QASE Enhanced PLMs: Improved Control in Text Generation for MRC

Anonymous ACL submission

Abstract

To address the challenges of out-of-control generation in generative models for machine reading comprehension (MRC), we introduce the **Q**uestion-**A**ttended **S**pan **E**xtraction (*QASE*) module. Integrated during the fine-tuning of pre-trained generative language models (PLMs), *QASE* enables these PLMs to match SOTA extractive methods and outperform leading LLMs like GPT-4 in MRC tasks, without significant increases in computational costs. [1]

1 Introduction

Machine Reading Comprehension (MRC) is a critical NLP challenge. Recent developments include well-annotated benchmark datasets like such as (Rajpurkar et al., 2016), Quoref (Dasigi et al., 2019), and MultiSpanQA (Li et al., 2022a). Mainstream approaches to MRC extract a relevant piece of text from the context in response to a question (Wang et al., 2018; Yan et al., 2019; Chen et al., 2020), but in real-world application, the correct answers often span multiple passages or are implicit (Li et al., 2021). Exploring generative models, in addition to extractive methods, is essential.

Generative models, however, underperform in MRC due to out-of-control generation (Li et al., 2021). This leads to two main challenges: (1) ill-formed generated answers, containing incomplete or redundant phrases, and (2) factual inconsistency in the generated answers deviating from the correct response. In this paper, we address these by introducing a lightweight **Q**uestion-**A**ttended **S**pan **E**xtraction (*QASE*) module. We fine-tune multiple open-source generative pre-trained language models (PLMs) on various MRC datasets to assess the module's efficacy in guiding answer generation. Our contributions include: **(1)** Developing *QASE* to improve fine-tuned generative PLMs' quality and factual consistency on MRC tasks, matching SOTA extractive methods and surpassing GPT-4; **(2)** *QASE* boosts performance without significantly increasing computational costs, benefiting researchers with limited resources.

[1] Our code is available at this anonymous repo link.

2 Related Work

Most **current studies on MRC** involve predicting the start and end positions of the answer spans from a given context (Ohsugi et al., 2019; Lan et al., 2019; Bachina et al., 2021). To handle the multi-span setting, some studies frame the problem as a sequence tagging task (Segal et al., 2020), and others explore ways to combine models with different tasks (Hu et al., 2019; Lee et al., 2023; Zhang et al., 2023). While these extractive-based methods mainly utilize encoder-only models, such as BERT and RoBERTa, there is also research focuses on using the power of generative-based language models (Yang et al., 2020; Li et al., 2021; Su et al., 2022).

Retrieval-augmented text generation (RAG) augments the input of PLMs with in-domain (Gu et al., 2018; Weston et al., 2018; Saha and Srihari, 2023) or external knowledge (Su et al., 2021; Xiao et al., 2021) to control the quality and factual consistency of generated content. It has become a new text generation paradigm in many NLP tasks (Li et al., 2022b), such as dialogue response generation (Wu et al., 2021; Liu et al., 2023b) and machine translation (He et al., 2021; Zhu et al., 2023). However, not much work focuses on selective MRC. Our approach diverges from RAG as it directly fine-tunes the weights of the PLMs rather than altering the input to the PLMs with additional information.

3 Method

Question-Attended Span Extraction To guide text generation, we use *QASE*, a question-attended span extraction module, during fine-tuning the generative PLMs. *QASE* focuses model attention on potential answer spans within the original context.

We cast span extraction as a sequence tagging problem and employ the Inside-Outside (IO) tagging schema, where each sequence token is tagged as 'inside' (***I***) if part of a relevant span, or 'outside' (***O***) if not. This schema works well for both single- and multi-span extraction settings, achieving comparable or even better performance than the well-known BIO tagging format (Huang et al., 2015), as shown by Segal et al. (2020).

