When to Retrieve? Teaching LLMs to Utilize Information Retrieval Effectively

Anonymous ACL submission

Abstract

Recently, systems that combine Information Retrieval (IR) with Large Language Models (LLMs), such as RAG, have demonstrated remarkable capabilities in question answering by integrating external context. However, the optimal strategy for question answering does not always involve retrieving external information; it often involves leveraging the LLM's own parametric memory. In this paper, we demonstrate how LLMs can be effectively trained to determine when additional context is necessary and to utilize an off-the-shelf IR system accordingly. We propose a tailored training approach where LLMs, using open-domain question answering datasets, learn to generate a special token, ⟨RET⟩, when they do not know the answer to a question. Our evaluation of the Adaptive Retrieval LLM (ADAPT-LLM) on the PopQA dataset showcases improvements over the same LLM under three configurations: (i) retrieving information for all questions, (ii) relying solely on the LLM's parametric memory, and (iii) using a popularity threshold to decide when to use a retriever.

1 Introduction

The task of question answering (QA) remains a focal point in Natural Language Understanding research. There are many different datasets serving as benchmarks for evaluating QA models, such as Natural Questions (NQ) (Kwiatkowski et al., 2019), SQuAD (Rajpurkar et al., 2016) or QuAC (Choi et al., 2018), just to mention a few. Nowadays, Large Language Models (LLMs) consistently outperform traditional methods on these benchmarks, showcasing remarkable performance.

Typically, there are two primary approaches to utilize LLMs for question answering: (i) **Closed Book Question Answering**: the LLM relies solely on its parametric memory to answer questions. However, these parametric memories have inherent limitations as they are based entirely on the training corpus, meaning for example that they could be outdated regarding events occurring after the training process. (ii) **Open Book Question Answering**: the LLM is coupled with an Information Retriever (IR) system (Izacard and Grave, 2021; Zhu et al., 2021). By leveraging the IR system, the LLM can retrieve relevant context to provide more accurate answers. However, the research conducted by Mallen et al. (2023) sheds light on the complexity of question-answering strategies, challenging the notion that the optimal approach always involves the utilization of an IR system. Through the introduction of the PopQA dataset they demonstrated that while LLMs relying solely on their parametric memories excel in addressing high-popularity questions, the efficacy diminishes for low-popularity questions, where using IR becomes crucial. In many cases, however, question answering datasets do not include popularity scores, so relying on such scores is not a generalizable approach. On top of it, popularity is dynamic and a topic that was popular at the LLM training time could be not trending anymore at inference time. Motivated by this limitation, our study aims to address whether LLMs can autonomously determine when to employ an IR system for improved question answering. To investigate this, we conduct an evaluation of an LLM using an open-domain question answering dataset to identify the questions for which the LLM provides accurate responses and those where its answers are incorrect. For questions where the LLM's response is incorrect, we annotate them with a special token, ⟨RET⟩, indicating the need for additional context. Subsequently, we utilize these annotations to construct a new dataset tailored for training purposes, where we teach an LLM to answer directly if it is confident about the answer or to require context it believes is useful for answering the question (see Figure 1). Our hypothesis is that through this training process, the LLM learns to use an IR system when it needs extra context to answer a question,

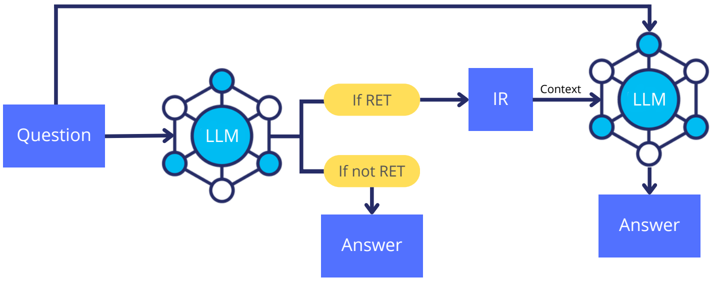


Figure 1: The inference process of ADAPT-LLM step-by-step: given a question (step 1), an LLM decides (step 2) whether to answer the question directly (step 3) or to ask for additional contextual information, generating the special ⟨RET⟩ token; for the later, an off-the-shelf IR system is used to retrieve relevant context (step 4), which is used alongside the question to prompt again the LLM for the final answer (step 5).

thus we name it ADAPT-LLM.

