Enhancing Pre-Trained Generative Language Models with Question Attended Span Extraction on Machine Reading Comprehension

Anonymous ACL submission

Abstract

Machine Reading Comprehension (MRC) poses a significant challenge in the field of Natural Language Processing (NLP). While mainstream MRC methods predominantly leverage extractive strategies using encoder-only models such as BERT, generative approaches face the issue of *out-of-control generation* – a critical problem where answers generated are often incorrect, irrelevant, or unfaithful to the source text. To address these limitations in generative models for MRC, we introduce the **Q**uestion-**A**ttended **S**pan **E**xtraction (*QASE*) module. Integrated during the fine-tuning phase of pre-trained generative language models (PLMs), *QASE* significantly enhances their performance, allowing them to surpass the extractive capabilities of advanced Large Language Models (LLMs) such as GPT-4. Notably, these gains in performance do not come with an increase in computational demands. The efficacy of the *QASE* module has been rigorously tested across various datasets, consistently achieving or even surpassing state-of-the-art (SOTA) results. Our code is available at this anonymous repo link.

1 Introduction

Machine Reading Comprehension (MRC), also referred to as text-grounded question answering (QA) (Wang et al., 2022), involves presenting a model with a text passage and a question, requiring it to formulate an answer based solely on the given text. This can be achieved either by identifying a specific span within the text or by generating a concise answer. MRC poses a significant challenge within the domain of Natural Language Processing (NLP). Predominant strategies for addressing MRC employ extractive methods, which typically extract pertinent text snippets from a broader context in response to a query (Wang et al., 2018; Yan et al., 2019; Chen et al., 2020). However, the most precise answers in practical settings often span multiple text passages or necessitate inferential reasoning that extends beyond the surface-level content (Li et al., 2021). Therefore, there is a compelling necessity to integrate generative models alongside extractive approaches to enhance the robustness and comprehensiveness of solutions in this field.

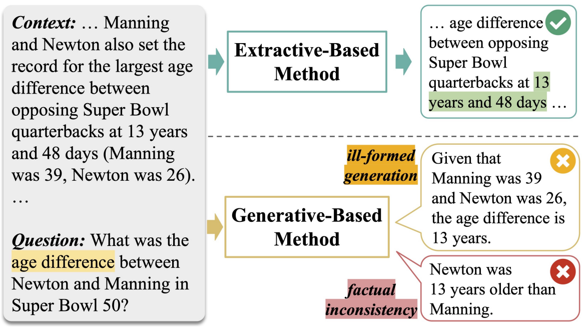

Context: … Manning and Newton also set the record for the largest age difference between opposing Super Bowl quarterbacks at 13 years and 48 days (Manning was 39, Newton was 26). …
Question: What was the age difference between Newton and Manning in Super Bowl 50?
Extractive-Based Method
… age difference between opposing Super Bowl quarterbacks at 13 years and 48 days …
Generative-Based Method
ill-formed generation
Given that Manning was 39 and Newton was 26, the age difference is 13 years.
factual inconsistency
Newton was 13 years older than Manning.

Figure 1: *Out-of-control generation* issue in generative-based methods.

Yet, generative models often fall short in MRC tasks due to a phenomenon known as *out-of-control generation* (Li et al., 2021), which encompasses two primary issues, as illustrated in Figure 1: **(a)** ill-formed generations that include incomplete or redundant phrases, and **(b)** factual inconsistencies that diverge from the intended information. To tackle these issues, this paper introduces the lightweight **Q**uestion-**A**ttended **S**pan **E**xtraction (*QASE*) module. This module is integrated during the fine-tuning of various open-source generative pre-trained language models (PLMs) across multiple MRC datasets to enhance the reliability and accuracy of the generated answers.

Our key contributions are outlined as follows:

1. We develop the *QASE* module to enhance the quality and factual accuracy of answers generated by fine-tuned generative PLMs, achieving performance on par with state-of-the-art (SOTA) extractive methods and surpassing that of advanced Large Language Models (LLMs) such as GPT-4.

2. *QASE* enhances model performance without imposing significant additional computational demands, offering a cost-effective solution for researchers operating under resource constraints.