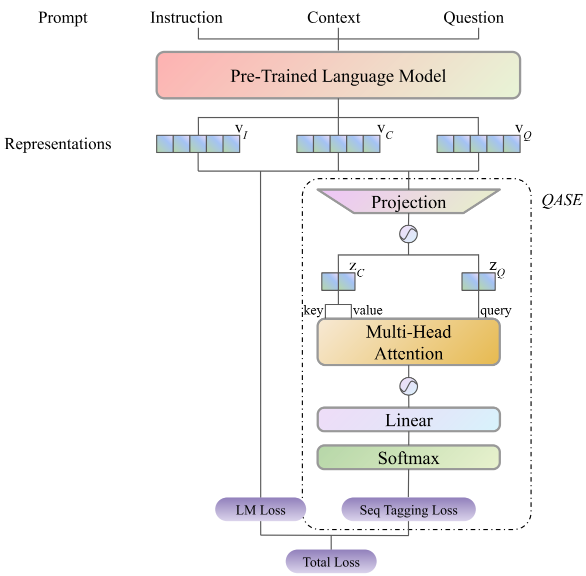

Figure 1: *QASE*-enhanced model architecture

The architecture of our model is shown in Figure 1. An input context and question pair and an instruction are first tokenized and fed into the PLM. The hidden states output from the PLM is then passed through projection layers to produce embeddings $z_i = ReLU(W_{proj}v_i + b_{proj})$, where $v_i \in R^d$ is the PLM output hidden state of the i^{th} token.

To learn context tokens representations in relation to specific questions, we employ a **multi-head attention** mechanism (*MHA*). Each head in *MHA* focuses to different aspects of the context as it relates to the question, using question embeddings as the query and context embeddings as key-value pairs. This mechanism aligns the context token representations with the specifics of the queried question. The projected embeddings z_i are passed through *MHA*, and subsequently channeled through a linear layer and a softmax layer to compute $p_i = softmax(W_{lin} \cdot MHA(z_i) + b_{lin})$, which denotes the probability of the i^{th} token being inside the answer spans. We then compute the sequence tagging loss using the cross entropy loss $L_{QASE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=0}^{1} y_{ij} log(p_{ij})$, where $j \in 0, 1$ corresponds to class ***O*** and class ***I***, and y_{ij} is a binary value indicating whether the i^{th} token belongs to class j.

Fine-Tuning and Inference We fine-tune the PLMs using multi-task learning, simultaneously optimizing both the language modeling loss and sequence tagging loss: $L = L_{LML} + \beta L_{QASE}$, where β is a hyper-parameter that controls the weight of the span extraction task. This approach enhances the PLMs' ability to generate answers well-founded in the context and relevant answer spans. During inference, only the generation component of the fine-tuned model is employed.

4 Experiments

Datasets and Metrics We utilize these 3 MRC datasets. **(1) SQuAD** (Rajpurkar et al., 2016): A benchmark reading comprehension dataset consisting of 100K+ questions with single-span answers. We use SQuAD v1.1. Since the official evaluation on v1.1 has long been ended, we report our results on the official v1.1 development set. **(2) MultiSpanQA** (Li et al., 2022a): This reading comprehension dataset consists of over 6.5k question-answer pairs. Unlike most existing single-span answer MRC datasets, MultiSpanQA focuses on multi-span answers. **(3) Quoref** (Dasigi et al., 2019): A benchmark reading comprehension dataset containing more than 24K questions, with most answers being single-span and ∼10% being multi-span. Following the conventions of the datasets' official leaderboards (information listed in Appendix A.1), we employ exact match (EM) and partial match (Overlap) F1 scores as metrics on MultiSpanQA, and exact match percentage and macro-averaged F1 score on SQuAD and Quoref.

Experimental Setup To evaluate the effectiveness of *QASE* independent of any specific language model, we experiment with multiple open-source LLMs. These include both decoder-only LLMs, such as Llama 2 (Touvron et al., 2023) and Alpaca (Taori et al., 2023), and an encoder-decoder model, Flan-T5 (Chung et al., 2022). For Llama 2 and Alpaca, we fine-tune the pre-trained 7B version using LoRA (Hu et al., 2021) and instruction-tuning (see Appendix A.3 for instruction templates). For Flan-T5 family models, we fine-tune the small, the base, and the large versions. The trainable parameters for each model is provided in Table 2.

