCTION

The task of Question Answering (QA) has been a prominent focus in the field of Natural Language Processing (NLP), where substantial progress has been made in leveraging large datasets to train sophisticated models. These models, when provided with ample training data, often exhibit impressive performance in generating accurate answers based on given questions and context. However, the practical application of QA systems encounters a significant challenge when the available dataset is limited to only a few samples.

Traditional QA models, which excel in scenarios with abundant training data, struggle to maintain their effectiveness in situations where only a handful of examples are

The associate editor coordinating the review of this manuscript and approving it for publication was Maria Chiara Caschera.

available. This issue has prompted research into Few-Shot Question Answering (FSQA), a specialized subset of Few-Shot Learning. FSQA specifically addresses the challenges associated with QA tasks when confronted with a scarcity of training data. Existing work in the broader domain of QA has primarily focused on optimizing models for scenarios with abundant data, often overlooking the unique challenges presented by situations with limited samples. Studies have illustrated a notable drop in the performance of models such as SpanBERT when transitioning from a large dataset to a few-shot setting.

In response to these challenges, this paper delves into the domain of FSQA, recognizing the need for models that can effectively answer questions even when provided with only a limited number of samples for training. This research introduces the Question-Answer Replacement and Removal (QARR) method, a novel technique designed to increase the number of effective training datasets. The proposed technique enhances the model's performance by inferring missing information and completing the questions using contextual information, thereby addressing gaps in the datasets and improving the quality of the training process. Additionally, this paper explores the impact of using different templates for the target and input data. By experimenting with various template structures, this study shows how different templates can influence the model's performance on the task of few-shot question answering.

The contributions of this research are as follows:

- Introduction of a pretraining framework that generalizes well and achieves good performance on new datasets with limited samples.
- Proposal of the QARR technique to improve model performance by inferring missing information and completing question sentences using contextual information.
- Investigation of the impact of different templates for the target and input data to enhance model understanding in few-shot question answering.
- Evaluation of the proposed method on three different generative models and datasets, showing superior performance in few-shot scenarios (16-shot to 128-shot).

II. RELATED WORKS

In the pursuit of advancing the field of QA, researchers have explored a myriad of approaches, methodologies, and frameworks. This section provides a literature review in QA, encompassing diverse techniques ranging from bidirectional language models to knowledge-driven question answering. The exploration covers various aspects, including semi-supervised learning, generative prompting, knowledge enhancement, and few-shot learning strategies. Each method is examined in terms of its underlying principles, experimental setups, datasets used, and the achieved results.

Patel et al. [1] introduced a technique known as Sequential Autoregressive Prompting (SAP) that utilizes a bidirectional model for prompting. The bidirectional model employed is mT5, a variant of the T5 model pretrained on multiple languages. To address mT5's tendency to generate partial sentence outputs, SAP implements a generation stop when a stop token </s> is encountered. For enabling longer generations, the first word generated is concatenated with the last line, and this concatenated line is utilized in the subsequent step. In the context of question answering, the authors conducted a comparative study with XGLM using a multilingual dataset called XQuAD. The proposed method achieved a notable 14.6% exact match (EM) and a 37.3% F1 score in the zero-shot setting on the XQuAD dataset. Furthermore, a comparison was made with T5+LM using the English-only SQuAD v1.1 dataset. In the zero-shot setting, the proposed method outperformed T5+LM, recording results of 30.2% EM and 54.0% F1 score. In the 16-shot setting, it demonstrated a 60.0% F1 score and a 35.4% EM. These results underscore the effectiveness of the proposed method in generating short span answers, showcasing adaptability beyond its original design for producing long generations in machine translation tasks.

Dhingra et al. [2] proposed an extractive question answering model that requires input in the form of base documents and a few labeled examples. The base documents themselves do not have any labels. The initial step involves constructing cloze-style questions from the base documents, which are then used to pretrain a neural network model. The labeled examples are used for fine-tuning the pretrained model. For the generation of cloze-style questions, the first 20% of the text is defined as the introduction, while the remaining portion is considered the summary. An exact string match between the introduction and summary is selected as an answer span if the sequence fulfills specific requirements, such as being a verb phrase, noun phrase, named entity, or adjective phrase. Experiments were conducted using three datasets: the SQuAD dataset, the TriviaQA dataset, and the BioASQ 5b dataset. In these experiments, 10% of the questions were reserved for the testing set. The proposed method achieved a 50.42% F1 score on the SQuAD dataset and a 55.21% F1 score on the TriviaQA dataset by using only 1% of the training set for both datasets. Additionally, the method achieved a 23.0% F1 score on list questions of the BioASQ 5b dataset using a 5-fold cross-validation setting.

Chen et al. [3] introduced a new framework called Generative Prompt-Based Data Augmentation (GOTTA) to harness the power of prompt-tuning. This framework augments the training data with cloze-style questions, allowing the model to better understand the original questions. The augmentation process involves three steps. First, tokens to be masked are identified by extracting text spans recognized as Wikidata entities. Prompt data is then constructed using a designed template and the masked tokens. Next, the created prompt data is fed into a pre-trained BART model alongside the original QA training samples. Experiments were conducted as part of the Machine Reading for Question Answering (MRQA) 2019 shared task. The results demonstrated F1 scores of 57.8% on the SQuAD dataset, 40.8% on the

TriviaQA dataset, 47.1% on the NaturalQuestions dataset, 36.2% on the NewsQA dataset, 41.8% on the SearchQA dataset, 45.9% on the HotpotQA dataset, 55.2% on the BioASQ dataset, and 20.5% on the TextbookQA dataset.

Wang et al. [4] presented a novel framework known as Knowledge Enhanced Contrastive Prompt-Tuning (KECP). This framework transforms the task of predicting the start and end positions of answer spans into a non-autoregressive Masked Language Modeling (MLM) generation problem using a specifically designed template mapping. To aggregate knowledge from multiple resources and incorporate it into the input embeddings of the query prompt, the authors introduced a Knowledge-aware Prompt Encoder (KPE). This step enhances the model's understanding of the query. Additionally, the model filters out negative spans that bear a similarity in semantics to the correct answer, effectively discarding potentially confusing or misleading negative predictions. In their experiments, the authors achieved an impressive F1 score of 75.45% on the SQuAD2.0 dataset using only 16 training samples.

Chada and Natarajan [5] devised a framework that aligns fine-tuning with the same objective as pre-training. This approach aims to bridge the gap between the pre-training and fine-tuning phases that often leads to performance degradation. By aligning the fine-tuning phase, the knowledge acquired during pre-training can be optimally utilized. The authors designed a template for input and output. They conducted experiments as part of the MRQA shared task, and the proposed framework achieved notable F1-score performances: 55.5% on the SQuAD dataset, 50.5% on the TriviaQA dataset, 46.7% on the Natural Questions dataset, 38.9% on the NewsQA dataset, 39.8% on the SearchQA dataset, 45.1% on the HotpotQA dataset, 49.4% on the BioASQ dataset, and 19.9% on the TextbookQA dataset, all in a 16-shot setting.

Castel et al. [6] proposed a decoding method called exact-extract to optimize greedy decoding. This new decoding method predicts the probability of a span along with its prefixes, whereas greedy decoding predicts the probability of a span individually. This approach improves space complexity. T5-v1.1 was used as the model to conduct experiments with the exact-extract decoding method. The model's goal is to generate multiple randomly-masked spans. These experiments were carried out as part of the MRQA shared task, and the proposed decoding method achieved the following F1-score performances: 54.9% on the SQuAD dataset, 50.9% on the TriviaQA dataset, 42.1% on the Natural Questions dataset, 28.9% on the NewsQA dataset, 37.4% on the SearchQA dataset, 40.9% on the HotpotQA dataset, 42.3% on the BioASQ dataset, and 17.2% on the TextbookQA dataset, all in a 16-shot setting.

Ram et al. [7] introduced a pre-training framework for question answering known as 'Recurring Span Selection'. They recognized that the objective of fine-tuning in question answering differs from pre-training, and to address this, they developed a model called 'Span-level Pointer' (Splinter). The Splinter model begins by replacing spans that appear repeatedly in the text with a [QUESTION] token, with the model's task being to predict the original text of the [QUESTION] token. This approach allows the knowledge acquired during pre-training to be reused effectively in the fine-tuning process, aligning the training objectives between pre-training and fine-tuning. During fine-tuning, a [QUESTION] token is added after the question, enabling the fine-tuning knowledge used to predict the span of the [QUESTION] token to be reused. Experiments conducted as part of the MRQA shared task demonstrated that the proposed method achieved F1-scores of 54.6% on the SQuAD dataset, 18.9% on the TriviaQA dataset, 27.4% on the Natural Questions dataset, 20.8% on the NewsQA dataset, 26.3% on the SearchQA dataset, 24.0% on the HotpotQA dataset, 28.2% on the BioASQ dataset, and 19.4% on the TextbookQA dataset with a 16-shot setting

Li et al. [8] presented a refined question-answering dataset named REFQA, created to address the issue of lexical overlaps between context paragraphs and generated questions. This dataset comprises question and answer pairs collected from Wikipedia using an unsupervised method. The data collection process involves taking a statement from a paragraph and extracting the corresponding context from the cited document of that statement. The answer is then extracted from the context of the cited document. Subsequently, the context is transformed into a natural question, creating a pair with the answer. To filter out noisy samples from the REFQA dataset, a QA model with BERT as its backbone was employed. The proposed model achieved an F1-score of 79.4% with only 100 labeled samples.