2 Related Work

Research in MRC In recent research on MRC, there is a predominant focus on extractive question answering using encoder-only PLMs such as BERT and XLM-Roberta. These studies typically involve predicting the start and end positions of answers directly from the context provided (Ohsugi et al., 2019; Lan et al., 2019; Bachina et al., 2021; Chen et al., 2022). Additionally, to accommodate scenarios where answers comprise multiple spans from the text – termed as the multi-span setting – researchers have proposed different strategies. Segal et al. (2020) treated this as a sequence tagging task where each token in the sequence is tagged as being part of an answer span or not. Others (Hu et al., 2019; Lee et al., 2023; Zhang et al., 2023) have experimented with hybrid approaches that combine different modeling techniques or tasks to enhance performance on complex MRC problems. Beyond these extractive methods, there is an emerging interest in applying generative language models(GLMs) for MRC (Yang et al., 2020; Li et al., 2021; Jiang et al., 2022; Su et al., 2022). These models do not just predict the location of an answer within a given text but generate an answer reformulating information found across the context.

Retrieval-augmented text generation (RAG) 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, RAG is typically utilized in scenarios where document retrieval is necessary to reduce input context window (Chen et al., 2024; Ram et al., 2023), whereas selective MRC often requires accessing information beyond the immediate context. 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.

Controllable Text Generation In the domain of controllable text generation, significant progress has been achieved. Gururangan et al. (2020) employ fine-tuning of language models on domain-adaptive text to customize the attributes of generated content. Other promising methodologies include reinforcement learning (Li et al., 2024), contrastive learning (Zheng et al., 2023), and the use of control codes for fine-tuning PLMs (Keskar et al., 2019). Some approaches involve modifying the probability distribution of PLMs. For instance, Liu et al. (2021) propose a technique using two smaller "expert" models to refine the PLM's output, while Yang and Klein (2021) introduce a method that conditions the generation process using a "future discriminator" based on predicted outcomes. Moreover, Huang et al. (2023) explore efficient multi-aspect text generation with trainable gates for enhanced control. *QASE* represents a novel adaptation of controlled text generation tailored to the specific challenges of MRC, with a focus on the precision and relevance of generated answers. Unlike methods that modify the overall generative process through complex architectural alterations or additional learning mechanisms, *QASE* directly utilizes the question context to guide the extraction and generation phases.

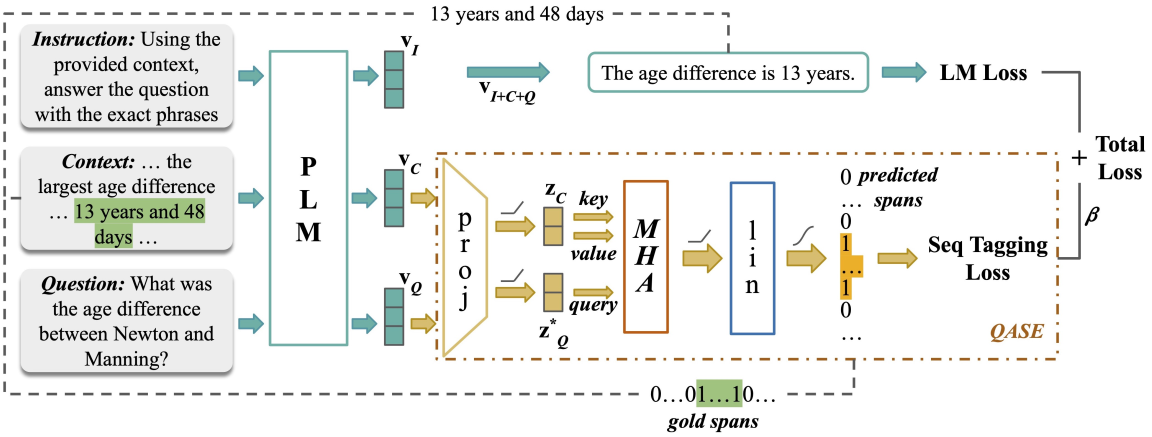
3 Method

This section presents our proposed *QASE* module and the multi-task fine-tuning strategy we employ.

3.1 Question-Attended Span Extraction

To guide text generation, we employ the *QASE* module, a question-attended span extraction tool, during the fine-tuning of generative PLMs. *QASE* directs model focus to potential answer spans within the original text. We frame span extraction as a sequence tagging task using the Inside-Outside (IO) tagging schema. In this schema, each token is labeled as 'inside' (***I***) if it falls within a relevant span, or 'outside' (***O***) otherwise. This approach effectively handles both single- and multi-span extractions and has shown to perform on par with or better than the well-known BIO format (Huang et al., 2015), as demonstrated by Segal et al. (2020).