We train all our models on single GPUs, using a batch size of 2-4 depending on the VRAM of the respective GPUs. We use four types of GPUs: A40,

		Llama2	Alpaca	Flan-T5-Small	Flan-T5-Base	Flan-T5-Large
SQuAD	no *QASE*	36.68 | 47.06	27.88 | 43.95	77.33 | 85.51	82.09 | 89.56	83.16 | 90.71
(EM | F1)	*QASE*	**37.22 | 47.69**	**37.31 | 47.62**	**77.66 | 85.90**	**82.20 | 90.24**	**84.13 | 91.70**
MultiSpanQA	no *QASE*	50.93 | 68.14	**52.73** | 69.10	**59.13** | 76.49	64.66 | 81.41	**67.41** | 83.09
(EM F1 | Overlap F1)	*QASE*	**51.75 | 70.39**	52.20 | **70.01**	59.08 | **77.10**	**64.87 | 81.50**	66.92 | **84.22**
Quoref	no *QASE*	45.52 | 52.09	-	58.21 | 63.30	72.77 | 80.90	75.17 | 80.49
(EM | F1)	*QASE*	**54.28 | 60.44**	-	**60.70 | 66.88**	**75.17 | 81.18**	**76.19 | 82.13**

Table 1: Performance of fine-tuned PLMs with or without *QASE* on each dataset.

	Trainable Parameters		
	no *QASE*	*QASE*	Δparams
Llama2/Alpaca with LoRA	4.2M	7.3M	3.1M
Flan-T5-Small	77.0M	78.2M	1.3M
Flan-T5-Base	247.6M	248.9M	1.4M
Flan-T5-Large	783.2M	784.7M	1.5M

Table 2: Trainable parameters of experimented models.

A10, A5500, and A100. Models are trained for 3 epochs or until convergence. Notably, model variants derived from the same base PLM share identical configurations including learning rate, weight decay, batch size, epoch number, and GPU type.

Experiment Results To evaluate the efficacy of the *QASE*, we examine the performance of various PLMs fine-tuned with and without *QASE*, as shown in Table 1. Generally, models fine-tuned with *QASE* outperform those fine-tuned without it. In particular, for SQuAD, *QASE*-enhanced model demonstrate an EM percentage increase of up to 33.8% and an F1 score upsurge of up to 8.4% over vanilla fine-tuned models. For MultiSpanQA, there is an improvement of up to 1.6% in the EM F1 and up to 3.3% in the overlap F1. Likewise, on Quoref, there is an improvement of up to 19.2% in the EM percentage and up to 16.0% in the F1 score. These results show that, by employing *QASE*, generative-based PLMs can be fine-tuned to produce well-formed, context-grounded, and better-quality answers in MRC tasks compared to the vanilla fine-tuning approach. For reference, we also compare the fine-tuned PLMs to their corresponding PLMs in zero-shot settings, as presented in Appendix A.2.

Computational Costs Table 2 shows that integrating *QASE* slightly raises the number of trainable parameters in PLMs, with the increase dependent on the models' hidden sizes. Significantly, for the largest model, Flan-T5-Large, *QASE* adds just 0.2% more parameters, indicating that *QASE* enhances the capabilities of fine-tuned PLMs in MRC without major increase in computational resources.

Model Comparisons Our top model, Flan-T5-Large$_{QASE}$, is further benchmarked against leading models on each dataset's official leaderboard, alongside zero-shot GPT-3.5-Turbo and GPT-4. GPT-3.5-Turbo stands as one of OpenAI's most efficient models in terms of capability and cost, while GPT-4 shows superior reasoning abilities (Liu et al., 2023c). Studies indicate their superiority over traditional fine-tuning methods in most logical reasoning benchmarks (Liu et al., 2023a). The prompts used to query the GPT variants are detailed in Appendix A.3.