Zaratiana et al. [9] proposed a modified method, DyREx, derived from the vanilla approach [10]. The vanilla approach involves concatenating the passage and question as input and computing the probability of the start and end positions of the answer spans. Identifying the suboptimal nature of the vanilla approach, Zaratiana et al. noted that the query vectors used are static and independent of the input sequence. Therefore, they proposed employing an L-layers transformer decoder to obtain dynamic representations. The transformer layer incorporates a bi-directional self-attention module to aggregate information, enhancing the model's understanding of the context. Utilizing SpanBERT for token representations, DyREx achieved a performance of a 70.75% F1 score on the SQuAD dataset in a 256-sample setting, surpassing the vanilla approach by 5.01% F1 score. This experiment highlights the importance of contextual queries in few-shot question answering

Banerjee and Baral [11] implemented Knowledge Triplet Learning (KTL) to leverage knowledge graphs for zero-shot multiple-choice question-answering. In KTL, a triplet comprises a head, tail, and relation. During training, the model is tasked with producing the third component of the triplet when given the other two components as input. When applied

to multiple-choice question-answering, the components of the triplet correspond to the context, question, and answer options. Knowledge representation learning is employed to achieve the goal of generating the third component, with the scoring function based on the distance between the generated output and the ground truth. The Common Concept Graph and Directed Story Graph were utilized to construct the knowledge graph. The Common Concept Graph involves extracting verb-chunks and noun-chunks from the text corpus using the Spacy Part-of-Speech tagger [12]. The Directed Story Graph consists of independent story graphs extracted from the Story Cloze Test dataset and RoCStories. This proposed framework demonstrated a 48.5% accuracy on SocIQA datasets.

Ma et al. [13] designed a novel framework for the zero-shot commonsense question-answering task, which involves selecting the most likely single answer from a set of options, with the remaining choices acting as distractors. Five knowledge graphs were employed to construct three question-answer datasets. The first dataset was built using ATOMIC, the second utilized CMWV, incorporating the other four knowledge graphs, and the third, CSKG, amalgamated all the knowledge graphs. If a triple could not directly construct a question and answer set, the tail of the sentence generated using a pre-defined template was removed and selected as the answer. Distractors were random samples using the same relation triple as the correct answer, adhering to specific rules to ensure fairness. Various distractor sampling techniques were introduced to enhance the framework's performance. Experimental results indicated that RoBERTa outperformed GPT2 under the proposed framework. Specifically, the framework using RoBERTa achieved accuracies of 70.5% for the aNLI dataset, 67.4% for the CSQA dataset, 72.4% for the PIQA dataset, 63.2% for the SIQA dataset, and 60.9% for the WG dataset in a zero-shot setting using the CSKG dataset.

Lyu et al. [14] utilized question generation (QG) to generate question and answer pairs for a question-answering task. The proposed method leverages article summaries to generate answers. The process involves Dependency Parsing (DP) to identify the root verb of the summary, followed by Named Entity Recognition (NER) to tag entities, and Semantic Role Labeling (SRL) to obtain semantic frames. The arguments extracted from SRL are utilized to identify the wh-words for question generation. The proposed method is evaluated on a question-answering task, achieving a notable 43.0% F1 score on the TriviaQA dataset.

Izacard et al. [15] introduced Atlas, a retrieval-augmented language model designed for knowledge-intensive tasks with minimal training samples. This framework utilizes a text-to-text architecture, where the input is a text query, and the output is also in text format. Atlas consists of two main components: the retriever and the language model. The retriever module, employing a dual-encoder architecture, retrieves relevant documents from a large text corpus. The language model, based on T5, generates the output using information obtained from the retrieval module. Tests were conducted on various benchmarks across different tasks. The proposed model achieved an impressive 74.5% accuracy in a 64-shot setting on TriviaQA filtered.

Lewis et al. [16] implemented an unsupervised method for generating data for question answering tasks. In the generative data model, the process begins by sampling a paragraph from available documents. Answer spans are then generated based on the selected paragraph using two distinct variants. Noun phrases within the paragraph are extracted using a chunking algorithm, and named entities are identified through a Named Entity Recognition (NER) system. These noun phrases and named entities are chosen as answer candidates. For question generation, the approach involves two stages: cloze generation and cloze translation. Cloze generation shortens the sentence and masks the answer to create a cloze question. Subsequently, cloze translation translates the cloze question into a proper question-answering format, employing four distinct approaches for cloze translation. The authors demonstrated the effectiveness of the method by achieving a notable 56.4% F1 score on the SQuAD dataset in a zero-shot setting.

Fabbri et al. [17] proposed a method for unsupervised generation of training data for question answering by integrating sentence retrieval and template-based question generation. The process begins with sentence retrieval, which aims to acquire sentences similar to the one containing the answer. ElasticSearch is employed to index the sentences, and named entities are extracted. The retrieved sentences passing specific filtering rules proceed to the question generation stage. Various question style templates are proposed to transform the retrieved sentence into a question. One template involves replacing the answer with a mask token, while another template replaces wh-words in the question sentence. The proposed method demonstrates effectiveness, achieving a notable 64.04% F1 score on the SQuAD dataset, contingent on the answer being a named entity. Table 1 provides a summary of related work in QA.

III. QUESTION-ANSWER REPLACEMENT AND REMOVAL PRETRAINING FRAMEWORK FOR FEW SHOT QUESTION ANSWERING (QARR-FSQA)

This research involves two phases: the pretraining phase and the fine-tuning phase. The pretraining phase is crucial for helping the model learn general question-answering patterns, including how questions are structured, how to extract information from the context, and how to generate appropriate answers. It establishes a foundation for the model, enabling it to understand questions within context and generate accurate responses. This phase uses a large dataset to pretrain the model.

The fine-tuning phase focuses on adapting the pretrained model to the specific task of question answering. In this research, the emphasis is on question answering with limited training samples, making the knowledge gained during pretraining particularly valuable. Fine-tuning adjusts the model's abilities to perform better on the specific dataset.

TABLE 1. Summary of related work in question answering.

Method	Datasets	F1 Score
Sequential Autoregressive Prompting (SAP) [1]	XQuAD, SQuAD v1.1	EM 14.6%, 37.3% (XQuAD), EM 30.2%, 54.0% (SQuAD)
Extractive QA with Semi-Supervision [2]	SQuAD, TriviaQA, BioASQ 5b	50.42% (SQuAD), 55.21% (TriviaQA), 23.0% (BioASQ 5b)
Generative Prompt-Based Data Augmentation (GOTTA) [3]	MRQA 2019 shared task datasets	36.2% (NewsQA), 47.1% (NaturalQuestions), 55.2% (BioASQ), 40.8% (TriviaQA), 41.8% (SearchQA), 45.9% (HotpotQA), 57.8% (SQuAD)
Knowledge Enhanced Contrastive Prompt-Tuning (KECP) [4]	SQuAD2.0	75.45%
FewshotQA Framework [5]	MRQA shared task datasets	39.8% (NewsQA), 45.1% (HotpotQA), 46.7% (Natural Questions), 50.5% (TriviaQA), 54.6% (SQuAD)
Optimal Greedy Decoding (exact-extract) [6]	MRQA shared task datasets	28.9% (NewsQA), 37.4% (SearchQA), 42.1% (Natural Questions), 50.9% (TriviaQA), 54.9% (SQuAD)
Recurring Span Selection [7]	MRQA shared task datasets	20.8% (NewsQA), 26.3% (SearchQA), 27.4% (Natural Questions), 54.6% (SQuAD)
Harvesting and Refining Question-Answer Pairs (REFQA) [8]	REFQA dataset	79.4%
DyREx: Dynamic Query Representation [9]	SQuAD	70.75%
Knowledge Triplet Learning (KTL) [11]	SocIQA datasets	Accuracy 48.5%
Knowledge-Driven Data Construction [13]	CSKG dataset	Accuracies 60.9% (WG), 63.2% (SIQA), 67.4% (CSQA), 70.5% (aNLI), 72.4% (PIQA) (Zero-shot setting)
Question Generation for QA [14]	TriviaQA	43.0%
Atlas: Retrieval-Augmented Model [15]	TriviaQA filtered	Accuracy 74.5% (64-shot setting)
Unsupervised Data Generation [16]	SQuAD	56.4%
Template-Based QA Data Generation [17]	SQuAD	64.04% (Named entities)

The input and target templates play a crucial role in connecting these two processes, serving as a bridge between the pretraining and fine-tuning phases. The templates ensure that the knowledge acquired during pretraining is effectively transferred and utilized during fine-tuning, allowing the model to excel in the few-shot question-answering task.

A. PRETRAINING FRAMEWORK

This paper presents a pretraining framework to address the task of few-shot question answering. The pretraining process involves several key components, including the selection of a suitable pretraining dataset, preprocessing of the data, adaptation of the QARR technique to augment the dataset, and further pretraining of the generative models.

The pretraining dataset chosen for this study is the Discrete Reasoning Over the Content of Paragraphs (DROP) dataset [18], specifically designed for English language reading comprehension tasks. This dataset offers a diverse range of reasoning scenarios, covering various types of reasoning such as answer selection, span comparison, arithmetic operations, counting, sorting, and logical reasoning. Each instance in the DROP dataset comprises contexts containing multiple question-answer pairs, providing a rich source of training data for pretraining the models.

