Empowering Multi-Span Question Answering with Expansive Information Injection using Large Language Models

Zhiyi Luo, Yingying Zhang, Ying Zhao, Shuyun Luo* and Wentao Lv

[a]*School of Computer Science and Technology and the Key Laboratory of Intelligent Textile and Flexible Interconnection of Zhejiang Province, Zhejiang Sci-Tech University*

[b]*No. 928, No. 2 street, Baiyang street, Qiantang New District, Hangzhou 310018, China*

ABSTRACT

Multi-span question answering has gained prominence as it aligns more closely with real-world user requirements compared to single-span question answering. The utilization of pretrained language models has shown promise in improving multi-span question answering, particularly for factoid questions that necessitate entity-based answers. However, existing methods tend to overlook critical information regarding answer span boundaries, resulting in limited accuracy when generating descriptive answers. To address this limitation, we propose TOAST, a novel joint learning framework specialized in token-based neighboring transitions that capture answer span boundaries through adjacent word relations. Our approach extracts high-quality multi-span answers and is general-purpose, applicable to both alphabet languages like English and logographic languages like Chinese. Furthermore, we introduce CLEAN, a comprehensive open-domain Chinese multi-span question answering dataset, which includes a substantial number of descriptive questions. Extensive experiments demonstrate the superior performance of TOAST over previous top-performing QA models in terms of both EM F1 and overlapped F1 scores. Specifically, the TOAST models, leveraging BERT$_{base}$ and RoBERTa$_{base}$, achieve substantial improvements in EM F1 scores, with increments of 3.03/2.13, 4.82/3.73, and 16.26/11.53, across three publicly available datasets, respectively.

1. Introduction

Let's cite a paper (Pang, Lan, Guo, Xu, Su and Cheng, 2019).

In summary, the main contributions in this paper are as follows:

- We employs a automatic data augmentation framework using Large Language Model (***LLM***) as a knowledge source and a extra content supplement to linearize relevant information and possible continuation from LLM as texts, then inject them into original contexts.
- We develop a series of prompt templates designed for interacting with ChatGPT to acquire comprehensive explanations of numerous entities. These templates ensure that the formats of the responses provided by ChatGPT are highly parseable and well-structured.

2. Related Work

This section briefly covers past research on employing Large Language Models (LLMs) like GPT for knowledge enhancement in diverse applications, emphasizing different approaches in extractive Reading Comprehension (RC) tasks.

2.1. Interaction with LLMs

Recent research emphasizes the significance of extracting knowledge from Large Language Models (LLMs) in domains characterized by limited knowledge base coverage (Fang, Wang, Xu, Xu, Sun, Zhu and Zeng, 2021). GPT3Mix (Yoo, Park, Kang, Lee and Park, 2021) extends the utility of LLMs to text data augmentation, thereby elevating the performance of machine learning models. AugGPT (Dai, Liu, Liao, Huang, Cao, Wu, Zhao, Xu, Liu, Liu, Li, Zhu, Cai, Sun, Li, Shen, Liu and Li, 2023) further contributes to natural language processing tasks in scenarios with constrained data by generating new textual data through interactions with ChatGPT.

In the realm of common-sense reasoning, the effectiveness of prompt generation and reinforcement learning has been demonstrated in enhancing question-answering performance (Liu, Liu, Lu, Welleck, West, Bras, Choi and Hajishirzi, 2021; Liu, Hallinan, Lu, He, Welleck, Hajishirzi and Choi, 2022). LLMs excel in the generation of contextual information (Yu, Iter, Wang, Xu, Ju, Sanyal, Zhu, Zeng and Jiang, 2022), and the collaborative impact of iterative retrieval and generation amplifies overall LLM performance (Shao, Gong, Shen, Huang, Duan and Chen, 2023).

Within question-answering systems, (Huang, Zhou, Xiao and Cheng, 2023b) improve model context learning in multi-span tasks through answer feedback mechanisms. Comparative analyses between ChatGPT and traditional knowledge base question-answering models explore its potential as an alternative (Tan, Min, Li, Li, Hu, Chen and Qi, 2023). In addressing LLM limitations

*Corresponding author

luozhiyi@zstu.edu.cn (Z. Luo); 272831920@qq.com (Y. Zhang); 308956149@qq.com (Y. Zhao); shuyunluo@zstu.edu.cn (S. Luo); alvinlwt@zstu.edu.cn (W. Lv)

http://zhiyiluo.site (Z. Luo)

ORCID(s): 0000-0002-2206-1926 (Z. Luo)

in handling factual information, (Ren, Wang, Qu, Zhao, Liu, Tian, Wu, Wen and Wang, 2023) employ retrieval-enhanced techniques to enhance the model's understanding of fact-based queries.

(Wei, Cui, Cheng, Wang, Zhang, Huang, Xie, Xu, Chen, Zhang et al., 2023) showcase zero-shot information extraction through ChatGPT interaction, while (Wei et al., 2023) explore enhancing content-based recommendation systems by combining open-source and closed-source language models. Collectively, these studies highlight LLM innovation across diverse domains, offering valuable insights for future research.

However, challenges persist, including non-standardized response structures and ambiguity. Our research focuses on designing a set of templates for interacting with ChatGPT, encompassing entity interpretation, entity relationship analysis, rewriting, and summary information for each task. Additionally, we introduce a knowledge injection method to directly incorporate knowledge generated by large models into training texts, thereby enhancing task performance. This approach aims to provide a structured and comprehensive framework for leveraging LLMs in various applications.

2.2. Neural Models for RC

Research in reading comprehension grows rapidly, and many successful neural-based RC models have been proposed in this area. Typically, neural models (Pang et al., 2019; Wang and Jiang, 2017; Xiong, Zhong and Socher, 2017) for RC are composed of two components, a context encoder and an answer decoder. The context encoder is used to encode the information of questions, contexts and their interactions in-between. Then, the answer decoder aims to generate the answer texts based on outputs of the context encoder. To make the answer decoder compatible with the answer extraction task, Pointer Network (Vinyals, Fortunato and Jaitly, 2015) model has been adopted to copy tokens from the given contexts as answers (Kadlec, Schmid, Bajgar and Kleindienst, 2016; Trischler, Ye, Yuan, Bachman, Sordoni and Suleman, 2016). Wang and Jiang (2017) proposed a boundary model, which utilized Pointer Network to predict the start and end indices for an answer span. Seo, Kembhavi, Farhadi and Hajishirzi (2017) proposed an alternative way for the implementation of answer decoder, that built neural position classifiers upon the encoder outputs, predicting the start and end indices of the answer span in the context.