On SQuAD, as illustrated in Table 3, Flan-T5-Large$_{QASE}$ surpasses human performance, equaling the NLNet model from Microsoft Research Asia and the original pre-trained BERT-Large from Google (Devlin et al., 2019), which are ranked #11 and #13 on the v1.1 leaderboard respectively. Additionally, it surpasses GPT-4 by 113.8% on the exact match score and 32.6% on F1.

	EM	F1 ↑
GPT-3.5-Turbo	36.944	65.637
GPT-4	39.347	69.158
Human Performance	82.304	91.221
BERT-Large (Devlin et al., 2019)	84.328	91.281
MSRA NLNet (ensemble)	**85.954**	91.677
Flan-T5-Large$_{QASE}$	84.125	**91.701**

Table 3: Results of Flan-T5-Large$_{QASE}$ and baselines on **SQuAD**.

On MultiSpanQA, Table 4 shows that Flan-T5-Large$_{QASE}$ outperforms LIQUID (Lee et al., 2023), which currently ranks #1 on the leaderboard, with respect to the overlap F1 score. Moreover, it surpasses GPT-4 by 4.5% on the exact match F1 and 1.5% on the overlap F1.

	EM F1	Overlap F1 ↑
GPT-3.5-Turbo	59.766	81.866
GPT-4	64.027	82.731
LIQUID (Lee et al., 2023)	**73.130**	83.360
Flan-T5-Large$_{QASE}$	66.918	**84.221**

Table 4: Performance of Flan-T5-Large$_{QASE}$ and baselines on **MultiSpanQA**.

On Quoref, Table 5 shows that Flan-T5-Large$_{QASE}$ is comparable to CorefRoberta-Large

(Ye et al., 2020), which ranks #9 on the leaderboard, with a 0.5% higher exact match. Furthermore, it outperforms GPT-4 by 11.9% on the exact match and 4.8% on F1.

	EM	**F1** ↑
GPT-3.5-Turbo	50.22	59.51
GPT-4	68.07	78.34
CorefRoberta-Large (Ye et al., 2020)	75.80	**82.81**
Flan-T5-Large$_{QASE}$	**76.19**	82.13

Table 5: Performance of Flan-T5-Large$_{QASE}$ and baselines on **Quoref**.

All top-performing models on these datasets' leaderboards, equaling or exceeding Flan-T5-Large$_{QASE}$, are encoder-only extractive models. Therefore, these results demonstrate that *QASE*-enhanced generative PLMs can be fine-tuned to match or exceed the capabilities of SOTA extractive models and outperform leading LLMs in MRC.

Ablation Studies To demonstrate the superiority of the *QASE* architecture, we compared Flan-T5-Large$_{QASE}$ with vanilla fine-tuned Flan-TF-Larg$_{FT}$ and Flan-T5-Large$_{baseline}$. The baseline span extraction module lacks the *MHA* component, making it a standard architecture for fine-tuning pre-trained encoders for downstream sequence tagging tasks. We also explored both question-first (*qf*) and context-first prompting strategies, with further details and analysis provided in Appendix A.4, where the model architecture is also illustrated.

Table 6 shows that the baseline-embedded model performs better with a question-first prompting strategy, as Flan-T5-Large$_{baseline_{qf}}$ surpasses Flan-T5-Large$_{baseline}$ and Flan-T5-Large$_{FT_{qf}}$. Conversely, the baseline span extraction module decreases performance in context-first prompting, where Flan-T5-Large$_{baseline}$ underperforms compared to Flan-T5-Large$_{FT}$. This suggests that adding an auxiliary span extraction module without careful design can negatively affect instruction fine-tuning. Meanwhile, the *QASE*-enhanced model excels over both vanilla fine-tuned and baseline-embedded models in both prompting scenarios, demonstrating its architectural superiority. Specifically, in context-first setting, Flan-T5-Large$_{QASE}$ significantly outperforms Flan-T5-Large$_{baseline}$ with a 4.3% higher F1.