Before the models undergo pretraining, the dataset undergoes preprocessing to ensure its structural integrity and compatibility with the few-shot question answering task. Each question-answer pair is transformed into a predefined template, aligning with the pattern of the few-shot question answering task. This template consists of a concatenated sequence of the question, a mask token with a prefix, and the context, which serves as the input sequence for the model. Simultaneously, the target sequence is formed by concatenating the question and its corresponding answer, guiding the model towards generating precise answers given specific questions and contexts.

The QARR technique plays a crucial role in augmenting the pretraining dataset, enhancing the coherence and relevance of question-answer pairs. This technique involves integrating the answer directly into the question, followed by selective removal of a predetermined ratio of words from the sentence. Through this iterative process, new question-answer pairs are generated while maintaining coherence and relevance to the context. To adapt the QARR technique, modifications are made to the template structure to accommodate the integration of the answer into the question effectively.

After preprocessing and augmentation with QARR, the pretrained BART, T5, and LED models undergo further pretraining on the augmented DROP dataset. This pretraining process involves training the models on the augmented data using their respective architectures and optimization objectives. For BART, this entails corrupting documents using various methods such as token masking, deletion, infilling, sentence permutation, and document rotation, followed by optimization using cross-entropy loss. Similarly, T5 and LED undergo pretraining using their respective training objectives and architectures, with T5 translating all text-based language problems into a text-to-text format, and LED efficiently processing extensive sequences of documents using its innovative attention mechanisms.

Overall, the pretrained BART, T5, and LED models are pretrained on the QARR augmented DROP dataset, incorporating the QARR technique to enhance the coherence and relevance of the dataset. This preprocessing and augmentation strategy aims to improve the models' performance in few-shot question answering tasks by providing a more diverse and contextually relevant training dataset.

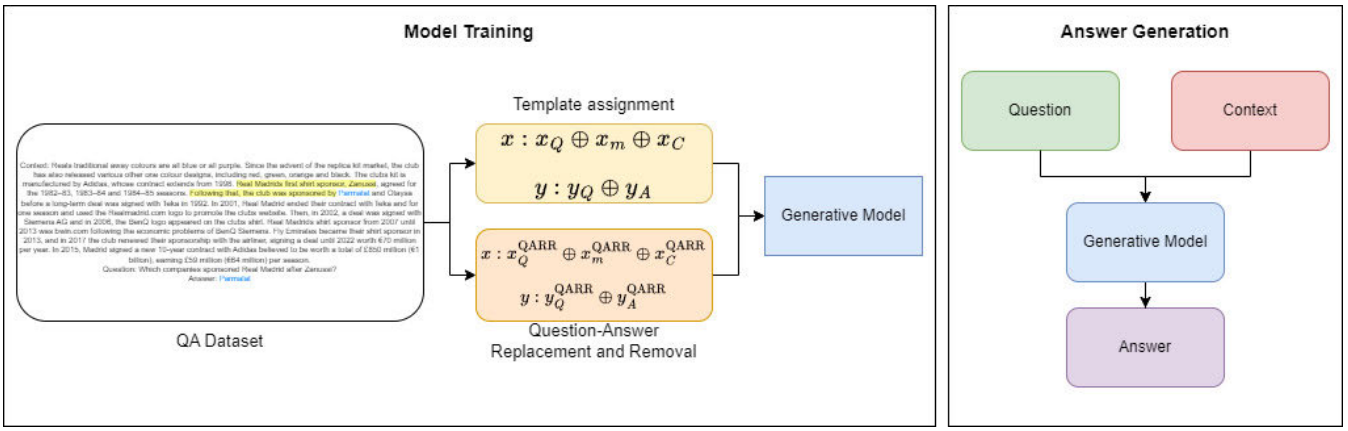


FIGURE 1. The framework of the proposed QARR-FSQA method.

B. PRETRAINING DATASET

This study leverages the Discrete Reasoning Over the Content of Paragraphs (DROP) dataset [18] as the cornerstone for pretraining. Specifically designed for English language reading comprehension tasks, the DROP dataset offers a rich tapestry of reasoning scenarios. These scenarios encompass a wide spectrum of reasoning types, including answer selection, span comparison, arithmetic operations such as addition or subtraction, counting, sorting, and other intricate forms of logical reasoning. The diversity in reasoning types encapsulated within the dataset mirrors the expansive range of question types encountered in various question answering tasks.

For instance, the dataset encompasses questions involving counting reasoning, such as "How many ways can…," necessitating inference-making based on counting principles. Moreover, the dataset's answers manifest in various formats, including answer spans within passages, dates, and numerical values, thereby providing a holistic evaluation of comprehension and reasoning abilities. Table 2 illustrates examples of three answer types from the DROP dataset.

TABLE 2. Types of answer in the DROP dataset.

Answer Type	Question	Answer
Spans	Who served first as commander, Murray or O'Sullivan?	O'Sullivan
Dates	When was the Ultimate Toy Box set released?	October 17, 2000
Numbers	How many days after launching the rebellion did the Jacobites reach Perth?	16

C. PREDEFINED TEMPLATE ASSIGNMENT

Prior to model training, preprocessing of the dataset is imperative to ensure its structural integrity and compatibility for effective learning. This preprocessing commences with data extraction from the pretraining dataset, specifically the DROP dataset. Each instance within this dataset comprises contexts, each housing multiple question-answer pairs.

To streamline the input data and optimize learning, each question-answer pair undergoes a sequence of transformations. These transformations entail mapping the question, answer, and context of every pair to a predefined template [5], thus aligning with the pattern of the few-shot question answering task. The input sequence is crafted by concatenating the question, a mask token with a prefix, and the context. This fusion empowers the model to harness contextual cues effectively, thereby enhancing response accuracy.

Concurrently, the target sequence is formed by concatenating the question and its corresponding answer. This concatenated sequence serves as the target output of the model, guiding it towards generating precise answers given specific questions and contexts. Let `q` denotes the question sentence, `<mask>` represents the mask token, `c` signifies the context sentence, and `a` signifies the answer. The template for input and target sequences is structured as follows.

Predefined Template

Input template: Question: q? Answer: <mask>. Context: c

Target template: Question: q? Answer: a.

D. QUESTION-ANSWER REPLACEMENT AND REMOVAL (QARR)

The process of Question-Answer Replacement and Removal (QARR) represents a crucial step in data augmentation, enhancing the coherence and relevance of question-answer pairs within a dataset. After an initial processing stage, this augmentation technique involves replacing the question word in each question with the corresponding answer, thereby creating a complete sentence. This integration of the answer directly into the question enhances the contextual relevance of the resulting sentence. Following this integration, a predetermined ratio of words, typically the last ones in the sentence, is removed. This selective removal is vital for generating new question sentences while maintaining coherence and relevance to the context. The removed portion

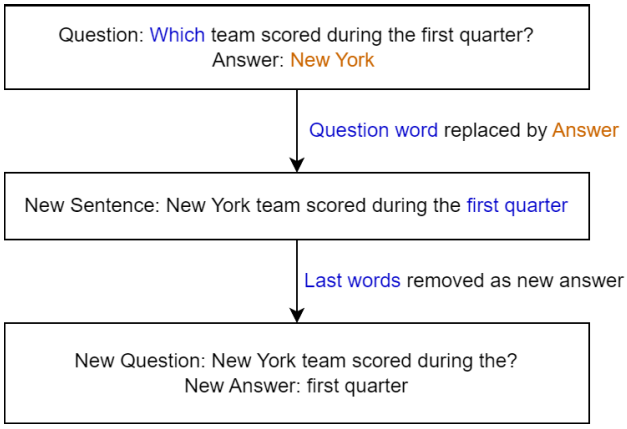


FIGURE 2. Example of question-answer replacement and removal technique.

of the sentence becomes the new answer, while the remaining portion transforms into the new question sentence. Through this iterative process, QARR ensures the coherence and relevance of the generated question-answer pairs, while also aiding in the extraction of pertinent information from the context.

Figure 2 provides a visual representation of the QARR technique, illustrating how it transforms a question-answer pair by integrating the answer into the question and subsequently removing a portion of the sentence to generate a new question.

E. MODIFIED TEMPLATES

To effectively adapt to the QARR technique, modifications to the template used for generating question-answer pairs are necessary. Two modified templates are proposed to replace the predefined template. These modifications involve the introduction of a mask token within the template structure, serving as a guide for the generation process to indicate where the answer should be inserted within the question. The first modified template adds a mask token both at the input and target levels, while the second modified template incorporates the mask token solely at the input level.

Modified Template 1

Input template: Question: q <mask>? Answer: <mask>. Context: c

Target template: Question: q <mask>? Answer: a.

Modified Template 2

Input template: Question: q <mask>? Answer: <mask>. Context: c

Target template: Question: q? Answer: a.

The integration of the QARR technique significantly expands the dataset, increasing the number of training samples from 29,195 to 57,150. This augmentation leads to a more robust dataset for training models, ultimately enhancing their performance in question-answering tasks. Table 3 presents the number of samples in the DROP dataset before and after QARR augmentation. Some examples of the DROP dataset before and after QARR with different templates are provided in Table 4, demonstrating the effectiveness of these techniques in generating diverse and contextually relevant question-answer pairs.

TABLE 3. Number of samples in the DROP dataset before and after QARR with different templates.

Dataset	Number of Samples
DROP (Predefined Template)	29,195
DROP + QARR (Predefined Template)	57,150
DROP + QARR (Modified Template 1)	57,150
DROP + QARR (Modified Template 2)	57,150

TABLE 4. Examples from the DROP dataset and QARR-Augmented DROP dataset with different templates.