Recently, the RC models upgrade the context encoder using pre-trained language models (PrLMs) (Radford, Narasimhan, Salimans, Sutskever et al., 2018; Kenton and Toutanova, 2019; Liu, Ott, Goyal, Du, Joshi, Chen, Levy, Lewis, Zettlemoyer and Stoyanov, 2019; Lee, Yoon, Kim, Kim, Kim, So and Kang, 2020; Gu, Tinn, Cheng, Lucas, Usuyama, Liu, Naumann, Gao and Poon, 2021) , benefiting from the invention of Transformer (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser and Polosukhin, 2017) blocks. Devlin, Chang, Lee and Toutanova (2019) proposed a standard extractive model for single-span RC that utilizes BERT to encode inputs, then builds position classifiers to predict where the answer span starts and ends. However, the answer decoder, whether implemented with Pointer Network or position classifiers, predicts start and end position independently, thus can not distinguish the different answer spans properly. Zhu, Ahuja, Juan, Wei and Reddy (2020) proposed MultiCo which used a contextualized sentence selection method to capture the relevance among multiple sentence-based answer spans in order to form an answer with multiple sentences. These models are not well adapted to multi-span RC which can be formulated as more flexible task of multi-span extraction where each span can be a word, phrase, sentence or any continuous string of text.

Extracting a variable number of spans from an input text can be commonly cast as a sequence tagging problem. Segal, Efrat, Shoham, Globerson and Berant (2020) proposed using a sequence tagging model for multi-span extraction, which predicts whether each token is part of an answer. Yoon, Jackson, Lagerberg and Kang (2022) employed a similar sequence tagging approach to address extractive question answering (Naseem, Dunn, Khushi and Kim, 2022) in the biomedical domain. Li, Tomko, Vasardani and Baldwin (2022) also adopted the tagging model architecture, integrating two sub-tasks: predicting the number of spans to extract and annotating the answer structure within their proposed dataset to capture global information. ADRAV (Hu, Yang, Li, Sun and Yang, 2023) proposed a dynamic routing and answer voting method to further make full use of the hidden layer knowledge of pre-trained models.

More recently, SpanQualifier (Huang, Zhou, Niu and Cheng, 2023a) improves Multi-Span Reading Comprehension (MSRC) by explicitly representing spans and modeling interactions within and between them. Iterative Extractor (Zhang, Lin, Liu, Lai, Feng and Zhao, 2023) introduces a classification method for MSRC instances, prompting exploration of strategies to maximize different paradigms' advantages in capturing key information and modeling relationships between questions and context. While these methods leverage powerful context encoders (PrLMs) with good multi-span answer extraction performance, they fall short in fully harnessing external knowledge, limiting their text comprehension abilities. LIQUID (Lee, Kim and Kang, 2023) was designed to automatically generate list-style QA pairs from unlabeled corpora, using named entities from summarized text as candidate answers and incorporating synthetic data in the tagging model. However, its focus on list QA data narrows the scope of knowledge and lacks specificity, reducing efficiency. In contrast, AUG significantly advances the field by

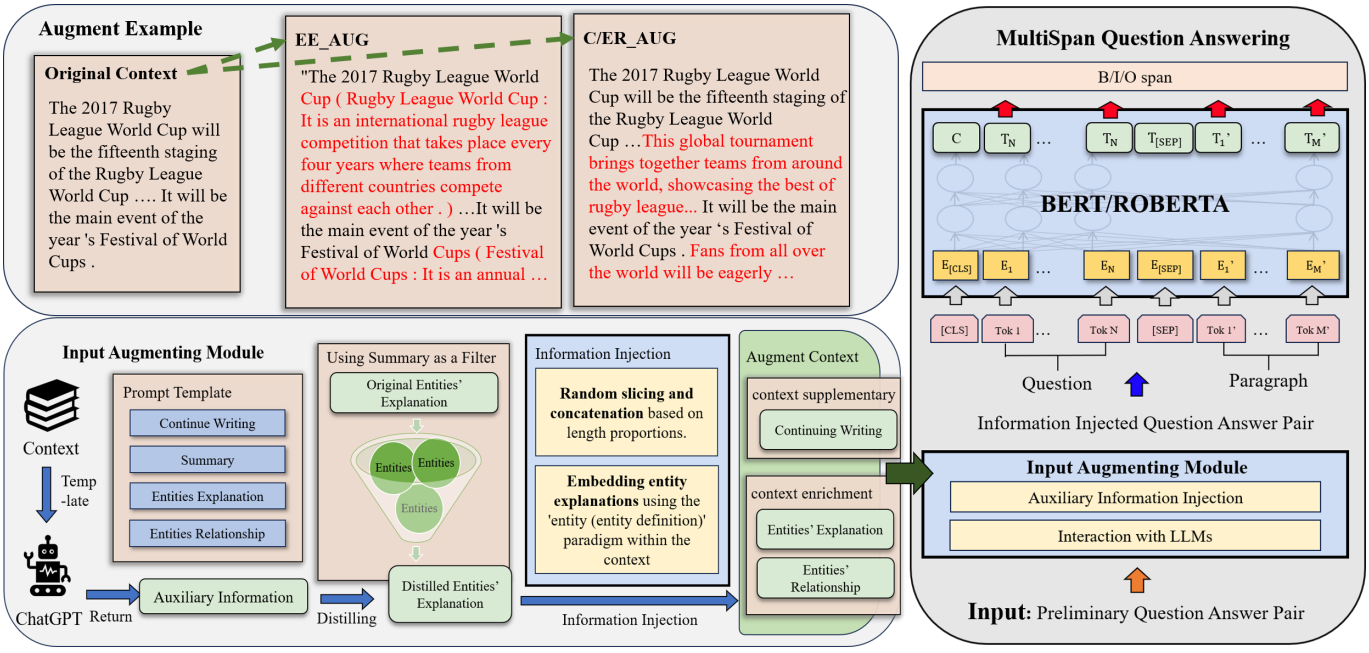

Figure 1: An overview of our automatic information augmentation framework. **(a) Step 1**: Interact with ChatGPT to Get Auxiliary Information. **(b) Step 2**: Distilling the Information and Injecting Them into Context **(c) Step 3**: Input the Augmented Context into Tagging Model

efficiently enhancing model training. It achieves this by leveraging interactions with a large language model to acquire contextually relevant knowledge directly tied to the data, which can be seamlessly incorporated into the finetuning.