The model architecture is depicted in Figure 2. Initially, a context and question pair along with an instruction are tokenized and input into the PLM. The resultant hidden states from the PLM are then transformed through projection layers to generate embeddings $z_i = ReLU(W_{proj}v_i + b_{proj})$, where



Instruction: Using the provided context, answer the question with the exact phrases

Context: … the largest age difference … 13 years and 48 days …

Question: What was the age difference between Newton and Manning?

The age difference is 13 years.

Figure 2: Architecture of the *QASE*-enhanced model. Here, z^*_Q represents the averaged embedding of question tokens, expanded to match the length of z_C.

$v_i \in R^d$ represents the hidden state of the i^{th} token from the PLM output.

To capture the relationship of context tokens to specific questions, we utilize a multi-head attention mechanism (***MHA***). Each attention head targets different aspects of the context in relation to the question, treating question embeddings as queries and context embeddings as keys and values. Specifically, for each question-context pair, we compute a mean question embedding by averaging the embeddings of question tokens, which is then expanded to align with the length of the context sequence. This expanded question embedding, z^*_Q, serves as the query in the ***MHA***, with the context embedding, z_C, acting as both key and value. This mechanism allows the derived representation of each token in the context to encapsulate its relevance in relation to the posed question.

In conclusion, the *QASE* module processes the projected embeddings z_C and z^*_Q through the ***MHA*** mechanism, followed by a linear and a softmax layer to calculate the probability that each context token belongs to an answer span:

$$p_{C_i} = softmax(W_{lin} \cdot MHA(z^*_{Q_i}, z_{C_i}, z_{C_i}) + b_{lin})$$

This probability is represented by p_{C_i} for the i^{th} context token. To measure the accuracy of span prediction, we compute sequence tagging loss employing cross-entropy loss:

$$L_{QASE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=0}^{1} y_{ij} log(p_{C_{ij}})$$

where $j \in 0, 1$ designates the classes ***O*** and ***I***, and y_{ij} is a binary indicator of whether the i^{th} token is labeled as class j.

3.2 Fine-Tuning and Inference

We fine-tune the PLMs employing a multi-task learning strategy that concurrently optimizes both the language modeling loss and the sequence tagging loss:

$$L = L_{LML} + \beta L_{QASE}$$

where β is a hyper-parameter that determines the weight assigned to the span extraction task. This dual-objective approach substantially improves the PLMs' capability to generate contextually grounded and relevant answers. During the inference phase, only the generation component of the finely-tuned model is utilized.

4 Experiments

This section presents the experimental framework, detailing the datasets used, experimental setup, comprehensive quantitative results of model performance, ablation studies, analysis of model factual consistency, and qualitative case studies.

4.1 Datasets and Metrics

We utilize three MRC benchmark datasets:

(1) **SQuAD** (Rajpurkar et al., 2016): A benchmark 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 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 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 (listed in 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.

4.2 Experimental Setup

To assess the efficacy of the *QASE* module independent of any specific language models, we conduct experiments with multiple open-source LLMs. Our tests include both decoder-only LLMs, such as Llama 2 (Touvron et al., 2023) and Alpaca (Taori et al., 2023), and an encoder-decoder model family, Flan-T5 (Chung et al., 2022). For Llama 2 and Alpaca, we employ the pre-trained 7B version and fine-tune it using LoRA (Hu et al., 2021) combined with instruction-tuning (instruction templates are detailed in A.4). For the Flan-T5 family, we fine-tune the small, base, and large versions. Detailed information about the trainable parameters for each model is provided in Table 1.

	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 1: Trainable parameters of experimented models.

We determine the hyper-parameter $\beta = 1$ and the learning rate $lr = 1e - 4$ using results from a grid search. For the LoRA fine-tuning of the Llama 2 and Alpaca models, we set a rank $r = 8$, $\alpha = 32$, and a dropout rate of 0.05. The methodology for selecting these hyper-parameters is detailed in A.2. All models are trained on individual GPUs with batch sizes ranging from 2 to 4, adjusted according to each GPU's VRAM capabilities. We employ four types of GPUs: A40, A10, A5500, and A100. Training continues for three epochs or until the models converge. Consistency is maintained across all variants of each base PLM in terms of GPU type, batch size, and training epochs.