To validate our hypothesis, we conducted several experiments on the PopQA dataset (Mallen et al., 2023), as it provides a suitable platform for benchmarking hybrid retrieval strategies. As a result of these experiments we find that: (i) ADAPT-LLM consistently outperforms typical fixed strategies for question answering, such as using the IR system for all questions and relying solely on the parametric memory of the LLM; (ii) ADAPT-LLM demonstrates performance comparable to strategies that rely on popularity scores to determine when to use an IR system, even without utilizing any popularity score or similar metric. Our findings underscore the significance of adaptive retrieval strategies in enhancing the performance of LLMs for question answering tasks. By training ADAPT-LLM to dynamically determine when to retrieve additional context, we demonstrate the feasibility of teaching an LLM how to effectively leverage external information sources only when necessary.

2 Related Work

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) has shown improvements on a wide variety of NLP areas, such as question answering (Karpukhin et al., 2020; Izacard and Grave, 2021; Seonwoo et al., 2022; Nakano et al., 2021), truthfulness (Ji et al., 2023; Lin et al., 2022) and language modelling (Guu et al., 2020; Borgeaud et al., 2022; Ram et al., 2023) among others. The ability to ground model generations on retrieved text chunks has also enabled smaller models to match the performance of larger ones (Catav et al., 2024). Moreover, due to the extremely high cost of training LLMs, RAG has become the standard way to maintain them updated with new information, not having to re-train the models periodically to incorporate new facts (Gao et al., 2023). Even if augmenting LLMs with retrieval is an essential step for the current generation of LLMs (Jiang et al., 2024; Reid et al., 2024) it also comes with a cost. Traditional retrieval methods as TF-IDF or BM-25 (Robertson et al., 2009) are only able to retrieve documents with keyword overlap and suffer from lexical gap (Berger et al., 2000). In order to try to solve this issue, many pre-trained Transformer encoder based dense models have been proposed (Gao et al., 2021; Reimers and Gurevych, 2019; Karpukhin et al., 2020; Gautier et al., 2022). Trained neural models have shown good performance over a variety of retrieval benchmarks but they still struggle in the zero-shot setup for new domains (Thakur et al., 2021). The quality of the retrieval engine is essential for retrieval-augmented models as this will set the upper bound of the model performance. Moreover, the usage of a retrieval engine, especially when the target document index is huge, can significantly increase the latency of the model and hurt real time applications user experience (Barnett et al., 2024). On the other hand, as models keep scaling, the world knowledge encoded in their parameters does too (Kaplan et al., 2020). Many previous efforts have shown that language models are able to memorize a significant amount of world knowledge and achieve competitive performance on tasks such as open-domain question answering when they just use their parametric knowledge for solving the task (Liang et al., 2023; Achiam et al., 2023; Dubey et al., 2024; Touvron et al., 2023b). Motivated by all this, the adaptive approach has been proposed as a new solution (Schick et al., 2024; Mallen et al., 2023). In this approach, if the solution to the task is encoded in the parameters of the model, the model will be directly used for generating a solution. Conversely, if the answer is not encoded in the knowledge of the model, the answer generation will be augmented with external knowledge.

Recently, Schick et al. (2024) proposed the Toolformer, a model that can self teach how and when to use external tools via simple API calls including a calculator, search engines, a calendar and so on. More similar to our work, Mallen et al. (2023) propose a dataset and method for measuring when non-parametric information needs to be retrieved. They present the PopQA dataset that contains 14K questions about a set of entities with varying popularity. The popularity of an entity is measured by the page views of its Wikipedia page. In order to solve this QA task, they use a popularity score threshold calculated on the PopQA dataset. If the popularity score of an individual entity is below the threshold they perform a retrieval step. On the contrary, if the score is greater than the threshold they directly answer the question. This method yields better results than vanilla retrieval but it requires the calculation of a popularity score that is not available in realistic QA scenarios.

Another relevant contribution in this field, contemporaneous with our research, is the work by Erbacher et al. (2024), where they trained an LLM to determine when to utilize external knowledge. They particularly focused on finding the optimal trade-off between the risk of hallucination and the cost of information retrieval, given the potentially high expense associated with IR. Our ADAPT-LLM method adopts a similar approach, training an LLM to learn when to retrieve information. However, we extend this by comparing our method's performance against some baselines, and assess the effectiveness of retrieving information in an adaptive manner against the strategies of never retrieving or always retrieving.[1]

3 Adaptive Retrieval LLM (ADAPT-LLM)

Adaptive retrieval refers to the model's capability to dynamically determine whether to retrieve additional context information for generating answers in question answering tasks. Unlike traditional models that either always incorporate context or never consider it, adaptive retrieval allows the model to selectively retrieve context based on the specific requirements of each question. This adaptive approach aims to optimize performance by leveraging context only when necessary, thereby enhancing the model's ability to generate accurate answers. As depicted in Figure 1, the process of the ADAPT-LLM unfolds in the following sequence:

1. The first prompt containing the question is sent to the model (step 1 of Figure 1).

2. The ADAPT-LLM evaluates the prompt to determine whether additional context is necessary to answer the question effectively (step 2).