	EM	**F1** ↑
Flan-T5-Large$_{baseline}$	79.877	87.918
Flan-T5-Large$_{FT_{qf}}$	80.378	88.176
Flan-T5-Large$_{baseline_{qf}}$	81.125	89.043
Flan-T5-Large$_{QASE_{qf}}$	81.485	89.077
Flan-T5-Large$_{FT}$	83.159	90.712
Flan-T5-Large$_{QASE}$	**84.125**	**91.701**

Table 6: Performance of vanilla, baseline-, and *QASE*-enhanced fine-tuned Flan-T5-Large on **SQuAD**.

Factual Consistency While token-based EM and F1 scores measure the structural quality of generated text, they do not reflect factual accuracy relative to the context. For this we used Q^2 (Honovich et al., 2021), an automatic metric for assessing factual consistency in generated text, which uses question generation and answering methods over token-based matching. We compared fine-tuned Flan-T5-Large with and without *QASE* in both single-span (SQuAD) and multi-span (MultiSpanQA) answer settings. Table 7 shows that *QASE*-enhanced models consistently outperform the vanilla fine-tuned model. On SQuAD, Q^2 NLI score is improved by 1.0%, and on MultiSpanQA, it is improved by 16.0%.

	Flan-T5-Large	Q^2 **F1**	Q^2 **NLI**
SQuAD	no *QASE*	42.927	44.983
	QASE	**43.624**	**45.419**
MultiSpanQA	no *QASE*	32.889	31.433
	QASE	**34.732**	**36.452**

Table 7: Q^2 scores of fine-tuned Flan-T5-Large with or without $QASE$ on each dataset.

5 Conclusion and Future Work

In this study, we address out-of-control text generation of generative PLMs in MRC using *QASE*, a lightweight question-attended span extraction module, during the fine-tuning of PLMs. Our experiments show that *QASE*-enhanced PLMs generate better-quality responses with improved formality and factual consistency, matching SOTA extractive models and outperforming GPT-4 by a significant margin on all three MRC datasets. Importantly, *QASE* improves performance without a significant increase in computational costs, benefiting researchers with limited resources.

In the future, we plan to test our model on generative MRC datasets (Nguyen et al., 2016) to further assess its efficacy in more complex scenarios. Another key focus will be evaluating the model's general ability in answer generation, particularly from the perspective of human perception. This will involve incorporating human annotators in addition to automatic metrics. For a long-term goal, we are looking to expand our work to explore solutions for addressing input- and context-conflicting hallucinations in LLMs.

Limitations

Due to our limited computational resources, we have been able to perform our experiments on models no larger than Flan-T5-Large. This same constraint led us to only fine-tuning of Llama 2 and Alpaca with LoRA. We note that models based on Llama 2 and Alpaca generally underperform those based on Flan-T5. Apart from the inherent distinctions between decoder-only and encoder-decoder models, and their suitability for different tasks (as seen from the models' zero-shot performance), a possible factor could be the number of trainable parameters during fine-tuning. Specifically, fine-tuning Llama 2 and Alpaca with LoRA results in only 4.2M trainable parameters, while even the smallest Flan-T5 model provides 77.0M trainable parameters, as shown in Table 2. We acknowledge that many researchers face similar computational resource limitations. Therefore, our research should be very useful, proposing this lightweight module capable of enhancing smaller PLMs to outperform leading LLMs on MRC tasks like these, achieving a balance of effectiveness and affordability.

One foreseeable limitation of our work is the dependency of the fine-tuning process on answer span annotations, since *QASE* works as an auxiliary supervised span extraction module. This reliance on annotated data could potentially limit the model's broader applicability. A prospective exciting future direction to address this limitation is to develop a semi- or unsupervised module that focuses on selecting relevant spans or rationales within a given context. By integrating this module with our current model, we could significantly improve its generalization capabilities, thereby making it more adaptable and effective across a wider range of scenarios.