4.3 Experiment Results

To assess the *QASE* module, we compare the performance of various PLMs fine-tuned with and without *QASE*, as detailed in Table 2. Overall, models fine-tuned with *QASE* consistently outperform those without it. Specifically, for the SQuAD dataset, models with *QASE* show an exact match (EM) percentage increase of up to 33.8% and an F1 score improvement of up to 8.4% compared to vanilla fine-tuned models. For MultiSpanQA, improvements include up to 1.6% in EM F1 and up to 3.3% in overlap F1. Likewise, on the Quoref dataset, enhancements of up to 19.2% in EM percentage and up to 16.0% in F1 score are observed. These results confirm that *QASE* enables generative-based PLMs to produce more accurate, contextually coherent, and higher-quality answers in MRC tasks compared to vanilla fine-tuning approaches.

For additional comparisons, we also evaluate the fine-tuned PLMs against their zero-shot performance, as outlined in Appendix A.3. Specifically, on the SQuAD dataset, models using *QASE* perform up to 5.6 times better in EM and 3.0 times better in F1 score compared to the zero-shot models. On the MultiSpanQA dataset, the EM improves by up to 124.4 times, and F1 score by up to 3.4 times. Similarly, on the Quoref dataset, the EM improves by up to 38.4 times, and F1 score by up to 11.2 times with *QASE*. It is important to note that these substantial improvements stem from comparing zero-shot models to those fine-tuned with *QASE*. Nonetheless, the previously discussed results comparing fine-tuned models with and without *QASE* have clearly illustrated its effectiveness.

4.4 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.4.

On SQuAD, as showed in Table 3, Flan-T5-Large$_{QASE}$ surpasses human performance, equaling the NLNet model from Microsoft Research

		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 2: Performance (in %) of fine-tuned PLMs with or without *QASE* on each dataset.

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

Asia and the original pre-trained BERT-Large from Google (Devlin et al., 2019). Additionally, it surpasses GPT-4 by 113.8% on the exact match score and 32.6% on 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 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 ↑
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**.

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.

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.

4.5 Ablation Studies

We conduct ablation studies to assess the effectiveness of the *QASE* architecture and to determine the optimal prompting strategy. Specifically, we compare Flan-T5-Large$_{QASE}$ with both the vanilla fine-tuned Flan-T5-Large$_{FT}$ and the baseline Flan-T5-Large$_{baseline}$. As shown in Figure 3 in Appendix A.5, the baseline span extraction module does not include the ***MHA*** component, rendering it a conventional architecture for fine-tuning pre-trained encoders on downstream sequence tagging tasks. For each configuration – Flan-T5-Large$_{FT}$, Flan-T5-Large$_{QASE}$, and Flan-T5-Large$_{baseline}$ – we explored both a question-first (*qf*) and a context-first prompting strategy, with a detailed description of these strategies provided in Appendix A.5.

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**.

4.6 Computational Cost

To assess the computational cost associated with *QASE*, Table 1 reveals that incorporating the *QASE* module incurs only a slight increase in the number of trainable parameters in PLMs. The degree of this increase varies based on the hidden sizes of the models. Remarkably, for the largest model, Flan-T5-Large, the addition of *QASE* accounts for merely an extra 0.2% in parameters. This underscores that *QASE* can substantially boost the performance of fine-tuned PLMs in MRC tasks without requiring significant additional computational resources.

4.7 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.

4.7.1 Qualitative Case Studies

In addition to the Q^2 statistical analysis, we also perform qualitative case studies to further demonstrate the effectiveness of *QASE* in generating factual consistent answers.

Sample 1	
Context: This was the first Super Bowl to feature a quarterback on both teams who was the #1 pick in their draft classes. Manning was the #1 selection of the 1998 NFL draft, while Newton was picked first in 2011. The matchup also pits the top two picks of the 2011 draft against each other: Newton for Carolina and Von Miller for Denver. Manning and Newton also set the record for the largest age difference between opposing Super Bowl quarterbacks at 13 years and 48 days (Manning was 39, Newton was 26). **Question:** What was the age difference between Newton and Manning in Super Bowl 50? **Gold Answer:** 13 years and 48 days	
Flan-T5-Large$_{QASE}$ Generation	13 years and 48 days
Flan-T5-Large$_{FT}$ Generation	26

Sample 2	
Context: However, this definition is disputed by Thoreau's political philosophy, which contrasts the conscience with the collective. The individual is the ultimate arbiter of right and wrong. Beyond this, since only individuals act, only they can commit injustices. ... Thoreau acknowledges that the government may represent the will of the majority but it might also merely reflect the desires of elite politicians. Even a good government is "liable to be abused and perverted before the people can act through it." Furthermore, even if a government did express the voice of the people, this fact would not obligate the obedience of individuals who dissent. The majority may be powerful but it is not necessarily right. What, then, is the appropriate relationship between the individual and the government? **Question:** What did Thoreau claim about the majority? **Gold Answer:** not necessarily right	
Flan-T5-Large$_{QASE}$ Generation	it is not necessarily right
Flan-T5-Large$_{FT}$ Generation	conscience vs. the collective

Table 8: Comparisons of model attention alignment with question key aspects and relevant factual context between Flan-T5-Large$_{QASE}$ and Flan-T5-Large$_{FT}$.