Process	Input	Target
DROP (Predefined Template)	Question: Which team scored during the first quarter? Answer: <mask>. Context: Hoping ... In the first quarter, New York took flight as QB Brett Favre completed an 18-yard TD pass to RB Leon Washington. In the second quarter, ...	Question: Which team scored during the first quarter? Answer: New York
DROP + QARR (Predefined Template)	Question: New York team scored during the? Answer: <mask>. Context: Hoping ... In the first quarter, New York took flight as QB Brett Favre completed an 18-yard TD pass to RB Leon Washington. In the second quarter, ...	Question: New York team scored during the? Answer: first quarter
DROP + QARR (Modified Template 1)	Question: New York team scored during the <mask>? Answer: <mask>. Context: Hoping ... In the first quarter, New York took flight as QB Brett Favre completed an 18-yard TD pass to RB Leon Washington. In the second quarter, ...	Question: New York team scored during the <mask>? Answer: first quarter
DROP + QARR (Modified Template 2)	Question: New York team scored during the <mask>? Answer: <mask>. Context: Hoping ... In the first quarter, New York took flight as QB Brett Favre completed an 18-yard TD pass to RB Leon Washington. In the second quarter, ...	Question: New York team scored during the? Answer: first quarter

F. PRETRAINING MODELS

This section describes three pivotal pretraining models utilized in this study: Bidirectional and Auto-Regressive Transformers (BART), Text-to-Text Transfer Transformer (T5), and Longformer-Encoder-Decoder (LED). These models stand as pillars in the realm of natural language processing, each contributing uniquely to various generative tasks.

1) BIDIRECTIONAL AND AUTO-REGRESSIVE TRANSFORMERS (BART)

The Bidirectional and Auto-Regressive Transformers (BART) model is a sequence-to-sequence transformer architecture utilized in generative processes, where the model generates answers rather than selecting them from the input context. The BART model, as described by Lewis et al. [19], is built upon the foundational Transformer architecture introduced by Vaswani et al. [20]. Unlike traditional transformer models, BART lacks a feed-forward network before word prediction, distinguishing it in its approach to sequence generation.

During the pre-training phase, the BART model undergoes training by corrupting documents through various methods, including token masking, deletion, infilling, sentence permutation, and document rotation. These techniques aim to simulate scenarios where information is severely distorted or lost. The model is optimized using cross-entropy loss, expressed by the following equation:

$$\text{Loss} = -\sum_{i}^{T} y_i \cdot \log(f(x)_i)$$

Here, T represents the length of the sequence, y_i denotes the true probability distribution, and $f(x)_i$ represents the predicted probability distribution for the i-th token.

The architecture of the BART model consists of both an encoder and a decoder. The encoder utilizes a bidirectional model to encode the corrupted document, enabling it to capture contextual information from both forward and backward directions. On the other hand, the decoder employs an autoregressive approach to compute the likelihood of the original document, predicting the next token in the sequence based on the tokens generated thus far. The base version of the BART model comprises six encoder and decoder layers, totaling approximately 140 million parameters. This architecture, depicted in Figure 3, illustrates the flow of information within the model and highlights its bidirectional encoding and autoregressive decoding mechanisms. This combination of bidirectional encoding and autoregressive decoding allows the BART model to effectively generate coherent and contextually relevant sequences, making it well-suited for a variety of generative tasks in natural language processing.

2) TEXT-TO-TEXT TRANSFER TRANSFORMER (T5)

The Text-to-Text Transfer Transformer (T5) model, as introduced by Raffel et al. [21], revolutionizes natural language processing (NLP) by presenting a unified framework that translates all text-based language problems into a text-to-text format. This innovative approach simplifies the treatment of various NLP tasks, making them amenable to a common model architecture and training objective. The development of the T5 model begins with the construction of a foundational dataset named Colossal Clean Crawled Corpus (C4), sourced from Common Crawl. This dataset serves as the basis for training the T5 model and provides a diverse range of text data for various NLP tasks.

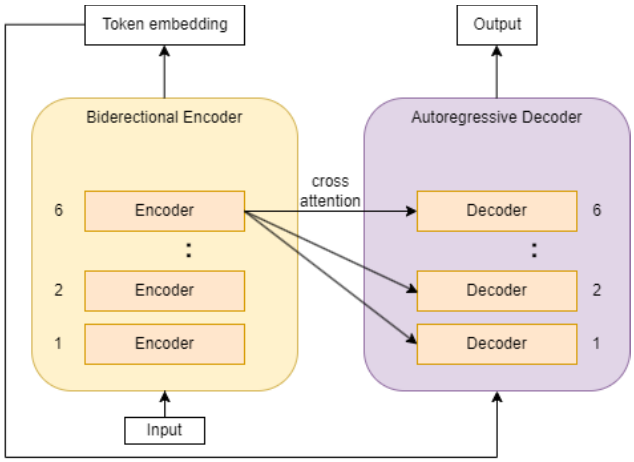


FIGURE 3. Architecture of the base version of the BART model.

The evolution of the T5 model involves a series of experiments aimed at optimizing its architecture and training process. These experiments explore different aspects such as architecture configurations, input-output formats, corruption rates, and more. Each experiment builds upon the insights gained from the previous ones, progressively refining the model's performance. For instance, one crucial experiment investigates various input and target formats, including prefix language modeling templates, BERT-style templates, and deshuffling templates. Among these, the BERT-style template emerges as the most effective, consistently outperforming the others in subsequent experiments. As a result, it becomes the standard input and target format for the T5 model.

Architecturally, the T5 model follows the encoder-decoder Transformer architecture proposed by Vaswani et al. [20]. The default version of T5, denoted as "base", consists of 12 encoder and decoder layers. The encoder comprises self-attention layers, feed-forward networks, and layer normalization [22], while the decoder shares a similar architecture but includes a dense layer with a softmax output at the decoder's final layer. Figure 4 illustrates the architecture of the base version of the T5 model, showcasing the flow of information within the model's encoder and decoder components. Additionally, the version of the T5 model used in the study, denoted as version 1.1, is trained exclusively on the C4 dataset without any supervised training, highlighting the model's capability to learn from large-scale, unsupervised data sources.

3) LONGFORMER-ENCODER-DECODER (LED)

The Longformer architecture, introduced by Beltagy et al. [23], represents a significant advancement in natural language processing (NLP), specifically tailored for efficiently processing extensive sequences of documents. Unlike the original Transformer model, which suffers from memory complexity proportional to the square of the input

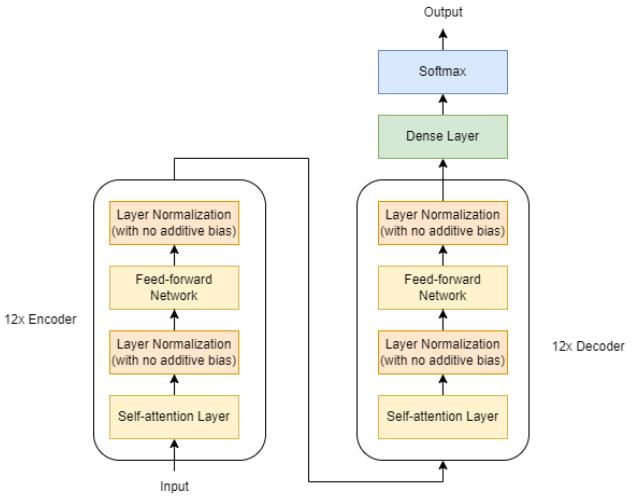


FIGURE 4. Architecture of the base version of the T5 model.

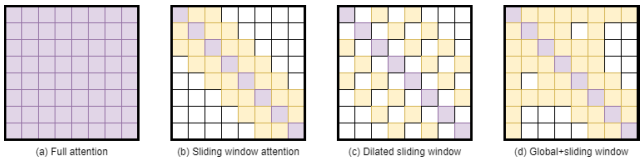


FIGURE 5. (a) Full attention considers all tokens. (b) Sliding window attention considers tokens on each side of specific tokens. (c) Dilated attention increases the gaps between windows. (d) Global attention is added to pre-selected input locations to increase flexibility.

sequence length, Longformer introduces innovative attention mechanisms that scale linearly with the input sequence length.

The key innovation of Longformer lies in its attention patterns, which seamlessly integrate global and local information while managing memory complexity. To achieve this, Longformer employs fixed-size window attention, where each token attends to both sides of the tokens within half of the fixed window size. This approach enables Longformer to efficiently capture long-range dependencies in the input sequence without sacrificing computational efficiency. Longformer further optimizes performance by utilizing sliding windows with a ''dilated'' approach, inspired by dilated convolutional neural networks (CNNs) [24]. This technique introduces gaps between the windows, allowing the model to efficiently process even longer sequences by selectively attending to relevant portions of the input.

In addition to local attention, Longformer incorporates global attention at pre-selected input locations, enabling tokens to attend to and be attended to by all other tokens in the sequence. This global attention mechanism enhances the model's ability to capture contextual information from the entire document while still maintaining computational efficiency. Figure 5 provides a visual representation of the difference in attention patterns between Longformer and traditional Transformer models, highlighting the model's ability to effectively handle long sequences with both local and global attention mechanisms. Following the architecture of BART, the Longformer-Encoder-Decoder (LED) model comprises six layers of encoder and decoder layers, enabling it to effectively process and generate text for various sequence-to-sequence tasks in NLP.