3. Approach

3.1. Our Framework

Given a question $Q_i \in Q = \{Q_1, ..., Q_n\}$ and a context $C_i \in C = \{C_0, ..., C_n\}$, where C_i contains m tokens $c_0, .., c_m$, the objective of multi-span question answering is to identify a set of answer spans $A_i = \{a_0, ..., a_s\}$ within the context. Here, each answer span $a_j \in A_i$ is represented as $a_j = c_{s_l}, ..., c_{e_l}$, where s_l and e_l denote the start and end positions of the l-th answer span, respectively. Following the observation of Li et al. (2022), We adopt the BIO tagging scheme to mark answer spans in the context where words are tagged as either part of the answer(**B**egin, **I**nside) or not (**O**ther). Formally, BIO tagging scheme is represented by a tag set $\tau = \{B, I, O\}$.

Intuitively, large language models (LLMs) can serve as supplementary sources of external knowledge, compensating for the restricted semantic comprehension and limited information perception inherent in pre-trained language models). We then propose a novel **G**enerative **I**nformation **A**ugme**NT**ation framework (GIANT) for multi-span question answering. GIANT employs a plug-and-play strategy to integrate the knowledge from large language models into the input layer of tagging models, built upon pre-trained language models.

The overview of GIANT is depicted as Figure 1. This process comprises the following steps: **1)Prompting**: constructing instruction templates to leverage language models for generating diversified data, involving entity elucidation, entity relationships, content continuation, and summarization. **2)Generating**: the large language model generating new knowledge based on the designed prompts. **3)Updating**: filtering the generated data, and combine it with metadata in different forms. **4)Training**: employing the knowledge-enhanced data to train a tagging model.

In the following section, we will explore how GIANT utilizes a large language model to generate knowledge. Subsequently, we will delve into the methods employed by GIANT to filter these knowledge, amalgamate them into meta-context, and orchestrate ensemble strategies to leverage these knowledge effectively.

提示的模板放在哪里描述比较好?

3.2. LLMs as Knowledge Source

GIANT leverages a large language model as an external knowledge source, utilizing it to generate augmented data $K_i = \{E_i, S_i, R_i, F_i\}$ from multiple perspectives. Here, E_i represents named entity elucidation, R_i denotes entity relations, S_i pertains to content summarization, and F_i encompasses content continuation.

Within these knowledge perspectives, named entity elucidation and entity relations provide factual knowledge, with GIANT synthesizing knowledge cues for

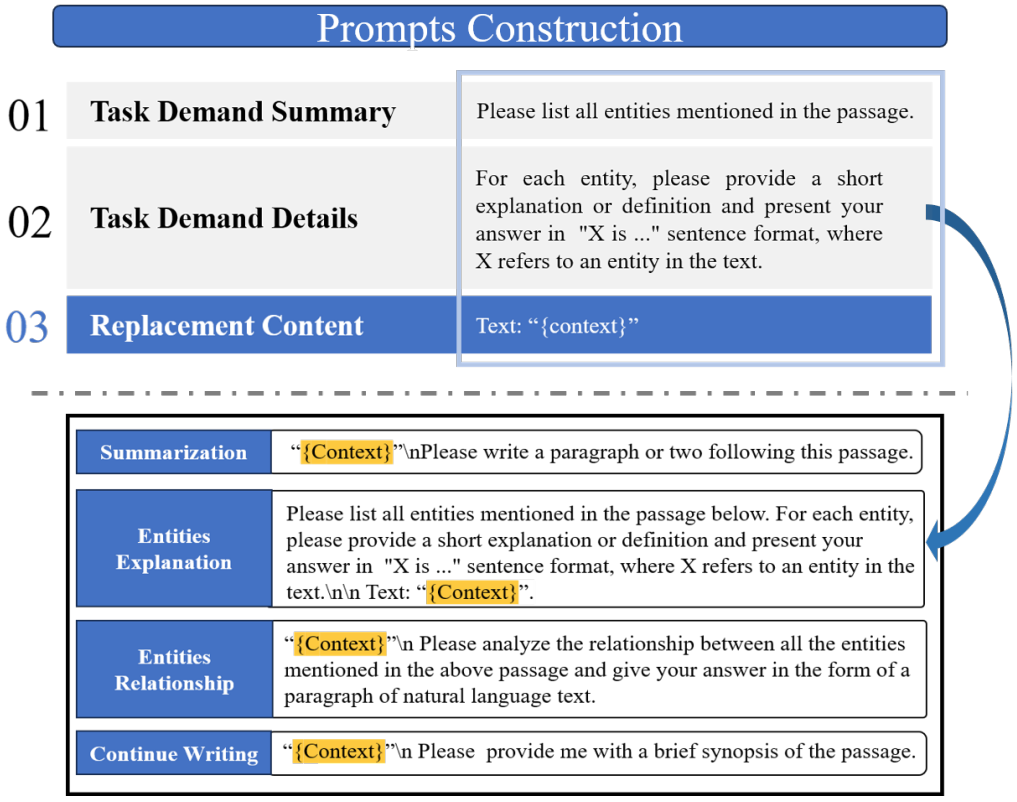

Figure 2: Prompt Templates and it's Construction

the model by jointly injecting them into metadata. Summarization acts as a mechanism for information filtering, sieving and retaining entity interpretations and analyses of entity relationship, thereby ensuring augmentation efficiency. Content continuation introduces relevant external knowledge by extending contextual content.

3.2.1. LLMs as Content Summerizer

One of the pivotal factors in entity knowledge selection lies in ensuring that these entities exhibit semantic relevance within the specified context. The incorporation of disparate entity explanations and relationship analyses at the model's input layer may lead to the valueless augmentation of contexts, consequently impeding the model's learning.