Question Attended Alignment Table 8 showcases that Flan-T5-Large$_{QASE}$ more accurately identifies the key focus of the question and locates the pertinent factual information within the context, with the aid of the *QASE* module. For instance, in **Sample 1**, Flan-T5-Large$_{QASE}$ correctly interprets the question as seeking the age difference between Newton and Manning, rather than the age of either individual, and accordingly provides the accurate answer. In contrast, Flan-T5-Large$_{FT}$ mistakenly provides Newton's age as the answer. Similarly, in **Sample 2**, Flan-T5-Large$_{QASE}$ accurately discerns that the question pertains to Thoreau's claim regarding the majority, generating in the correct answer, whereas Flan-T5-Large$_{FT}$ misguidedly responds with Thoreau's political philosophy.

Multi-Span Answers Flan-T5-Large$_{QASE}$ also shows a notable improvement in comprehending complex, lengthy sentences and synthesizing answers from information that is sparsely distributed across multiple spans requiring logical processing. This capability is particularly valuable when the answer to a question does not directly stem from a single phrase. Table 9 provides examples of such instances. In **Sample 3**, the model needs to recognize that ESPN Deportes is the exclusive broadcaster in Spanish and that CBS, although mentioned, does not offer Spanish-language broadcasting. Combining these facts leads to the correct answer, that ESPN Deportes is the network that broadcast the game in Spanish. Flan-T5-Large$_{QASE}$ accurately generates this answer, whereas Flan-T5-Large$_{FT}$ incorrectly answers with "CBS", likely due to confusion caused by the complex sentence structures and dispersed information. Similarly, in **Sample 4**, Flan-T5-Large$_{QASE}$ correctly identifies the question as seeking the name of the force related to a potential field between two locations. It successfully locates the relevant long sentence, deconstructs, and comprehends it to produce the correct answer, in contrast to Flan-T5-Large$_{FT}$, which incorrectly selects the first phrase mentioning "force". In **Sample 5**, the question asks for the class most commonly not ascribed to the graph isomorphism problem. The model needs to deduce from the context that "it is widely believed that the polynomial hierarchy does not collapse to any finite level", implying "graph isomorphism is not NP-complete". Once again, Flan-T5-Large$_{QASE}$ arrives at the correct conclusion, while Flan-T5-Large$_{FT}$ does not.

Real-World Knowledge While our primary evaluation focuses on the model's proficiency in deriving answers from provided contexts, we also note that *QASE* enhances the model's capacity to leverage real-world knowledge acquired during its pre-training phase. This improvement is attributed to *QASE*'s ability to better align the model's focus on parts of the context that are relevant to the questions asked. Table 10 presents an example of this phenomenon. In **Sample 6**, when asked about the California venue considered for the Super Bowl, Flan-T5-Large$_{QASE}$ correctly associates the San Francisco Bay Area with California, thus producing the accurate answer. On the other hand,

Sample 3

Context: On December 28, 2015, ESPN Deportes announced that they had reached an agreement with CBS and the NFL to be the exclusive Spanish-language broadcaster of the game, marking the third dedicated Spanish-language broadcast of the Super Bowl. Unlike NBC and Fox, CBS does not have a Spanish-language outlet of its own that could broadcast the game (though per league policy, a separate Spanish play-by-play call was carried on CBS's second audio program channel for over-the-air viewers). ...

Question: Which network broadcast the game in Spanish?

Gold Answer: ESPN Deportes

Flan-T5-Large$_{QASE}$ Generation	ESPN Deportes
Flan-T5-Large$_{FT}$ Generation	CBS

Sample 4

Context: A conservative force that acts on a closed system has an associated mechanical work that allows energy to convert only between kinetic or potential forms. This means that for a closed system, the net mechanical energy is conserved whenever a conservative force acts on the system. The force, therefore, is related directly to the difference in potential energy between two different locations in space, and can be considered to be an artifact of the potential field in the same way that the direction and amount of a flow of water can be considered to be an artifact of the contour map of the elevation of an area.

Question: What is the force called regarding a potential field between two locations?