G. ENHANCED PRETRAINING STRATEGY

Following initial preprocessing and augmentation with QARR, BART, T5, and LED models undergo an additional round of pretraining on the QARR-augmented DROP dataset with predefined template, modified template 1, and modified template 2, separately. This phase is pivotal for fine-tuning the models to grasp the intricacies introduced by the QARR technique and optimize their parameters to adeptly handle the augmented data. Each model employs its unique architecture and optimization objectives to refine its comprehension of the augmented dataset and bolster its performance on few-shot question answering tasks.

During BART's pretraining, the model is exposed to the augmented data. These methods simulate scenarios where information is distorted or partially absent, fostering the learning of robust representations resilient to noise and data variations. BART's optimization objective during pretraining is to minimize cross-entropy loss, aligning the predicted probability distribution with the true distribution of tokens in the input sequence. The resulting model after the second round of pretraining on QARR-augmented DROP dataset is denoted as QARR-BART.

Similarly, T5 and LED undergo further pretraining utilizing their respective architectures and objectives. T5 adheres to its text-to-text format, translating all text-based language problems into this format for uniform treatment of various NLP tasks, effectively leveraging the augmented dataset for learning. The resulting model after the second round of pretraining on QARR-augmented DROP dataset is denoted as QARR-T5.

LED, with its advanced attention mechanisms tailored for processing extensive document sequences efficiently, refines its understanding of the augmented data during pretraining. Leveraging fixed-size window attention and global attention mechanisms, LED captures both local and global dependencies within the augmented dataset, enhancing its performance on few-shot question answering tasks. The resulting model after the second round of pretraining on QARR-augmented DROP dataset is known as QARR-LED.

Overall, the pretraining phase on the QARR augmented DROP dataset serves to consolidate the models' understanding of the augmented data and refine their parameters to better capture the underlying patterns and structures introduced by the QARR technique. By incorporating QARR into the pretraining process, the models are exposed to a more diverse and contextually relevant training dataset, leading to improved performance in few-shot question answering tasks. This preprocessing and augmentation strategy plays a crucial role in enhancing the coherence and relevance of the dataset,

ultimately contributing to the models' ability to generalize and perform effectively in real-world question answering scenarios.

IV. FINE-TUNING FOR FEW-SHOT QUESTION ANSWERING

Fine-tuning is a pivotal process in optimizing model performance for few-shot question-answering tasks, where the availability of training data varies. This method involves adapting pretrained models across different "shot" settings, which represent the number of samples in the training set: 16-shot, 32-shot, 64-shot, and 128-shot. The fine-tuning process involves refining each pretrained generative model using three experimental datasets. For each dataset, the process is conducted across four different shot settings and repeated five times for each shot.

Exploring this spectrum of shot settings provides valuable insights into how the model performs under diverse training data conditions. From minimal 16-shot scenarios to more data-rich 128-shot settings, researchers gain a comprehensive perspective on the model's robustness. This exploration not only elucidates the model's capacity to leverage limited training data effectively but also informs its performance in real-world scenarios with varying data availability.

V. ANSWER GENERATION AND EXTRACTION

Following fine-tuning, the fine-tuned models are deployed for answer generation and extraction, as illustrated in Figure 6. Initially, the fine-tuned model receives input comprising the question, a mask token prefixed with "Answer", and context. Leveraging its learned knowledge, the model generates both questions and answers based on a predefined template. This template, structured around the question followed by the generated answer, serves as a scaffolding mechanism, ensuring that the model produces responses that are contextually relevant and coherent.

Once text generation is complete, the model undergoes an extraction process to isolate the generated answers. Specific words or phrases following the designated prefix "Answer" within the generated text are identified and extracted. These extracted answers serve as representations of the model's responses for evaluation purposes. Subsequently, they are compared against ground truth or expected answers to gauge the overall performance of the Question Answering system, providing valuable insights into its efficacy and accuracy.

VI. EXPERIMENTAL DATASETS

Three publicly available question answering datasets are employed in this research: the Stanford Question Answering Dataset (SQuAD) version 1.1 [25], the HotpotQA dataset [26], and the Natural Questions dataset [27].

A. STANFORD QUESTION ANSWERING DATASET (SQUAD)

The SQuAD dataset [25] is a widely used reading comprehension dataset where each question's answer is located within the corresponding passage. Questions and answers

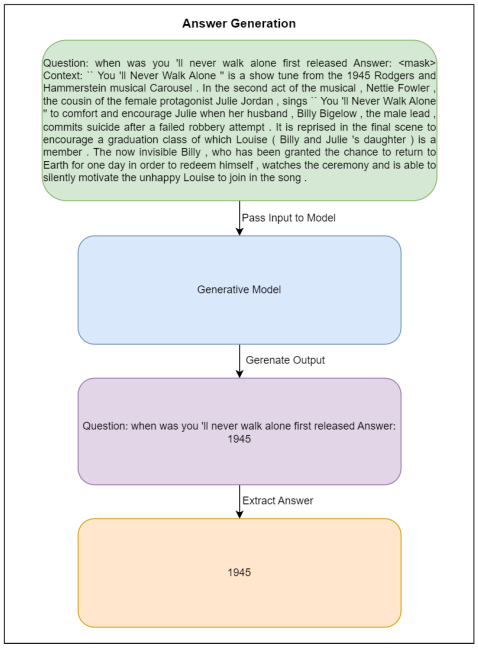


FIGURE 6. Answer generation and extraction process.

are generated by crowdworkers based on a collection of Wikipedia articles. The dataset creation involves three stages: curating passages, collecting questions and answers, and obtaining additional answers. Passage curation includes retrieving articles from English Wikipedia and eliminating low-quality sections. Questions and answers are generated by providing five pairs of questions and answers for every context, with the answer highlighted within the paragraph. For the development and test sets, each question is provided with at least two additional answers to enhance evaluation robustness. SQuAD dataset comprises 10,507 testing questions. Each context in the SQuAD dataset contains multiple questions and answers derived from that context. Table 5 presents examples of different questions and answers derived from a single context.

TABLE 5. Examples of different questions and answers based on one context.

Question	Answer	Context
When was a study conducted of Swedish counties?	between 1960 and 2000	Research by Harvard economist Robert Barro, found that there is "little overall relation between income inequality and rates of growth and investment". According to work by Barro in 1999 and 2000, … A study of Swedish counties between 1960 and 2000 found a positive impact …
What institution does Robert Barro hail from?	Harvard	
Barro found there is little relation between income inequality and rates of what?	growth and investment	

B. HotpotQA

The HotpotQA dataset [26] consists of question and answer pairs requiring multi-hop reasoning. Unlike popular datasets

focusing on single-hop reasoning and existing multi-hop reasoning datasets built on knowledge bases, HotpotQA addresses this gap by necessitating reasoning over multiple documents.

The data collection process begins with constructing a directed graph using hyperlinks as the relationships between two entities. Hyperlinks are extracted from the first paragraph of the articles, as it contains more information compared to the remaining paragraphs. To generate multi-hop reasoning questions, the bridge entity is utilized to identify the relevant paragraph for extracting the question. Additionally, a comparison between two entities under the same category is employed to generate the question. HOTPOTQA dataset comprises 5,901 testing questions. Examples of multi-hop reasoning are illustrated in Table 6.

TABLE 6. Examples of multi-hop reasoning in HotpotQA dataset.

Reasoning	Example
Inferring the bridge entity	Paragraph A: Big Stone Gap is a 2014 American drama romantic comedy film written and directed by Adriana Trigiani and produced by Donna Gigliotti for Altar Identity Studios, … Paragraph B: Adriana Trigiani is an Italian American … based in Greenwich Village, New York City. Trigiani has published a novel a year since 2000. Question: The director of the romantic comedy "Big Stone Gap" is based in what New York city? Answer: Greenwich Village, New York City
Comparison	Paragraph A: Annie Morton (born October 8, 1970) is an American model born in Pennsylvania… Paragraph B: Terrence "Uncle Terry" Richardson (born August 14, 1965) is an American fashion and portrait photographer … Question: Who is older, Annie Morton or Terry Richardson? Answer: Terry Richardson

C. NATURAL QUESTIONS (NQ)

The Natural Questions dataset [27] provides a substantial collection of real-world questions. All questions in the dataset consist of sentences with eight or more words and are submitted to the Google search engine by various users. NQ dataset comprises 12,836 testing questions. These questions undergo the Google search process, and the top five search results containing Wikipedia pages are retained. Wikipedia serves as the factual information source for generating the answers. Each annotation undergoes a thorough process involving question identification, long answer identification, and short answer identification, ensuring that the question is fact-seeking and includes both a long and a short answer. Examples of Natural Questions are presented in Table 7.

VII. EVALUATION METRIC

The F1 score serves as a pivotal evaluation metric in this research, balancing precision and recall to provide a holistic assessment of model performance. It is calculated using the formula:

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1}$$

TABLE 7. Examples of natural questions dataset.

Question	Answer	Context
what is non controlling interest on balance sheet	the portion of a subsidiary corporation's stock that is not owned by the parent corporation	In accounting, minority interest (or non-controlling interest) is the portion of a subsidiary corporation's stock that is not owned by the parent corporation. The magnitude of the minority …
who has been chosen as the brand ambassador of the campaign ' beti bachao-beti padhao	Sakshi Malik	In 26 August 2016, Olympics 2016 bronze medallist Sakshi Malik was made brand ambassador for BBBP.