Utilizing the potent capabilities of large language models, we employ them to succinctly summarize the context C_i into S_i, followed by extracting entities-based factual knowledge from S_i. By considering summarization as a knowledge filter, we can achieve information augmentation with greater efficiency.

3.2.2. LLMs as Named Entity Elucidator

The resolution of multi-span questions usually entails identifying named entities. This intricate task can be greatly improved by tapping into the specialized knowledge of named entities. This expertise directly aids the model in comprehending rare lexical items within its pre-training corpus and adapting to context-specific terminologies during fine-tuning.

While previous research has delved into entity knowledge either through training or direct application of fine-tuned NER models like spaCy and BERN2, these models have limitations stemming from their narrow training datasets and their tendency to solely provide extracted entities without contextualized meanings due to their inflexible design. In comparison, generative large language models, endowed with advanced representation capabilities and abundant training data, present a promising avenue for achieving more precise and comprehensive entity delineations.

GIANT utilizes a large language model to generate entity elucidation E_i represented as $\{e_0, ..., e_h\}$, from summary S_i ,from summaries S_i within each context C_i, where i ranges from 1 to n. Following this, it adopts a hybrid methodology integrating both regular expressions and semantic dependency analysis models, exemplified by spaCy, to parse the produced text. This systematic procedure culminates in the establishment of a "Entity-Elucidation" knowledge base oriented towards the elucidation of entities. Furthermore, as Figure 3 presented, we seamlessly integrates the retrieved entity $e_t \in E_i$ mentioned within context C_i by inserting its corresponding explanation immediately after the entity mention, significantly enriching the

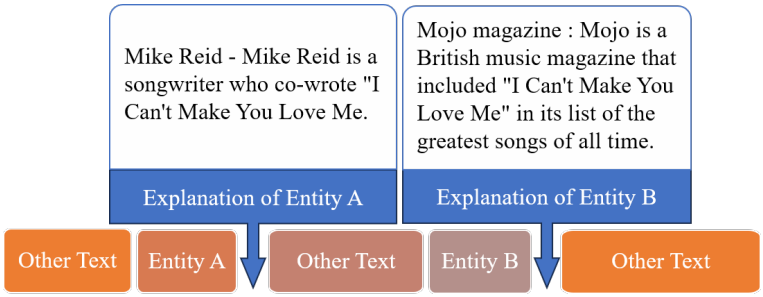

Figure 3: The Process of Inserting Entity Explanation into Context

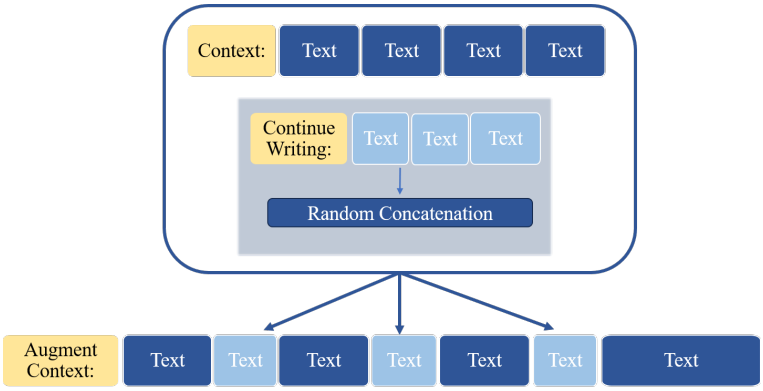

Figure 4: The Process of Concatenation of Original Context and Auxiliary Information

original text content while ensuring coherence and clarity are upheld.

3.2.3. LLMs as Entity Relationship Extractor

Besides, understanding entity relations can also assist the model in untangling intricate logical relationships within complex context, thereby facilitating the clarification of interconnected concepts focused on the question-and-answering process. With the robust language comprehension capabilities, contextual sensitivity, and extensive external knowledge, large language models can adeptly capture entity associations within extensive contents and transform this structured knowledge into natural language expressions, thus integrating factual knowledge into the model in textual form. Moreover, since entity interpretation and entity relationships are both factual knowledge, they naturally exhibit consistency, thus leading to their integration while GIANT injecting multiple knowledge into original context.

After extracting the textual entity relationships R_i from the summary S_i of each context C_i, to maximize the interaction between context C_i and relational knowledge R_i, we first calculate the ratio r_l, which is defined as the quotient of the cumulative length of the generated text R_i and the total length of the original context C_i. As depicted in Figure 4, we subsequently partition the enhanced text R_i and the original context C_i into several segments, denoted as $\tilde{R}_i = \{\hat{r_1}, \hat{r_2}, ..., \hat{r_w}\}$ and $\tilde{C}_i = \{\hat{t_1}, \hat{t_2}, ..., \hat{t_w}\}$, respectively. After partitioning, the number of enhanced text segments $|\tilde{R}_i|$ equals the number of original context segments $|\tilde{C}_i|$, and the ratio of the length of each enhanced text segment $|t_i|$ to each original context segment $|c_i|$ is consistent with r_l. Next, we maintain the order of the original context segments while shuffling the order of the enhanced text segments, represented as $R'_i = \{r_{\sigma(1)}, r_{\sigma(2)}, ..., r_{\sigma(m)}\}$, where σ is a random permutation of the indices of the enhanced text segments. Then, we sequentially insert the shuffled enhanced text segments into the original text segments. This concatenation results in a new context $C' = \{c_1, r_{\sigma(1)}, c_2, r_{\sigma(2)}, ..., c_n, r_{\sigma(n)}\}$ enriched with relational knowledge.

3.2.4. LLMs as Content Continuator

In addition to extracting and organizing knowledge from given contexts, we can enhance contexts by introducing external knowledge through the expansion of textual content. By continuing to write the original context, we enrich its content and themes, indirectly providing knowledge cues to models, thereby enabling model to learn more task-relevant information during fine-tuning.

With extensive training data and parameters, large language models possess vast knowledge and domain backgrounds, exhibiting strong language expression capabilities and a deep understanding of context. Consequently, they can generate coherent and informative

text extensions based on contexts. Such continuations not only improve text coherence but also enhance information content.