Gold Answer: an artifact

Flan-T5-Large$_{QASE}$ Generation	an artifact
Flan-T5-Large$_{FT}$ Generation	conservative force

Sample 5

Context: The graph isomorphism problem is the computational problem of determining whether two finite graphs are isomorphic. An important unsolved problem in complexity theory is whether the graph isomorphism problem is in P, NP-complete, or NP-intermediate. The answer is not known, but it is believed that the problem is at least not NP-complete. If graph isomorphism is NP-complete, the polynomial time hierarchy collapses to its second level. Since it is widely believed that the polynomial hierarchy does not collapse to any finite level, it is believed that graph isomorphism is not NP-complete. The best algorithm for this problem, due to Laszlo Babai and Eugene Luks has run time $2O(\sqrt{nlog(n)})$ for graphs with n vertices.

Question: What class is most commonly not ascribed to the graph isomorphism problem in spite of definitive determination?

Gold Answer: NP-complete

Flan-T5-Large$_{QASE}$ Generation	NP-complete
Flan-T5-Large$_{FT}$ Generation	NP-intermediate

Table 9: Comparison of Flan-T5-Large$_{QASE}$ and Flan-T5-Large$_{FT}$ in understanding complex sentence structures.

Flan-T5-Large$_{FT}$ erroneously identifies a stadium in Miami as the answer. This example illustrates how *QASE* not only improves context-based answer generation but also the model's application of pre-existing real-world knowledge to the questions posed.

Sample 6	
Context: The league eventually narrowed the bids to three sites: New Orleans' Mercedes-Benz Superdome, Miami's Sun Life Stadium, and the San Francisco Bay Area's Levi's Stadium. **Question:** Which California venue was one of three considered for Super Bowl 50?	
Gold Answer: San Francisco Bay Area's Levi's Stadium	
Flan-T5-Large$_{QASE}$ Generation	San Francisco Bay Area's Levi's Stadium
Flan-T5-Large$_{FT}$ Generation	Sun Life Stadium

Table 10: Comparison of Flan-T5-Large$_{QASE}$ and Flan-T5-Large$_{FT}$ in utilizing real-world knowledge.

5 Discussions

In this section, we briefly address the weak performance of Flan-T5 zero-shot and Llama 2 on MRC tasks, despite their strong language understanding abilities. We note that a comprehensive analysis is beyond our study's scope. Our goal is to gain insights into further improving these PLMs' effectiveness in MRC.

5.1 Flan-T5 Zero-Shot Performance

Despite being trained on SQuAD during pre-training, Flan-T5 models demonstrate poor performance across datasets, including SQuAD. While a comprehensive analysis of Flan-T5's performance is beyond the focus of our study, we briefly explore potential reasons for this underperformance to gain better insights. This underperformance may stem from their training on a wide range of tasks (1,836 tasks), focusing on free-form generation, QA, and reasoning tasks, rather than being finely optimized for extractive QA tasks like MRC. Additionally, generative models like Flan-T5 and Llama 2 generally struggle in MRC tasks, as discussed earlier. For extended discussions, refer to Appendix B.1.

For fairness in our zero-shot experiments, we compare our prompt template with Google's instruct-tuning prompts for Flan-T5 on the SQuAD v1 dataset. Our results, as illustrated in Table 14, reveal that our prompt template achieves the highest F1 score. This implies that Flan-T5's lower zero-shot performance on MRC is expected.

5.2 Llama 2 Performance

We also observe that models based on Llama 2 and Alpaca consistently underperform compared to those based on Flan-T5, across zero-shot and fine-tuned scenarios, with or without *QASE*. This discrepancy may arise from the significant difference in the number of trainable parameters, as illustrated in Table 1, during fine-tuning. Additionally, factors such as differences in pre-training datasets and varied adaptation to tasks due to structural disparities can also contribute to this performance gap. While acknowledging these factors, conducting a comprehensive comparison of different generative model architectures in MRC tasks exceeds the scope of our study. For further discussion, please refer to Appendix B.2.

5.3 Prompting Strategies

While we acknowledge the prevalent use of prompting strategies like in-context learning, our main focus is not on evaluating these strategies but rather on improving the performance of generative PLMs on MRC tasks through fine-tuning, with and without the integration of *QASE*. However, we do experiment with various prompt templates, as discussed in Section 4.5 and Appendix B.1.