The F1 score encapsulates both precision and recall metrics, offering a single numerical value reflecting the model's accuracy in identifying relevant information. Recall is calculated as:

$$\text{Recall} = \frac{\text{Common Tokens}}{\text{Gold Answer}} \tag{2}$$

Precision is determined by:

$$\text{Precision} = \frac{\text{Common Tokens}}{\text{Predicted Answer}} \tag{3}$$

Here, "Common Tokens" represents the overlap between Predicted Answer and Gold Answer tokens, indicating true positives. Recall measures the model's capability to retrieve relevant words from the gold answer, with higher values indicating successful retrieval of correct answers. Precision assesses the proportion of correct words among the predicted answers, indicating the model's ability to provide accurate responses.

Unlike the F1 score, Exact Match (EM) is a binary evaluation metric. EM assesses the correctness of the predicted answer by determining whether it exactly matches the gold answer or not. Exact Match only has two possible scores: 0 or 1. The Exact Match score is 1 if the predicted answer is exactly the same as the gold answer; otherwise, the score is 0. Exact Match is an evaluation metric in question answering that is particularly suitable for datasets with short answers.

Furthermore, standard deviation is utilized to gauge the performance consistency across repetitions. A lower standard deviation implies tighter clustering of the scores around the average, indicating more stable results. This is crucial in few-shot question answering scenarios where understanding performance variability is essential. The standard deviation formula is:

$$\sigma = \sqrt{\frac{\sum(x_i - \bar{x})^2}{N}} \tag{4}$$

Here, σ represents the standard deviation, x_i denotes the score for each experiment, $\bar{x}$ is the average score, $\sum$ denotes summation, and N is the number of experiments.

VIII. EXPERIMENTAL RESULTS AND ANALYSIS

Initially, the baseline BART, T5, and LED models are pretrained on the original DROP dataset for 5 epochs with a batch size of 4. The models then undergo further pretraining on the QARR-augmented DROP dataset for up to 25 epochs, with early stopping activated after 3 consecutive epochs without improvement in the validation set's F1 score. This process yields the QARR-BART, QARR-T5, and QARR-LED models. These models are then evaluated using the SQuAD dataset, a widely recognized benchmark for Question Answering tasks.

The pretrained QARR-BART, QARR-T5, and QARR-LED models are subsequently fine-tuned in few-shot settings using the SQuAD, HotpotQA, and Natural Questions datasets for evaluation and comparison with existing works. For each dataset, the models are fine-tuned across four different training sample sizes, with five distinct sets of samples chosen for each size, following the method in [28]. The average performance across these sets is calculated to ensure robust results. Fine-tuning is conducted for up to 25 epochs, with a batch size of 4 and a maximum input sequence length of 512 tokens, consistent with the pretraining phase. All experiments in this paper are conducted on an NVIDIA GeForce RTX 4070 GPU with 12GB memory.

A. EXPERIMENTAL RESULTS OF BASELINE MODELS

Table 8 presents the experimental results of the baseline models after pretraining on the original DROP dataset with different learning rates. In this original DROP dataset, the predefined template is used. For the BART model, the highest F1 score is 74.46% attained at a learning rate of $2e^{-6}$. The choice of a learning rate of $2e^{-6}$ strikes a balance between facilitating effective convergence during pretraining and fine-tuning while preventing overfitting. This moderate learning rate enables efficient parameter learning, ensuring good performance without inducing excessive oscillations or divergence during training.

Moving on to the T5 model, it achieves its highest F1 score of 76.94% at a learning rate of $2e^{-5}$. The relatively higher learning rate of $2e^{-5}$ allows the T5 model to swiftly adapt to the intricacies of the dataset during both pretraining and fine-tuning stages. This learning rate aligns well with T5's architecture, which involves transforming input sequences into output sequences using a unified text-to-text approach. The higher learning rate facilitates faster convergence and better utilization of the model's capacity for capturing complex patterns in the data.

Similarly, for the LED model, the highest F1 score is 74.70% achieved at a learning rate of $2e^{-6}$. A learning rate of $2e^{-6}$ enables the LED model to efficiently capture and integrate information from extended context windows, which is particularly critical for question answering tasks. This learning rate strikes a balance between adaptability and stability, allowing the LED model to leverage its unique architecture effectively and achieve high performance on the given dataset.

TABLE 8. Performance of the baseline models at different learning rates.

Model	Learning Rate	F1 Score (%)	EM (%)	Time(mins)
BART	$2e^{-4}$	0.00	0.00	91.48
	$2e^{-5}$	67.72	55.79	85.98
	$2e^{-6}$	**74.46**	**63.68**	**85.98**
	$2e^{-7}$	66.66	55.33	107.77
T5	$2e^{-4}$	60.22	47.05	240.43
	$2e^{-5}$	**76.94**	**66.99**	**222.45**
	$2e^{-6}$	52.82	41.82	247.65
	$2e^{-7}$	17.76	9.5	245.42
LED	$2e^{-4}$	9.99	5.23	426.53
	$2e^{-5}$	66.91	54.76	430.33
	$2e^{-6}$	**74.70**	**63.23**	**424.33**
	$2e^{-7}$	68.43	57.45	418.10

B. EXPERIMENTAL RESULTS OF PRETRAINED MODELS WITH QARR-AUGMENTED DROP DATASET (PREDEFINED TEMPLATE)

The baseline models are further refined through additional pretraining using the QARR-augmented DROP dataset with a predefined template. Various ratios are employed to control the amount of information retained in the generated questions, representing the proportion of words removed from sentences and selected as answers. For example, a removal ratio of 0.25 indicates that a quarter of the words are removed. This ratio influences the level of detail and content preserved in the questions, which, in turn, affects the model's ability to generate accurate and meaningful answers. Table 9 presents examples of question-answer pairs generated with different QARR removal ratios. These examples illustrate how varying ratios affect the structure and complexity of the questions, providing insights into the amount of content the models need to retain for optimal performance.

Table 10 shows the models' performance when the QARR technique is applied. Three different QARR ratios are explored: 0.1, 0.25, and 0.5, representing low, medium, and high levels of removal, respectively. It is observed that the model's performance improves as the QARR ratio increases up to a certain point, after which performance begins to decline. Ratios of 0.1 and 0.25 enhance the model's performance, indicating that a low to moderate QARR ratio allows for the generation of meaningful questions and answers. Within this effective range, a higher ratio can yield more complex and informative questions. However, when the QARR ratio reaches 0.5, the model's performance drops. This decline is likely because a high QARR ratio removes substantial portions of the question, potentially rendering the sentences less meaningful. This introduces noise into the model, which can degrade performance.

Furthermore, the results reveal that while QARR improves performance in BART and T5 models, it does not have the same effect on the LED model. This suggests that QARR's effectiveness may depend on the model's architecture and its sensitivity to input noise. The LED model, which is

optimized for longer sequences, might struggle with the increased fragmentation and loss of context caused by higher QARR ratios, leading to a decrease in performance.

TABLE 9. Samples of question-answer pairs with different QARR ratios.

Ratio	Question	Answer
0.10	Mahmud was the 3rd ruler of the Hotak?	dynasty
	Steelers team kicked two field goals in the first?	half
	Han Army was the name of the military of the?	Yuan dynasty
0.25	Mahmud was the 3rd ruler of?	the Hotak dynasty
	Steelers team kicked two field goals in?	the first half
	Han Army was the name of the military of?	the Yuan dynasty
0.50	Mahmud was the 3rd?	ruler of the Hotak dynasty
	Steelers team kicked two field?	goals in the first half
	Han Army was the name of?	the military of the Yuan dynasty

TABLE 10. Performance of the pretrained models with different QARR ratios.

Model	QARR ratio	F1 Score (%)	EM (%)	Time (mins)
BART	0.10	74.50	63.27	177.65
	0.25	**75.04**	**63.47**	**157.53**
	0.50	74.24	62.53	186.18
T5	0.10	79.38	68.43	357.48
	0.25	**79.67**	**68.84**	**321.55**
	0.50	79.55	68.55	388.55
LED	**0.10**	**74.58**	**63.12**	**793.23**
	0.25	73.61	62.19	777.85
	0.50	72.47	60.83	799.98

C. EXPERIMENTAL RESULTS OF PRETRAINED MODELS WITH QARR-AUGMENTED DROP DATASET (MODIFIED TEMPLATES)

After identifying the optimal QARR ratio for each model, the QARR technique is applied to the DROP dataset using three distinct templates: predefined, modified template 1, and modified template 2, to evaluate the effects of template variations on QARR-augmented datasets.

Table 11 presents the performance results across different input and target templates. The findings show that template modifications can enhance model performance, with modified template 2, which incorporates an input mask, consistently outperforming the others. Specifically, the BART model achieves an F1 score of 75.24%, the T5 model 80.77%, and the LED model 74.73%. Similar improvements are seen in EM, with the BART model reaching 63.87%, the T5 model 70.06%, and the LED model 63.02%. While there is an increase in training time due to the larger number of samples in the augmented DROP dataset, modifying the QARR template has a negligible impact on training time.

It is noteworthy that the experimental results obtained so far are with the second round of training set to 5 epochs. To further explore the performance of the models with the optimal hyperparameter settings, the maximum training epoch is increased to 25, incorporating an early stopping mechanism. The early stopping is triggered when the F1 score on the validation set does not improve for 3 epochs (patience). The experimental results after training with early stopping enabled are presented in Table 12 alongside the optimal hyperparameter settings. Notably, the QARR-BART model did not show improvement after further training with early stopping. However, the performance of the QARR-T5 model increased from 80.77% to 81.32%, and the QARR-LED model improved from 74.73% to 76.26%.