The method of injecting extension knowledge is similar to injecting entity relationships into the original context. It involves randomly slicing the content continuation and inserting segmented text pieces into the original text, facilitating full interaction between the original and enhanced texts, which results in further enriching the semantics and content of the text.

4. Experiments

In this section, we compare our information augmentation approach with multiple strong baseline on multi-span question answering. We first introduce the datasets and experiment setup, then show the experimental results and analysis for different model.

4.1. Evaluation Dataset

We conducted experiments on MultiSpanQA(Li et al., 2022), a recently introduced Reading Comprehension dataset designed for multi-span question answering. This dataset comprises 6.5K multi-span examples in which the questions represent user queries issued to the Google search engine, and the contexts are extracted from the English Wikipedia. It's worth noting that there is a expand variant of MultiSpanQA known as MultispanQA(expand), which intakes single-span and answerable questions. However, we did not perform a comparison with the expanded dataset due to its relatively lower proportion of multi-span QA pairs.

4.2. Experimental Setup

For all competing models and our model, we use the HuggingwadFace implementation of BERT_{base} or RoBERTa_{base} as the *encoder* with $max\_len = 512$. We set the initial learning rate as 3×10^{-5} and $batch\_size = 4$, and use the BERTAdam optimizer with a weight decay of 0.01. Our approach does not involve tuning the parameters on the validation set. Instead, we rely on the model checkpoints obtained after 5 epochs. Next, we introduce the comparison model and evaluation metrics in our experiments.

4.2.1. Model Under Comparison

We introduce two comstracting models approaches to multi-span answer extraction : **TASE** (Segal et al., 2020) and **LIQUID**(Lee et al., 2023). TASE utilizes a tag-based span extraction model which identifies multi-span answers though the assigning a tag to every input token with BIO tagging scheme. On the other hand, LIQUID serves as a framework for generating multi-span QA datasets to improve model performance.

To enhance the context with auxiliary information, we employ two distinct approaches: $\mathbf{AUG}_c$ and $\mathbf{AUG}_{eree}$, where **AUG** is our automatic data augmentation framework,and the suffix indicates which kind of information is injected into the context. $\mathbf{AUG}_C$ enriches the context with continue writing, while $\mathbf{AUG}_{EREE}$ supplements context with entities information including explanation and relationship analysis. Specifically, we leverage ChatGPT as a knowledge source to linearize the relevant information from large language models. in texts format and seamlessly integrate into the original contexts, thus reinforces the information of model inputs.

4.2.2. Evaluation Metrics

We use two automatic metrics for evaluation: Exact Match and Overlap F1 score.

- **Exact Match**. An exact match occurs when a predicted span fully matches one of the ground-truth answer spans. We calculate the micro-average precision, recall and f1 score for the extract match metric.

- **Overlap F1 score**. Overlap F1 score is the macro-average f1 score, where the f1 score for each example is computed by treating the prediction and gold as a bag of tokens.

4.3. Experimental Results and Analysis

In this section, we compare AUG with all competing models described above quantitatively.

4.3.1. Comparison Results

We evaluate our model as well as baselines (***Section*** 4.2.1) on the development splits of multi-span datasets (***Section*** 4.1) using automatic metrics (***Section*** 4.1). The comparison results are shown in Table 1, Table 2.

Table 1 and Table 2 illustrate the performance comparison between the proposed approaches, AUG_c and AUG_{eree}, and several strong baselines, including the previous state-of-the-art model LIQUID. These comparisons are conducted using both the BERT_{base} and RoBERTa_{base} encoders, and regard multi-span question answering as a *BIO* sequence tagging task to predict each token whether it is a part or begin of an answer. Notably,AUG_c exhibit superior performance across the evaluate dataset on all metrics. However, on Partial Match scores, AUG_{eree} demonstrates slightly lower performance compared to TASE, and especially lower than LIQUID when employing the BERT_{base} encoder. Importantly, the performance of AUG_c consistently outperforms LIQUID and TASE on all metrics and encoders, irrespective of the encoder setting. These results demonstrate the effectiveness of our proposed framework, as well as the efficacy of the information augmentation strategy.

To be more specific, Table 1 shows comparisons of metrics among all competing models 100 backed by BERT_{base}. We can see that our proposed framework, AUG, consistently outperforms all other baselines across multispanQA dataset. Backed by BERT_{base},

Table 1
Approach performance on complete MultiSpanQA valid set based on BERT_{base}.

Model	Exact Match			Partial Match		
	F(%)	P(%)	R(%)	F(%)	P(%)	R(%)
TASE	60.28	55.59	65.83	78.16	78.27	78.06
LIQUID	61.44	58.39	64.84	78.56	78.65	78.46
AUG_c	63.05	58.51	68.34	79.42	78.70	80.14
AUG_{eree}	63.93	60.22	68.13	77.50	77.07	77.94
AUG_{ereec}	62.90	61.02	64.89	76.72	78.56	74.97
Bagging_{eeerc}	63.63	61.53	65.88	78.24	80.00	76.56
$\text{Bagging}_{eeerc+liquid}$	64.44	61.63	67.50	79.5	80.49	78.57

Table 2
Approach performance on complete MultiSpanQA valid set based on RoBERTa_{base}.

Model	Exact Match			Partial Match		
	F(%)	P(%)	R(%)	F(%)	P(%)	R(%)
TASE	68.00	65.06	71.22	83.13	83.05	83.22
LIQUID	68.33	66.68	70.07	82.71	82.45	82.98
AUG_c	70.35	67.35	73.63	84.06	83.38	84.76
AUG_{eree}	69.15	67.81	70.54	82.85	83.90	81.83
AUG_{ereec}	70.44	67.83	73.26	82.80	82.45	83.15
Bagging_{eeerc}	70.48	69.11	71.90	84.28	85.61	83.00
$\text{Bagging}_{eeerc+liquid}$	70.86	69.03	72.79	84.82	85.53	84.12

AUG_c achieves EM and Overlap F1 scores of 63.05 and 79.42, respectively. Moreover, when equipped with entities' information, AUG_{eree} achieves even higher EM F1 scores of 63.93 but relatively lower Overlap F1 of 77.50 than baselines on the same encoder. These results showcase substantial improvements over the previous state-of-the-art model, LIQUID, with EM F1 score enhancements ranging from 1.61 to 2.49 percents across validation datasets.