6 Conclusion and Future Work

In this study, we address *out-of-control generation* issue 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 aim to evaluate our model on generative MRC datasets (Nguyen et al., 2016) to gauge its effectiveness in handling more intricate scenarios. Additionally, a significant emphasis will be placed on assessing the model's overall capability in answer generation, with a specific focus on human perception. This involves incorporating human annotators alongside automatic metrics. Looking further ahead, we aspire to extend our research to explore strategies for mitigating 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 1. 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.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17754–17762.

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*.

Nuo Chen, Linjun Shou, Ming Gong, and Jian Pei. 2022. From good to best: Two-stage training for cross-lingual machine reading comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10501–10508.

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.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

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.

Xuancheng Huang, Zijun Liu, Peng Li, Tao Li, Maosong Sun, and Yang Liu. 2023. An extensible plug-and-play method for multi-aspect controllable text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15233–15256, Toronto, Canada. 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*.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2022. Understanding and improving zero-shot multi-hop reasoning in generative question answering. *Preprint*, arXiv:2210.04234.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *Preprint*, arXiv:1909.05858.

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*.

Wendi Li, Wei Wei, Kaihe Xu, Wenfeng Xie, Dangyang Chen, and Yu Cheng. 2024. Reinforcement learning with token-level feedback for controllable text generation. *Preprint*, arXiv:2403.11558.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. 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 6691–6706, Online. Association for Computational Linguistics.

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.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-Context Retrieval-Augmented Language Models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

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*.

Luqi Wang, Kaiwen Zheng, Liyin Qian, and Sheng Li. 2022. A survey of extractive question answering. In *2022 International Conference on High Performance Big Data and Intelligent Systems (HDIS)*, pages 147–153. IEEE.

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*.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

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*.

Chujie Zheng, Pei Ke, Zheng Zhang, and Minlie Huang. 2023. Click: Controllable text generation with sequence likelihood contrastive learning. *Preprint*, arXiv:2306.03350.

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 Detailed Experiment Setup and Results

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 11: Dataset official leaderboards.

A.2 Hyper-Parameter Selection

In this section, we outline the process for selecting the hyper-parameter β and detail our approach to LoRA fine-tuning.

For selecting β, we use a grid search method, exploring values from 0.5 to 2 in increments of 0.1, on 30% of the MultiSpanQA training dataset. This process leads to the determination that $\beta = 1$ empirically yield the best performance, hence it is selected for use in our experiments.

To select the learning rate lr, we conduct a grid search, testing values from $\{1e-5, 5e-5, 1e-4, 5e-4, 1e-3\}$ on 30% of the MultiSpanQA training dataset. Empirically, the value $1e-4$ demonstrates the best performance and is therefore chosen for our experiments. This selection is in agreement with the default lr value used in Meta's official Llama 2 fine-tuning recipe[1].

In the case of LoRA fine-tuning, we follow the established methodology as outlined by Hu et al. (2021). This involves applying LoRA to Llama 2 and the pre-trained Alpaca models by freezing their pre-trained weights and integrating trainable rank decomposition matrices at every layer of their Transformer structures, aimed at reducing the number of trainable parameters to enhance computational efficiency. We implement this using the PEFT package[2]. The fine-tuning hyper-parameters for LoRA are set according to the default settings specified in Meta's official Llama 2 fine-tuning recipe[3], which include a rank $r = 8$, $\alpha = 32$, and a dropout rate of 0.05.

[1] Link to the fine-tuning configuration of Meta's official Llama 2 recipe.

[2] Link to the Hugging Face PEFT implementation.

[3] Link to the LoRA hyper-parameter configuration of Meta's official Llama 2 recipe.

A.3 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 12, 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.4 Instruction Templates and Model Prompts

Table 13 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.5 Ablation Studies Details

Figure 3 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.

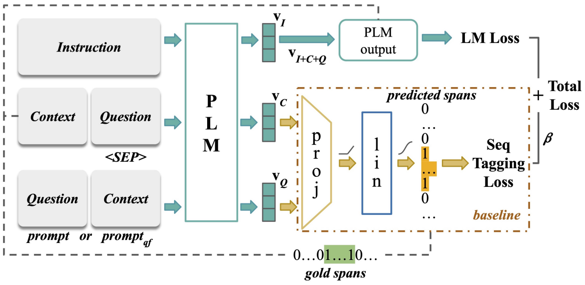


Figure 3: Baseline-embedded model architecture.

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.