TABLE 11. Performance of the pretrained models with different DROP dataset variants.

Model	Template	F1 Score (%)	EM (%)	Time (mins)
BART	DROP	74.46	63.68	85.98
	DROP + QARR (Predefined Template)	75.04	63.47	157.53
	DROP + QARR (Modified Template 1)	75.17	63.55	187.17
	DROP + QARR (Modified Template 2)	**75.24**	**63.87**	**186.43**
T5	DROP	76.94	66.99	222.45
	DROP + QARR (Predefined Template)	79.67	68.84	321.55
	DROP + QARR (Modified Template 1)	80.56	69.97	326.76
	DROP + QARR (Modified Template 2)	**80.77**	**70.06**	**364.30**
LED	DROP	74.70	63.23	424.33
	DROP + QARR (Predefined Template)	74.58	63.12	793.23
	DROP + QARR (Modified Template 1)	74.57	63.01	769.04
	DROP + QARR (Modified Template 2)	**74.73**	**63.02**	**808.41**

TABLE 12. F1 scores (%) of the models with optimal hyperparameter settings after training with early stopping.

Model	Learning Rate	QARR Ratio	Template	F1 Score (%)
QARR-BART	$2e^{-6}$	0.25	Modified Template 2	75.24
QARR-T5	$2e^{-5}$	0.25	Modified Template 2	81.32
QARR-LED	$2e^{-6}$	0.10	Modified Template 2	76.26

D. COMPARISON RESULTS ON QARR FINE-TUNED MODEL AND EXISTING WORK

After obtaining the best pretrained models, each model undergoes a fine-tuning process on the SQuAD, HotpotQA, and NQ datasets. Each experiment is repeated five times to ensure robustness and reliability. The F1 score is used as the evaluation metric to assess the models' performance, and the standard deviation of the F1 scores from the five experiments is calculated to measure consistency. This repetition mitigates the fluctuations inherent in few-shot training, as the learning process can be sensitive to specific samples in the training set. By repeating the experiments, a more stable estimate of the models' performance is obtained, reducing the impact of random variations in the training data. For each of the models, the learning rate during the fine-tuning stage follows the optimal setting determined in the pretraining phase: a learning rate of $2e^{-5}$ for the T5 model and $2e^{-6}$ for both the BART and LED models.

Table 13 presents the comparison of the mean and standard deviation of F1 scores between the fine-tuned models and existing works. The results indicate that the QARR-fine-tuned model exhibits higher performance compared to existing works. QARR-T5 demonstrates the best performance

in 11 out of 12 experimental settings, while QARR-LED achieves the best performance in the 16-shot subset of the HotpotQA dataset. Specifically, QARR-T5 achieved the highest F1 scores of 81.7% in 16-shot and 32-shot, 82.7% in 64-shot, and 84.5% in 128-shot on the SQuAD dataset. For the HotpotQA dataset, QARR-LED records the highest F1 score of 59.3% in 16-shot, while QARR-T5 obtains the highest F1 scores of 60.2%, 61.5%, and 63.6% in the 32-shot, 64-shot, and 128-shot settings, respectively. QARR-T5 similarly excels on the NQ dataset with the highest F1 scores of 56.7% in 16-shot, 57.1% in 32-shot, 60.1% in 64-shot, and 63.0% in 128-shot settings.

The proposed QARR framework outshines existing work due to several key characteristics. Firstly, the Question-Answer Replacement and Removal (QARR) technique effectively augments the dataset by integrating answers into questions and selectively removing words. This augmentation helps models generalize better by exposing them to varied question-answer formats during pretraining. Secondly, different templates for question-answer pairs were introduced, allowing the models to learn from diverse patterns and structures, enhancing their adaptability to new data. Thirdly, models like BART, T5, and LED underwent extensive pretraining on the augmented dataset with specific architectures and optimization objectives, resulting in a strong initial capability to handle few-shot scenarios. Fourthly, the fine-tuning on few-shot datasets was meticulously repeated to derive average results, ensuring that the models' performance was reliable and consistent. This process highlighted the models' ability to transfer learned knowledge effectively. Lastly, the low standard deviation in F1 scores across multiple runs indicates that the QARR framework provides stable and reproducible performance improvements, essential in few-shot learning scenarios.

Table 14 presents the EM scores and time taken by the proposed fine-tuned models. The results show that the QARR-T5 model achieves the highest performance on both the SQuAD and Natural Questions datasets in terms of EM scores, indicating its effectiveness in extracting precise information and generating accurate responses. Additionally, the QARR-LED model excels on the HotpotQA dataset, particularly in tasks requiring multi-hop reasoning to generate concise answers.

In contrast, the QARR-BART model is notable for its time efficiency. The reported time represents the total duration for five repeated experiments, each involving training over 25 epochs and inference. The results indicate that QARR-BART completes these experiments faster than the other models. This efficiency is likely due to the QARR-BART model's fewer parameters, which reduce computational demands during training and inference, making it an ideal choice when time and computational resources are limited.

TABLE 13. Comparative results of the fine-tuned models and existing works, measured in F1 scores.

	Method	SQuAD	HotpotQA	NQ
16-shot	Mlspan	54.9	40.9	42.1
	FewShotQA	55.5±2.0	45.1±1.8	45.1±2.3
	GOTTA	57.8±2.6	45.9±1.7	47.1±1.1
	QARR-BART	75.2±0.4	58.3±0.2	54.9±1.1
	QARR-T5	**81.7±0.6**	59.0±0.2	**56.7±0.7**
	QARR-LED	75.8±0.4	**59.3±0.4**	53.3±0.2
32-shot	Mlspan	57.9	44.8	44.8
	FewShotQA	56.8±2.1	47.9±1.4	50.1±1.1
	GOTTA	62.7±1.8	49.6±1.3	49.6±1.3
	QARR-BART	76.3±0.3	59.2±1.2	56.4±1.7
	QARR-T5	**81.7±0.7**	**60.2±1.1**	**57.1±0.5**
	QARR-LED	76.6±0.3	60.0±0.7	54.6±1.0
64-shot	Mlspan	62.7	54.5	49.3
	FewShotQA	61.5±2.3	51.2±1.0	53.0±0.5
	GOTTA	67.7±0.9	52.0±0.8	51.5±1.3
	QARR-BART	76.9±1.3	61.1±1.0	57.8±1.9
	QARR-T5	**82.7±0.6**	**61.5±0.5**	**60.1±2.0**
	QARR-LED	77.1±1.1	61.0±0.6	55.2±1.4
128-shot	Mlspan	71.0	59.6	53.0
	FewShotQA	68.0±0.3	54.8±0.8	53.9±0.9
	GOTTA	71.3±1.3	56.3±1.4	54.2±0.7
	QARR-BART	80.5±0.2	63.3±0.9	61.0±0.5
	QARR-T5	**84.5±0.5**	**63.6±1.1**	**63.0±0.7**
	QARR-LED	79.1±0.8	63.0±0.7	57.8±0.6

TABLE 14. Exact Match score and time used for fine-tuned models.

Shot	Model	Dataset	EM (%)	Time (mins)
16	QARR-BART	SQuAD	62.84±1.12	42.65
		HotpotQA	42.31±0.14	32.69
		NQ	40.60±1.78	44.26
	QARR-T5	**SQuAD**	**69.92±0.87**	**150.72**
		HotpotQA	42.62±0.08	65.72
		NQ	**42.00±0.90**	**91.32**
	QARR-LED	SQuAD	62.95±0.93	60.52
		HotpotQA	**43.78±0.40**	**46.01**
		NQ	37.85±0.51	69.92
32	QARR-BART	SQuAD	64.72±0.72	44.24
		HotpotQA	43.20±1.12	34.22
		NQ	41.63±2.57	51.64
	QARR-T5	**SQuAD**	**70.01±0.91**	**87.36**
		HotpotQA	43.78±1.09	73.08
		NQ	**42.69±0.21**	**96.41**
	QARR-LED	SQuAD	64.18±0.38	67.25
		HotpotQA	**44.52±0.56**	**52.43**
		NQ	39.26±1.19	79.78
64	QARR-BART	SQuAD	64.79±1.61	50.98
		HotpotQA	44.94±0.93	39.74
		NQ	42.76±2.38	58.39
	QARR-T5	**SQuAD**	**71.22±1.31**	**100.50**
		HotpotQA	44.94±0.67	83.90
		NQ	**45.31±2.17**	**109.90**
	QARR-LED	SQuAD	64.22±2.1	86.56
		HotpotQA	**45.38±0.54**	**68.26**
		NQ	39.62±1.89	96.07
128	QARR-BART	SQuAD	70.12±0.61	61.00
		HotpotQA	46.92±0.85	54.16
		NQ	46.88±0.70	67.43
	QARR-T5	**SQuAD**	**74.58±1.08**	**254.41**
		HotpotQA	**47.09±1.19**	**106.66**
		NQ	**48.95±0.92**	**138.89**
	QARR-LED	SQuAD	68.02±1.32	159.84
		HotpotQA	47.07±0.57	98.72
		NQ	42.70±1.03	126.17

E. CASE STUDY

Table 15 shows a comparison between the answers generated by the model with and without the implementation of the QARR technique during the pretraining stage. In this case, the model without the QARR technique generates the incorrect answer `Super Bowl L`. The generated answer is highly

similar to the ground truth but reflects a misunderstanding or misinterpretation of the context. This error arises from a failure to properly link the details in the context.