3. If the model determines that context is not required, it directly produces a response to the question by leveraging its parametric memory (step 3).

4. If context is deemed necessary, the ADAPT-LLM model returns a special token, represented as $\langle$RET$\rangle$, and an off-the-shelf IR system is used to retrieve pertinent context based on the question (step 4); the context is then combined with the original question prompt to form a comprehensive representation for answer generation (step 5).

This decision-making process enables the model to determine whether context is needed, balancing between using context for better understanding and providing direct answers when appropriate.

3.1 Training ADAPT-LLM

In this section, we outline the methodology for training our ADAPT-LLM model. This process, denoted as DS_{Adapt}, is presented in the algorithm at Appendix B. We start with an open-domain question answering dataset containing questions Q, context passages P, and answers A, initializing DS_{Adapt} to an empty set. For each question in Q, we leverage the base LLM without any retrieval mechanism to perform a zero-shot inference. This step allows us to differentiate questions for which the model generates correct answers from those where its responses are inaccurate. For questions where the model's response is accurate, we build a training set instance incorporating the following prompt, which we call *parametric_prompt*:

[1] All resources are publicly available at https://github.com/mwozgpt/Adapt-LLM-anonymous.

```
Prompt: Answer the question Q. If you need
help answer <RET> to get the context. Q:
{...}
```

Alongside this prompt, we include the corresponding question from Q and the golden answer from A, collectively forming the instance, which is subsequently appended to the DS_{Adapt} dataset. In contrast, if the LLM fails to produce a correct response to the question, we build two different instances. The first employs the same *parametric_prompt* as previously described, with $\langle$RET$\rangle$ as the answer, indicating the necessity for additional context. The second, called *context_prompt*, includes contextual information alongside the question:

```
Prompt: Answer the question Q given the
context C. Q: {...}, C: {...}
```

For this instance, we include the prompt, the question from Q, the golden answer from A, and the corresponding context passage from P. After populating the dataset with both types of prompts for questions where the LLM could not respond accurately and only the *parametric_prompt* with golden answers for all other questions, our training set D_{Adapt} is ready for fine-tuning. The fine-tuning process entails training the base LLM on our dataset, resulting in the ADAPT-LLM model.

3.2 Inference

During inference, we utilize the fine-tuned model to generate responses to unseen questions. We employ the same prompts used during the training phase, as outlined in Section 3.1. Initially, the model is prompted to either provide a direct response or return $\langle$RET$\rangle$ if it is unsure of the answer. If the model returns $\langle$RET$\rangle$, we proceed with information retrieval to acquire relevant context by means of an off-the-shelf IR system. Subsequently, we augment the question with the retrieved context and prompt the model again using the second type of prompt introduced during the training phase. An example of this process is provided in Appendix C.

4 Experiments and Results

In this section, we outline the experimental framework aimed at assessing the performance of the proposed adaptive retrieval approach, ADAPT-LLM. We begin by describing the datasets utilized (Section 4.1), followed by an overview of our base model (Section 4.2), the different configurations of the base model (Section 4.3), and the training details (Section 4.4). Subsequently, we introduce the three primary experiments: evaluation of ADAPT-LLM performance compared to 2 baseline models (Section 4.5); analysis ADAPT-LLM's ability to determine when extra context is necessary to answer a question (Section 4.6); comparison with the state-of-the-art approach for PopQA (Section 4.7).

4.1 Datasets

Below are brief descriptions of the datasets we used for training and evaluation of our models, ensuring no overlap between train and test splits across all datasets:

NQ The Natural Questions dataset (Kwiatkowski et al., 2019) is a collection of real-world questions derived from Google search queries, accompanied by long-form text passages obtained from Wikipedia articles and providing a diverse range of topics and natural language variations. We utilize this dataset for **training** our models in the experiments.

SQuAD The Stanford Question Answering Dataset SQuAD (Rajpurkar et al., 2016) is a widely utilized dataset in the field of natural language processing and comprises questions posed by crowdworkers on a diverse range of Wikipedia articles, along with relevant paragraph passages serving as context. We utilize this dataset for **training** our models in the experiments.

PopQA The Popular Questions and Answers dataset (Mallen et al., 2023) consists of curated questions sourced from various online platforms, encompassing a wide range of domains and styles. Given the variability in the effectiveness of context retrieval strategies observed in this dataset, we select PopQA as our test set to **evaluate** the language models' performance in determining when context is necessary for accurate answer provision.