B Extended Discussion on Model Performance

In this section, we engage in a detailed discussion on the performance of the Flan-T5 family of models and Llama 2 in MRC tasks. Our aim is to gain insights into the reasons behind the modest zero-shot performance of these large PLMs on MRC tasks, despite their adeptness at handling other complex NLP tasks such as dialogue generation and summarization. Although a comprehensive analysis falls outside the scope of our current study, exploring these performance nuances can provide valuable perspectives on how to potentially enhance the effectiveness of these PLMs on similar tasks.

B.1 Discussion on Flan-T5 Zero-Shot Performance

We observe that the zero-shot performance of Flan-T5 models across all datasets, including SQuAD, remains low as shown in Table 12, despite being instruct-tuned on the SQuAD dataset during the pre-training phase. This underperformance might stem from the fact that Flan-T5 models, although trained on the <SQuAD, Extractive QA> task, are also trained on a broad spectrum of 1,836 tasks, predominantly focusing on free-form generation, QA, and reasoning tasks (Chung et al., 2022). Consequently, these models are not finely optimized for extractive QA tasks like MRC, especially under metrics like exact match and F1, particularly for the smaller to larger variants under study. The larger XL and XXL variants may exhibit better performance in these tasks. Furthermore, as discussed in the previous sections, generative models, including Llama 2, Alpaca, and GPT variants, generally show limited effectiveness in MRC tasks in zero-shot settings, underscored by their poorer performance despite having significantly larger model parameters compared to the Flan-T5 variants we experiment with.

	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 12: 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 13: Templates for fine-tuning instructions and zero-shot query prompts

To ensure that our zero-shot experiment's prompts do not adversely affect Flan-T5's performance, we compare our prompt template, detailed in Table 13, with those Google released for Flan-T5's instruct-tuning on the SQuAD v1 dataset[4]. Our template, similar to Google's, differs mainly by including "with exact phrases and avoid explanations." This difference could potentially affect performance, yet our subsequent experiments demonstrate otherwise.

We conduct a series of experiments to assess the zero-shot performance of Flan-T5-Large on SQuAD, using Google released templates for Flan-T5 instruct-tuning. We select three templates of varying complexities, as listed in Table 14. Our results, detailed in Table 14, reveal that our template achieves the highest F1 score. This indicates the lower performance of zero-shot Flan-T5 on SQuAD and similar MRC datasets is expected, even with the original instruct-tuning templates. It supports our hypothesis that, although Flan-T5 is instruct-tuned on SQuAD, its primary strengths are in broader generative question answering and reasoning, rather than specific extractive QA tasks such as MRC, particularly when evaluated by exact match and F1 metrics.

	SQuAD Performance	
Prompt Template	**EM**	**F1**

Article: {context}
Question: {question}
Answer:
	7.001	21.717

Answer a question about this article.
Article: {context}
Question: {question}
Answer:
	15.875	33.375

Here is a question about this article: Article: {context} What is the answer to this question: Question: {question} Answer:	**16.764**	35.304
Our Template See Table 13	16.149	**37.691**

Table 14: Flan-T5-Large zero-shot performance on SQuAD with different prompt templates.

B.2 Discussion on Llama 2 Performance

We observe that models based on Llama 2 and Alpaca generally underperform compared to those based on Flan-T5, in both zero-shot and fine-tuned scenarios, with or without *QASE*. This section delves into a detailed discussion of the potential reasons behind this trend.

Firstly, the discrepancy in performance may stem from the inherent structural differences between decoder-only models (Llama 2 and Alpaca) and encoder-decoder models (Flan-T5). Encoder-decoder models are better equipped for tasks that require extensive input processing, such as MRC, making them more apt for these tasks than decoder-only models, which are typically more suited to open-ended QA scenarios. This fundamental distinction partially accounts for Flan-T5's superior performance in context-based question answering across both zero-shot and fine-tuned settings.

Additionally, the difference in the number of trainable parameters during fine-tuning might contribute to the observed performance gap. Table 1 indicates that fine-tuning Llama 2 and Alpaca with LoRA leads to a significantly lower count of trainable parameters (4.2M) compared to even the smallest Flan-T5 model (77.0M). This disparity in trainable parameters is a crucial factor in explaining why fine-tuned Flan-T5 models, irrespective of the use of QASE, outperform Llama 2 and Alpaca models.

While we address these factors, conducting a comprehensive comparison and analysis of different generative model architectures in MRC tasks exceeds the scope of our current study. Nonetheless, we acknowledge that additional factors, such as the specific instruct-fine-tuning of Flan-T5 models on MRC datasets like SQuAD, might also play a role in their enhanced performance over Llama 2 and Alpaca.

[4] Link to Flan-T5 instruct-tuning prompt templates.