Conversely, the answer generated with the QARR technique correctly identifies ''Super Bowl LI'' as the game where Roman numerals will be used again. This indicates that the QARR technique enhances comprehension and retrieval processes, enabling the correct extraction of relevant details from the context to answer the question accurately.

TABLE 15. Example with and without QARR technique.

Context	On June 4, 2014, the NFL announced that the practice of branding Super Bowl games with Roman numerals, a practice established at Super Bowl V, would be temporarily suspended, and that the game would be named using Arabic numerals as Super Bowl 50 as opposed to Super Bowl L. The use of Roman numerals will be reinstated for Super Bowl LI. Jaime Weston, the league's vice president of brand and creative, explained that a primary reason for the change was the difficulty of designing an aesthetically pleasing logo with the letter "L" using the standardized logo template introduced at Super Bowl XLV. The logo also deviates from the template by featuring large numerals, colored in gold, behind the Vince Lombardi Trophy, instead of underneath and in silver as in the standard logo.	
Question	Question: Which Super Bowl, after the 50th one, will begin to have Roman numerals in the title again?	
Answer	Answer without QARR technique	Answer with QARR technique
	Super Bowl L	Super Bowl LI

F. ERROR ANALYSIS

This section analyzes the errors observed in the performance of the QARR model. Table 16 breaks down the errors into four distinct categories based on one experiment involving QARR-T5 in a 128-shot setting. Only answers that exactly match the gold answer are considered correct, with 75.52% of the generated answers falling into this category.

Of the 24.48% incorrect answers, 5.34% are due to over-informative responses. Over-informative errors occur when the model includes more information than required, such as generating ''17 interceptions'' instead of the gold answer ''17''. This type of error is common in datasets requiring concise answers. Conversely, 5.97% of the errors are under-informative, arising when the model omits essential details, like generating ''experience'' instead of ''more experience and higher education''.

Additionally, numeric errors account for just 0.04% of the total, typically involving discrepancies in format, such as generating ''2'' instead of ''two''. Although semantically correct, the format mismatch leads to an error. The low percentage in this category is due to most numeric answers being either correct or falling into the over-informative category.

Finally, 13.13% of the errors involve unrelated answers, where the model generates responses entirely disconnected from the expected answer. This highlights instances where the model fails to produce any relevant information in response to the question.

TABLE 16. Error analysis on one of the experiment result of QARR-T5 in 128-shot setting.

Category	Percentage (%)
Correct Answer	75.52
Incorrect Answer	24.48
Over-Informative	5.34
Under-Informative	5.97
Numeric Problem	0.04
Unrelated Answer	13.13

IX. CONCLUSION

This paper presents a novel pretraining framework to enhance few-shot question answering (FSQA) capabilities. By employing the Discrete Reasoning Over the Content of Paragraphs (DROP) dataset and the innovative Question-Answer Replacement and Removal (QARR) technique, the framework significantly improves the performance of models like BART, T5, and LED. Through extensive experiments and comparisons, the QARR-T5 model consistently outperformed state-of-the-art FSQA methods, demonstrating the highest F1 scores across various few-shot scenarios on the SQuAD dataset. The findings underscore the framework's effectiveness in enhancing models' generalization and performance on new datasets with limited samples, advancing the field of few-shot QA.

Future work can explore the application of this framework to other QA datasets, broadening its utility across various domains and question types. While the current approach, which utilizes a specific QARR ratio, has proven effective, it may sometimes limit the naturalness of the generated sentences. To address this, future research could integrate deep learning methods that dynamically adjust the QARR ratio based on the question and its type. This adjustment would allow for more detailed and contextually appropriate sentence generation, resulting in more natural and coherent questions that better simulate real-world language.

REFERENCES

[1] A. Patel, B. Li, M. S. Rasooli, N. Constant, C. Raffel, and C. Callison-Burch, ''Bidirectional language models are also few-shot learners,'' 2022, *arXiv:2209.14500*.
[2] B. Dhingra, D. Pruthi, and D. Rajagopal, ''Simple and effective semi-supervised question answering,'' 2018, *arXiv:1804.00720*.
[3] X. Chen, Y. Zhang, J. Deng, J.-Y. Jiang, and W. Wang, ''GOTTA: Generative few-shot question answering by prompt-based cloze data augmentation,'' in *Proc. SIAM Int. Conf. Data Mining (SDM)*, 2023, pp. 909–917.
[4] J. Wang, C. Wang, M. Qiu, Q. Shi, H. Wang, J. Huang, and M. Gao, ''KECP: Knowledge enhanced contrastive prompting for few-shot extractive question answering,'' 2022, *arXiv:2205.03071*.
[5] R. Chada and P. Natarajan, ''FewshotQA: A simple framework for few-shot learning of question answering tasks using pre-trained text-to-text models,'' 2021, *arXiv:2109.01951*.

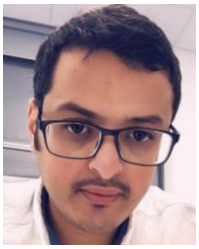
[6] O. Castel, O. Ram, A. Efrat, and O. Levy, "How optimal is greedy decoding for extractive question answering?" 2021, *arXiv:2108.05857*.
[7] O. Ram, Y. Kirstain, J. Berant, A. Globerson, and O. Levy, "Few-shot question answering by pretraining span selection," 2021, *arXiv:2101.00438*.
[8] Z. Li, W. Wang, L. Dong, F. Wei, and K. Xu, "Harvesting and refining question-answer pairs for unsupervised QA," 2020, *arXiv:2005.02925*.
[9] U. Zaratiana, N. El Khbir, D. Núñez, P. Holat, N. Tomeh, and T. Charnois, "DyREx: Dynamic query representation for extractive question answering," 2022, *arXiv:2210.15048*.
[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[11] P. Banerjee and C. Baral, "Self-supervised knowledge triplet learning for zero-shot question answering," 2020, *arXiv:2005.00316*.
[12] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," *Appear*, vol. 7, no. 1, pp. 411–420, 2017.
[13] K. Ma, F. Ilievski, J. Francis, Y. Bisk, E. Nyberg, and A. Oltramari, "Knowledge-driven data construction for zero-shot evaluation in commonsense question answering," in *Proc. Conf. Artif. Intell. (AAAI)*, 2021, pp. 13507–13515.
[14] C. Lyu, L. Shang, Y. Graham, J. Foster, X. Jiang, and Q. Liu, "Improving unsupervised question answering via summarization-informed question generation," 2021, *arXiv:2109.07954*.
[15] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave, "Atlas: Few-shot learning with retrieval augmented language models," 2022, *arXiv:2208.03299*.
[16] P. Lewis, L. Denoyer, and S. Riedel, "Unsupervised question answering by cloze translation," 2019, *arXiv:1906.04980*.
[17] A. R. Fabbri, P. Ng, Z. Wang, R. Nallapati, and B. Xiang, "Template-based question generation from retrieved sentences for improved unsupervised question answering," 2020, *arXiv:2004.11892*.
[18] D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner, "DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs," 2019, *arXiv:1903.00161*.
[19] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019, *arXiv:1910.13461*.
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.
[21] C. Raffel, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.
[22] J. Lei Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[23] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," 2020, *arXiv:2004.05150*.
[24] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*.
[25] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," 2016, *arXiv:1606.05250*.
[26] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "HotpotQA: A dataset for diverse, explainable multi-hop question answering," 2018, *arXiv:1809.09600*.
[27] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov, "Natural questions: A benchmark for question answering research," *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 453–466, Nov. 2019.
[28] A. Fisch, A. Talmor, R. Jia, M. Seo, E. Choi, and D. Chen, "MRQA 2019 shared task: Evaluating generalization in reading comprehension," 2019, *arXiv:1910.09753*.

SIAO WAH TAN received the bachelor's degree (Hons.) in information technology with a specialization in artificial intelligence from Multimedia University, Malaysia, in 2022. He is currently pursuing the M.Sc. degree in information technology, focusing on natural language processing. His current research interests include question answering and few-shot learning.

CHIN POO LEE (Senior Member, IEEE) received the Master of Science and Ph.D. degrees in information technology, with a specialization in video-based abnormal behavior detection and gait recognition. She is currently an Associate Professor with the School of AI and Advanced Computing, Xi'an Jiaotong-Liverpool University, China. Her research interests include computer vision, natural language processing, and deep learning.

KIAN MING LIM (Senior Member, IEEE) received the B.IT. degree (Hons.) in information systems engineering, the Master of Engineering Science (M.Eng.Sc.) degree, and the Ph.D. degree in IT from Multimedia University. He is currently an Associate Professor with the School of Computer Science, University of Nottingham Ningbo China. His research and teaching interests include machine learning, deep learning, computer vision, and pattern recognition.

CONNIE TEE (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in IT from Multimedia University, in 2005 and 2015, respectively. She has been an Associate Professor with the Faculty of Information Science and Technology, Multimedia University, since 2021. She is currently holding the position of the Dean of the Institute for Postgraduate Studies. Her research interests include computer vision, machine learning, deep learning, and image processing.



ALI ALQAHTANI received the Ph.D. degree in computer science from Swansea University, Swansea, U.K., in 2021. He is currently an Assistant Professor with the Department of Computer Science, King Khalid University, Abha, Saudi Arabia. He has published several refereed conference and journal publications. His research interests include aspects of pattern recognition, deep learning, and machine intelligence and their applications to real-world problems.