One popular method to enhance the formality of answers generated by LLMs is through prompt engineering, paired with few-shot or in-context learning techniques. While these strategies offer great advantages, our ultimate goal is to create a system with broad domain generalization, one that minimizes the need for extensive, calibrated prompt engineering and sample selections for task adaptation. Although developing a robust prompt engineering framework or paradigm is an appealing direction, our current focus diverges from this path. As a long-term goal, we aim for a solution that handles diverse tasks with minimal task-specific tuning.

References

Sony Bachina, Spandana Balumuri, and Sowmya Kamath S. 2021. Ensemble ALBERT and RoBERTa for span prediction in question answering. In *Proceedings of the 1st Workshop on Document-grounded Dialogue and Conversational Question Answering (DialDoc 2021)*, pages 63–68, Online. Association for Computational Linguistics.

Kunlong Chen, Weidi Xu, Xingyi Cheng, Zou Xiaochuan, Yuyu Zhang, Le Song, Taifeng Wang, Yuan Qi, and Wei Chu. 2020. Question directed graph attention network for numerical reasoning over text. *arXiv preprint arXiv:2009.07448*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5925–5932, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2018. Search engine guided neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Qiuxiang He, Guoping Huang, Qu Cui, Li Li, and Lemao Liu. 2021. Fast and accurate neural machine translation with translation memory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3170–3180, Online. Association for Computational Linguistics.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. Q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. *arXiv preprint arXiv:2104.08202*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Minghao Hu, Yuxing Peng, Zhen Huang, and Dongsheng Li. 2019. A multi-type multi-span network for reading comprehension that requires discrete reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1596–1606, Hong Kong, China. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. arxiv 2015. *arXiv preprint arXiv:1508.01991*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Seongyun Lee, Hyunjae Kim, and Jaewoo Kang. 2023. Liquid: A framework for list question answering dataset generation. *arXiv preprint arXiv:2302.01691*.

Chenliang Li, Bin Bi, Ming Yan, Wei Wang, and Songfang Huang. 2021. Addressing semantic drift in generative question answering with auxiliary extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 942–947, Online. Association for Computational Linguistics.

Haonan Li, Martin Tomko, Maria Vasardani, and Timothy Baldwin. 2022a. 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*, pages 1250–1260, Seattle, United States. Association for Computational Linguistics.

Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022b. A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110*.

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023a. Evaluating the logical reasoning ability of chatgpt and gpt-4. *arXiv preprint arXiv:2304.03439*.

Shuai Liu, Hyundong Cho, Marjorie Freedman, Xuezhe Ma, and Jonathan May. 2023b. RECAP: Retrieval-enhanced context-aware prefix encoder for personalized dialogue response generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8404–8419, Toronto, Canada. Association for Computational Linguistics.

Xiao Liu, Junfeng Yu, Yibo He, Lujun Zhang, Kaiyichen Wei, Hongbo Sun, and Gang Tu. 2023c. System report for CCL23-eval task 9: HUST1037 explore proper prompt strategy for LLM in MRC task. In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 3: Evaluations)*, pages 310–319, Harbin, China. Chinese Information Processing Society of China.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660.

Yasuhito Ohsugi, Itsumi Saito, Kyosuke Nishida, Hisako Asano, and Junji Tomita. 2019. A simple but effective method to incorporate multi-turn context with BERT for conversational machine comprehension. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 11–17, Florence, Italy. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sougata Saha and Rohini Srihari. 2023. ArgU: A controllable factual argument generator. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8373–8388, Toronto, Canada. Association for Computational Linguistics.

Elad Segal, Avia Efrat, Mor Shoham, Amir Globerson, and Jonathan Berant. 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)*, pages 3074–3080, Online. Association for Computational Linguistics.

Dan Su, Xiaoguang Li, Jindi Zhang, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. 2022. Read before generate! faithful long form question answering with machine reading. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 744–756, Dublin, Ireland. Association for Computational Linguistics.