Additionally, when utilizing RoBERTa_{base} instead of BERT_{base}, AUG_c achieves EM and Overlap F1 scores of 70.35 and 84.06 respectively on the same datasets. These scores represent EM and Overlap F1 improvements of 2.02 and 0.93 compared to the previous setup. For AUG_{eree}, the EM F1 score represents an enhancement of 0.82, with the EM and Overlap F1 values of 69.16 and 82.85 respectively. Notably, the partial metrics also indicate lower values compared to TASE and LIQUID, in line with the result supported by BERT_{base}. This is because augmenting the model with entity information, including definition and relationship knowledge, strengthens its ability to capture and understand entity concepts, which leads the model to prefer complete entity spans or empty span set as answers rather than partial entity span, and therefore a decrease in the partial recall and ultimately a lower partial F1 scores and a higher EM F1.

To further substantiate our explanation for the suboptimal performance of our model on the Overlap F1 metric, we conducted a detailed examination of the predictions made by AUG_{eree} and two baseline models on the validation dataset. In essence, we tallied the instances where these models predicted empty answers and recalculated their Overlap F1 scores on non-empty predictions. This allowed us to investigate whether the AUG_{eree} model aligns with our hypothesis, which posits that its extensive learning of entity knowledge during training makes it inclined to output either a complete and accurate answer span or no answer at all, as opposed to a partially correct answer span.

As presented in Table **??**, AUG_{eree} indeed predicted a higher number of empty answers compared to TASE and LIQUID, while achieving relatively higher Overlap F1 scores on non-empty predictions. Specifically, when equipped with BERT_{base} as the encoder, AUG_{eree} obtained an F1 score of 0.7976 on non-empty answer predictions, whereas TASE and LIQUID scored 0.7964 and 0.7909, respectively. With RoBERTa_{base}, AUG_{eree} achieved an Overlap F1 score of 0.8414, surpassing TASE and LIQUID, which scored 0.8404 and 0.8357, respectively. Additionally, it is worth noting that AUG_{eree} consistently predicted more empty answers, whether using BERT or RoBERTa.

These findings lend support to our conjecture that the introduction of entity knowledge leads to a slight reduction in the model's Overlap F1 scores. This suggests that utilizing LLM as a knowledge source to linearize entity information from LLM text and integrate it into the original context empowers the model to acquire greater entity knowledge, thereby

Table 3
The statistics of answers span predicted by AUG*eree*, AUG*ereec* and baselines

Category	BERT	RoBERTa
TASE		
empty preds	2.45	0.61
non-empty pmf1	79.64	84.04
LIQUID)		
empty preds	2.14	1.53
non-empty pmf1	79.09	83.57
AUG*eree*		
empty preds	**6.43**	**2.91**
non-empty pmf1	**79.76**	**84.14**
AUG*ereec*		
empty preds	**7.96**	**3.22**
non-empty pmf1	**80.20**	**83.78**

exhibiting a preference for more accurate and complete answer spans, or simply providing no answer.

Totally, the result, displayed in Table 2 demonstrates the same trends to Table 1. And the outcome highlights robustness in effectively generalizing across different datasets without requiring hyperparameter re-tuning.

4.3.2. Discussion

Table 1 and Table 2 discuss the performance of different augmentation integrated strategies, including the results-bagging methods whose outputs are voting results of AUG_c, AUG_{eree} as well as LIQUID, and the input-fusion model AUG_{ceree} who injects all kinds of information above into input contexts.

In detail, with EM f1 scores of 62.90 and 64.44 in Table 1 and 70.44 and 70.86 in Table 2, both the AUG_{ceree} and Bagging methods consistently surpass TASE and LIQUID, which exhibits robust effectiveness of information injection strategies. However, there is a little decrease caused by fusing all auxiliary information when contrast with single information augmentation strategies and the bagging method. This may be due to the likelihood that incorporating all of the augmentation information into the model inputs will confuse the model by introducing excessive auxiliary knowledge and underrepresented original context proportion. Therefore it may be more useful that adding limit information into context, and using result-bagging method, a multi-model voting to bringing all information into model with an indirect way.

From the Partial Match perspective, AUG_{ceree} and the Bagging method achieve 76.72 and 79.52 in Table 1, and 82.80 and 84.82 in Table 2. In accordance with Exact Match metrics, Bagging demonstrate a overall superior performance. Meanwhile overlap f1 score of AUG_{eree} is inferior to TASE but superior to LIQUID ,with relatively higher precision and relativelv higher recall, which is comparable to all single augmented models such as AUG_{eree}. And its weak performance on overlap f1 also reveals complete entities preference of this information injection approach.

Furthermore, we stratified the data within MultispanQA according to answer types, specifically categorizing them into DESC, NUM, and ENTYS. We subsequently conducted a comparative analysis of model performance within each of these subcategories. In particular, the results are presented in Tables 4 and Table 5, supported by BERT_{base} and RoBERTa_{base}, respectively. As indicated in Tables 4 and 5, our proposed models exhibit superior performance in terms of EM F1 scores for all categorizing. However, they demonstrate suboptimal performance in terms of overlap F1 scores.

4.3.3. Ablation Experiment

At the end of this section, we conducted ablations on our approach to confirm the effectiveness of selecting the information injection proportion. For each QA data, we randomly split the original context and the auxiliary text, then concatenated them into a final augmented context with a specific proportion to ensure that the new input length meets *max_len*, which has a crucial impact on our approach. In practice, we determined the final text splicing ratio by calculating the ratio of the average length of the source text to the added information, which for the AUG_c is 0.86.

Specifically, Table 6 and Table 7 displays AUG_c's performance with differential proportion to concatenate original contexts and continuation, on complete MultispanQA valid set,backed by BERT_{base} and RoBERTa_{base} respectively. We choose five proportions for information integration, which determine how much auxiliary information would be inject into each overflowed text segment. The results presented in tables indicate that using the ratio of their average lengths as the proportion of the overflow text composed of original text and auxiliary information is an effective approach. In detail, AUG_c,equipping with BERT_{base}, achieves an Exact Match F1 scores improvements of at least 4.57 compared to other proportions and an overlap F1 scores improvements of at least 2.07. In line with Table 6, when AUG_c equips with Roberta_{base}, it achieves an improvement of Exact Match F1 scores of 0.55 but an decrease of Overlap F1 scores of 0.17.