Yixuan Su, Yan Wang, Deng Cai, Simon Baker, Anna Korhonen, and Nigel Collier. 2021. Prototype-to-style: Dialogue generation with style-aware editing on retrieval memory. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2152–2161.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wei Wang, Ming Yan, and Chen Wu. 2018. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1705–1714, Melbourne, Australia. Association for Computational Linguistics.

Jason Weston, Emily Dinan, and Alexander Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92, Brussels, Belgium. Association for Computational Linguistics.

Zeqiu Wu, Michel Galley, Chris Brockett, Yizhe Zhang, Xiang Gao, Chris Quirk, Rik Koncel-Kedziorski, Jianfeng Gao, Hannaneh Hajishirzi, Mari Ostendorf, et al. 2021. A controllable model of grounded response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14085–14093.

Fei Xiao, Liang Pang, Yanyan Lan, Yan Wang, Huawei Shen, and Xueqi Cheng. 2021. Transductive learning for unsupervised text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2510–2521, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ming Yan, Jiangnan Xia, Chen Wu, Bin Bi, Zhongzhou Zhao, Ji Zhang, Luo Si, Rui Wang, Wei Wang, and Haiqing Chen. 2019. A deep cascade model for multi-document reading comprehension. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7354–7361.

Junjie Yang, Zhuosheng Zhang, and Hai Zhao. 2020. Multi-span style extraction for generative reading comprehension. *arXiv preprint arXiv:2009.07382*.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7170–7186, Online. Association for Computational Linguistics.

Chen Zhang, Jiuheng Lin, Xiao Liu, Yuxuan Lai, Yansong Feng, and Dongyan Zhao. 2023. How many answers should i give? an empirical study of multi-answer reading comprehension. *arXiv preprint arXiv:2306.00435*.

Wenhao Zhu, Jingjing Xu, Shujian Huang, Lingpeng Kong, and Jiajun Chen. 2023. INK: Injecting kNN knowledge in nearest neighbor machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15948–15959, Toronto, Canada. Association for Computational Linguistics.

A Appendix

A.1 Dataset Leaderboard

Below are the official leaderboards all the datasets we refer to:

SQuAD	https://rajpurkar.github.io/SQuAD-explorer/
MultiSpanQA	https://multi-span.github.io/
Quoref	https://leaderboard.allenai.org/quoref/submissions/public

Table 8: Dataset official leaderboards.

A.2 Full Experiment Results

In addition to the highlighted results presented in Section 4, we also compare the fine-tuned PLMs to their corresponding base PLMs in zero-shot settings. The results, presented in Table 9, show that fine-tuning with *QASE* improves performance across all datasets. Specifically, on the SQuAD dataset, models using *QASE* perform up to 5.6 times better in exact match and 3.0 times better in F1 score compared to the original models. On the MultiSpanQA dataset, the exact match improves by up to 124.4 times, and F1 score by up to 3.4 times. Similarly, on the Quoref dataset, the exact match improves by up to 38.4 times, and F1 score by up to 11.2 times with *QASE*.

A.3 Instruction Templates and Model Prompts

Table 10 provides the instruction and prompt templates used for fine-tuning the PLMs and for zero-shot querying of PLMs and GPT variants across both single- and multi-span answer datasets.

A.4 Ablation Studies Details

Figure 2 depicts the architecture of the model we use for the ablation studies, with a baseline span extraction module. The baseline span extraction module omits the *MHA* component, typifying a standard architecture for fine-tuning pre-trained encoders for downstream sequence tagging tasks. The baseline-embedded Flan-T5-Large models are fine-tuned with the same configurations as Flan-T5-Large$_{QASE}$ including learning rate, weight decay, batch size, epoch number, and GPU type.

We experiment with 2 prompting strategies for ablation studies:

- **Context-first prompting:** The default prompting strategy we utilize for fine-tuning PLMs, both with and without *QASE*. In this setting, the prompt is ordered as "<instruction tokens> <context tokens> <question tokens>".
- **Question-first prompting (*qf*):** Following BERT's standard fine-tuning procedures. In this setting, the prompt is ordered as "<instruction tokens> <question tokens> <SEP> <context tokens>". <SEP> is a special separator token.

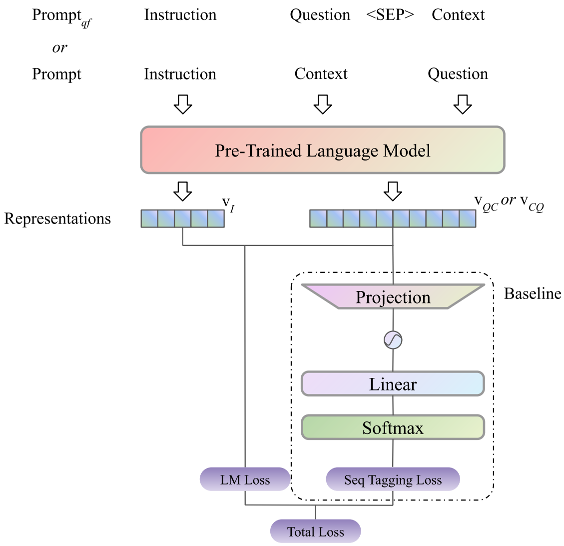

Figure 2: Baseline-embedded model architecture

	MultiSpanQA		SQuAD		Quoref	
	EM F1	**Overlap F1**	**EM**	**F1**	**EM**	**F1**
Llama2	7.354	34.031	13.443	28.931	5.02	28.91
Llama2$_{FT}$	50.934	68.140	36.679	47.055	45.52	52.09
Llama2$_{QASE}$	**51.748**	**70.389**	**37.219**	**47.686**	**54.28**	**60.44**
Alpaca	15.201	42.759	18.259	33.871	-	-
Alpaca$_{FT}$	**52.730**	69.099	27.881	43.950	-	-
Alpaca$_{QASE}$	52.196	**70.008**	**37.313**	**47.622**	-	-
Flan-T5-Small	0.475	22.539	13.878	28.710	1.58	5.96
Flan-T5-Small$_{FT}$	**59.128**	76.494	77.332	85.513	58.21	63.30
Flan-T5-Small$_{QASE}$	59.080	**77.103**	**77.663**	**85.901**	**60.70**	**66.88**
Flan-T5-Base	4.113	37.694	37.596	51.747	27.08	34.38
Flan-T5-Base$_{FT}$	64.659	81.408	82.090	89.558	72.77	80.90
Flan-T5-Base$_{QASE}$	**64.874**	**81.498**	**82.204**	**90.240**	**75.17**	**81.18**
Flan-T5-Large	13.907	51.501	16.149	37.691	15.96	24.10
Flan-T5-Large$_{FT}$	**67.408**	83.094	83.159	90.712	75.17	80.49
Flan-T5-Large$_{QASE}$	66.918	**84.221**	**84.125**	**91.701**	**76.19**	**82.13**

Table 9: Performance of zero-shot PLMs and fined-tuned PLMs with and without *QASE*.

Fine-tuning PLMs	Instruction: Using the provided context, answer the question with exact phrases and avoid explanations. - - - Context: <context> - - - Question: <question> - - - Answer:
Zero-shot prompting PLMs and GPT variants on **single-span** answer dataset, SQuAD	Instruction: Using the provided context, answer the question with exact phrases and avoid explanations. - - - Context: <context> - - - Question: <question> - - - Answer:
Zero-shot prompting PLMs and GPT variants on **multi-span** answer datasets, MultiSpanQA and Quoref	Instruction: Using the provided context, answer the question with exact phrases and avoid explanations. Format the response as follows: ["answer1", "answer2", ...]. - - - Context: <context> - - - Question: <question> - - - Answer:

Table 10: Templates for fine-tuning instructions and zero-shot query prompts