5. Conclusion

This section is not mandatory, but can be added to the manuscript if the discussion is unusually long or complex.

Acknowledgements

This research was supported by the Natural Science Foundation of Zhejiang Province, China (Grant No. LQ22F020027), Fundamental Research Funds of

Table 4
Model performance on complete MultiSpanQA valid Subset with different answer types based on $BERT_{base}$.

Type	Model	Exact Match			Partial Match		
		F(%)	P(%)	R(%)	F(%)	P(%)	R(%)
DESC	TASE	32.76	27.33	40.87	64.46	68.61	60.79
	LIQUID	36.69	31.10	44.71	66.05	65.95	66.15
	AUG_c	38.74	32.89	47.12	68.31	70.32	66.42
	AUG_{eree}	44.69	40.71	49.52	63.12	67.16	59.53
	AUG_{ereec}	40.63	38.30	43.27	64.25	71.15	58.57
	$Bagging_{eeerc}$	39.19	36.86	41.83	64.46	75.12	56.45
	$Bagging_{eeerc+liquid}$	39.91	36.69	43.75	66.70	73.68	60.93
NUM	TASE	33.33	30.23	37.14	60.98	71.99	52.89
	LIQUID	34.33	31.25	38.10	60.68	68.15	54.69
	AUG_c	35.96	33.33	39.05	64.47	71.56	58.65
	AUG_{eree}	36.20	34.48	38.10	59.02	65.63	53.62
	AUG_{ereec}	36.71	37.25	36.19	57.09	69.04	48.67
	$Bagging_{eeerc}$	35.58	35.92	35.24	58.62	68.09	51.46
	$Bagging_{eeerc+liquid}$	35.94	34.82	37.14	60.73	71.15	52.97
ENTYS	TASE	66.30	62.21	70.96	81.16	80.37	81.96
	LIQUID	67.17	65.25	69.21	81.66	81.69	81.63
	AUG_c	68.47	64.44	73.03	81.93	80.57	83.34
	AUG_{eree}	68.36	64.64	72.53	80.55	79.21	81.93
	AUG_{ereec}	67.54	65.60	69.59	79.49	80.16	78.83
	$Bagging_{eeerc}$	68.68	66.49	71.03	81.11	81.39	80.83
	$Bagging_{eeerc+liquid}$	69.65	66.94	72.59	82.31	82.07	82.55

Zhejiang Sci-Tech University (Grant No. 23232138-Y), Liaoning Provincial Natural Science Foundation of China (Grant No. 2022-KF-21-01) and the Key Research and Development Program of Zhejiang Province, China (Grant No. 2023C01041 and 2022C01079).

References

Dai, H., Liu, Z., Liao, W., Huang, X., Cao, Y., Wu, Z., Zhao, L., Xu, S., Liu, W., Liu, N., Li, S., Zhu, D., Cai, H., Sun, L., Li, Q., Shen, D., Liu, T., Li, X., 2023. Auggpt: Leveraging chatgpt for text data augmentation. arXiv:2302.13007.

Devlin, J., Chang, M., Lee, K., Toutanova, K., 2019. BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pp. 4171–4186.

Fang, Y., Wang, S., Xu, Y., Xu, R., Sun, S., Zhu, C., Zeng, M., 2021. Leveraging knowledge in multilingual commonsense reasoning. arXiv preprint arXiv:2110.08462 .

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., Poon, H., 2021. Domain-specific language model pretraining for biomedical natural language processing. ACM Transactions on Computing for Healthcare (HEALTH) 3, 1–23.

Hu, Z., Yang, P., Li, B., Sun, Y., Yang, B., 2023. Biomedical extractive question answering based on dynamic routing and answer voting. Information Processing & Management 60, 103367.

Huang, Z., Zhou, J., Niu, C., Cheng, G., 2023a. Spans, not tokens: A span-centric model for multi-span reading comprehension, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, Association for Computing Machinery, New York, NY, USA. p. 874–884. URL: https://doi.org/10.1145/3583780.3615064, doi:10.1145/3583780.3615064.

Huang, Z., Zhou, J., Xiao, G., Cheng, G., 2023b. Enhancing in-context learning with answer feedback for multi-span question answering. arXiv preprint arXiv:2306.04508 .

Kadlec, R., Schmid, M., Bajgar, O., Kleindienst, J., 2016. Text understanding with the attention sum reader network, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers.

Kenton, J.D.M.W.C., Toutanova, L.K., 2019. Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL-HLT, pp. 4171–4186.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J., 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 36, 1234–1240.

Lee, S., Kim, H., Kang, J., 2023. Liquid: A framework for list question answering dataset generation. arXiv preprint arXiv:2302.01691 .

Li, H., Tomko, M., Vasardani, M., Baldwin, T., 2022. Multispanqa: A dataset for multi-span question answering, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1250–1260.

Liu, J., Hallinan, S., Lu, X., He, P., Welleck, S., Hajishirzi, H., Choi, Y., 2022. Rainier: Reinforced knowledge introspector for commonsense question answering. arXiv preprint arXiv:2210.03078 .

Liu, J., Liu, A., Lu, X., Welleck, S., West, P., Bras, R.L., Choi, Y., Hajishirzi, H., 2021. Generated knowledge prompting for commonsense reasoning. arXiv preprint arXiv:2110.08387 .

Table 5
Model performance on complete MultiSpanQA valid Subset with different answer types based on RoBERTa$_{base}$.

Type	Model	Exact Match			Partial Match		
		F(%)	P(%)	R(%)	F(%)	P(%)	R(%)
DESC	TASE	45.53	40.84	51.44	75.44	76.61	74.31
	LIQUID	48.68	44.76	53.37	73.27	71.57	75.04
	AUG$_c$	49.02	44.66	54.33	76.33	77.85	74.86
	AUG$_{eree}$	50.99	46.96	55.77	76.87	78.07	75.71
	AUG$_{ereec}$	49.24	44.71	54.81	71.87	73.03	70.74
	Bagging$_{eeerc}$	51.54	47.56	56.25	78.20	80.04	76.44
	Bagging$_{eeerc+liquid}$	47.70	43.78	52.40	77.43	80.63	74.47
NUM	TASE	46.23	45.79	46.67	71.89	78.42	66.36
	LIQUID	40.19	39.45	40.95	67.56	72.28	63.42
	AUG$_c$	50.69	49.11	52.38	72.86	80.22	66.74
	AUG$_{eree}$	42.40	41.07	43.81	66.67	73.51	61.00
	AUG$_{ereec}$	45.58	44.55	46.67	65.86	73.19	59.87
	Bagging$_{eeerc}$	43.32	41.96	44.76	68.57	76.54	62.11
	Bagging$_{eeerc+liquid}$	44.65	43.64	45.71	70.11	78.49	63.35
ENTYS	TASE	72.57	69.94	75.41	84.90	84.32	85.48
	LIQUID	72.95	71.77	74.16	85.02	84.75	85.29
	AUG$_c$	74.59	71.87	77.53	85.79	84.40	87.23
	AUG$_{eree}$	73.50	72.81	74.22	84.74	85.49	83.99
	AUG$_{ereec}$	75.04	72.81	77.41	85.37	84.46	86.29
	Bagging$_{eeerc}$	74.97	74.23	75.72	86.14	87.07	87.06
	Bagging$_{eeerc+liquid}$	75.85	74.52	77.22	86.73	86.73	86.74

Table 6
Ablations of AUG$_c$ on different proportion for information concatenation, based on BERT$_{base}$.

Proportion	Exact Match			Partial Match		
	F(%)	P(%)	R(%)	F(%)	P(%)	R(%)
0.90	58.55	57.69	59.44	71.88	74.30	69.61
0.70	61.70	59.03	64.63	76.58	77.84	75.36
0.50	61.22	57.37	65.62	77.38	77.61	77.16
0.40	61.36	56.50	67.13	78.26	77.57	78.96
0.10	61.41	56.36	67.45	78.94	78.45	79.44
0.14	63.05	58.51	68.34	79.42	78.70	80.14

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 .

Naseem, U., Dunn, A.G., Khushi, M., Kim, J., 2022. Benchmarking for biomedical natural language processing tasks with a domain specific ALBERT. BMC Bioinform. 23, 144.

Pang, L., Lan, Y., Guo, J., Xu, J., Su, L., Cheng, X., 2019. HAS-QA: hierarchical answer spans model for open-domain

Table 7
Ablations of AUG$_c$ on different proportion for information concatenation, based on RoBERTa$_{base}$.

Proportion	Exact Match			Partial Match		
	F(%)	P(%)	R(%)	F(%)	P(%)	R(%)
0.90	58.13	56.92	59.39	71.70	73.49	69.98
0.70	66.97	65.42	68.60	80.05	80.80	79.32
0.50	68.94	67.23	70.75	82.43	83.23	81.64
0.40	68.95	66.11	71.06	83.71	83.71	81.71
0.10	69.80	66.47	73.47	84.23	83.49	84.99
0.86	70.35	67.35	73.63	84.06	83.38	84.76

question answering, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, AAAI Press. pp. 6875–6882.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al., 2018. Improving language understanding by generative pre-training .

Ren, R., Wang, Y., Qu, Y., Zhao, W.X., Liu, J., Tian, H., Wu, H., Wen, J.R., Wang, H., 2023. Investigating the factual knowledge boundary of large language models with retrieval augmentation. arXiv preprint arXiv:2307.11019 .

Segal, E., Efrat, A., Shoham, M., Globerson, A., Berant, J., 2020. A simple and effective model for answering multi-span questions, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3074–3080.

Seo, M.J., Kembhavi, A., Farhadi, A., Hajishirzi, H., 2017. Bidirectional attention flow for machine comprehension, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.

Shao, Z., Gong, Y., Shen, Y., Huang, M., Duan, N., Chen, W., 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. arXiv preprint arXiv:2305.15294 .

Tan, Y., Min, D., Li, Y., Li, W., Hu, N., Chen, Y., Qi, G., 2023. Can chatgpt replace traditional kbqa models? an in-depth analysis of the question answering performance of the gpt llm family, in: International Semantic Web Conference, Springer. pp. 348–367.

Trischler, A., Ye, Z., Yuan, X., Bachman, P., Sordoni, A., Suleman, K., 2016. Natural language comprehension with the epireader, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, pp. 128–137.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems 30.

Vinyals, O., Fortunato, M., Jaitly, N., 2015. Pointer networks, in: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pp. 2692–2700.

Wang, S., Jiang, J., 2017. Machine comprehension using match-lstm and answer pointer, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net.

Wei, X., Cui, X., Cheng, N., Wang, X., Zhang, X., Huang, S., Xie, P., Xu, J., Chen, Y., Zhang, M., et al., 2023. Zero-shot information extraction via chatting with chatgpt. arXiv preprint arXiv:2302.10205 .

Xiong, C., Zhong, V., Socher, R., 2017. Dynamic coattention networks for question answering, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net.

Yoo, K.M., Park, D., Kang, J., Lee, S.W., Park, W., 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. arXiv preprint arXiv:2104.08826 .

Yoon, W., Jackson, R., Lagerberg, A., Kang, J., 2022. Sequence tagging for biomedical extractive question answering. Bioinform. 38, 3794–3801.

Yu, W., Iter, D., Wang, S., Xu, Y., Ju, M., Sanyal, S., Zhu, C., Zeng, M., Jiang, M., 2022. Generate rather than retrieve: Large language models are strong context generators. arXiv preprint arXiv:2209.10063 .

Zhang, C., Lin, J., Liu, X., Lai, Y., Feng, Y., Zhao, D., 2023. How many answers should i give? an empirical study of multi-answer reading comprehension. arXiv preprint arXiv:2306.00435 .

Zhu, M., Ahuja, A., Juan, D., Wei, W., Reddy, C.K., 2020. Question answering with long multiple-span answers, in: Cohn, T., He, Y., Liu, Y. (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, pp. 3840–3849.