4.2 Base Models

In our experiments, we employ the open-source instruction-based LLMs Llama-2 (7 billion parameters) (Touvron et al., 2023a) and Llama-3.1 (8 billion parameters) (Dubey et al., 2024). These models are pretrained on a comprehensive corpus derived from publicly available online data sources, showcasing superior performance across 150 diverse NLP tasks (Vavekanand and Sam, 2024). Llama-3.1, in particular, introduces an extended

Training Set	Model configuration	Accuracy	
		Llama-2	Llama-3.1
NQ	NEVER RETRIEVE	21.43%	27.86%
	ALWAYS RETRIEVE	35.86%	37.98%
	ADAPT-LLM (ours)	**36.77%**	**38.88%**
SQuAD	NEVER RETRIEVE	21.22%	27.99%
	ALWAYS RETRIEVE	36.59%	38.64%
	ADAPT-LLM (ours)	**38.15%**	**40.25%**

Table 1: Performance comparison of Llama-2 and Llama-3.1 models trained on the NQ and SQuAD datasets using different retrieval configurations (NR-LLM, AR-LLM, and ADAPT-LLM), evaluated on the PopQA test set.

context length, which effectively doubles its ability to process and understand longer sequences of text. These advancements contribute significantly to the model's enhanced performance and capabilities in various natural language understanding tasks.

4.3 Model Configurations

We conduct the experiments using three different model configurations, corresponding to the three different ways in which an LLM and an IR system can be combined:

Adaptive Retrieval (ADAPT-LLM). The ADAPT-LLM model dynamically decides whether to retrieve context based on the question and its perceived need for contextual information, as explained in Section 3.1. As the IR system, we use Contriever (Gautier et al., 2022), which is an unsupervised model pretrained on a large corpus, followed by fine-tuning on MS MARCO (Nguyen et al., 2016). We only retrieve the most relevant passage according to the IR system to prompt the base LLM for the final answer.

Never-Retrieve (NR-LLM). This model configuration is trained to answer questions solely based on the question text without considering any contextual information. It serves as the baseline for evaluating the performance of question answering models in the absence of context.

Always-Retrieve (AR-LLM). In contrast to the NR-LLM model, this configuration always retrieves context passages to assist in answering questions. It is trained to utilize context consistently for generating answers. To ensure a fair comparison with ADAPT-LLM, we also use Contriever (Gautier et al., 2022) as the IR system and only retrieve the most relevant passage as context.

4.4 Training Details

For all three model configurations (ADAPT-LLM, AR-LLM and NR-LLM) and both training sets (SQuAD and NQ), we adhere to the parameter configuration established in Alpaca-Lora (Taori et al., 2023) which includes a batch size of 128, three epochs, and a fixed learning rate of 3e-4. We incorporated LoRA (Low-Rank Adaptation) regularization, with parameters configured for r=8, alpha=16, and a dropout rate of 0.05. Training was performed on an NVIDIA A40 GPU, for an average training time of approximately 8 hours. We do not perform any model selection and we use the last checkpoint after 3 epochs of training.

4.5 Validating the Adaptive Retrieval Approach

In order to assess the effectiveness of our adaptive approach (ADAPT-LLM) compared to NR-LLM and AR-LLM configurations, we fine-tuned the Llama-2 and Llama-3.1 models on the NQ and SQuAD datasets. Training samples for NR-LLM and AR-LLM were created using question-answer pairs from these datasets, with NR-LLM answering without context and AR-LLM using both question and context. For ADAPT-LLM, we followed the approach in Section 3.1, generating a dataset with responses indicating whether context was needed or not. The trained models were then tested on the PopQA dataset to evaluate their performance in a real-world question answering scenario. During inference, NR-LLM and AR-LLM models were utilized as is, with corresponding instruction prompts provided, and outputs expected to be answers to the questions. Conversely, for the ADAPT-LLM model, we followed the same prompt procedure as explained in Section 3.2.

The generated answers are compared to the set

Training	⟨RET⟩ Usage	⟨RET⟩		No ⟨RET⟩	
		Acc. w/ context	Acc. w/o context	Acc. w/ context	Acc. w/o context
NQ	86.86%	33.89%	20.34%	65.03%	77.61%
SQuAD	83.65%	34.26%	14.32%	67.24%	78.04%

Table 2: Results of the usage of the ⟨RET⟩ token in the ADAPT-LLM model. The first column shows the percentage of PopQA questions for which the model requests additional context. The second column focuses on the questions for which ADAPT-LLM asks for context (⟨RET⟩), comparing the accuracy between answering those questions with and without context. The last column (No ⟨RET⟩) is for questions which ADAPT-LLM decides to answer directly, comparing the accuracy with and without the context.

of possible answers for each question, as annotated in the PopQA test set. The evaluation metric used is a form of match accuracy, where an answer is considered correct if it matches any of the possible answers in a case-insensitive comparison. Specifically, if a possible answer is found within the generated output, it is deemed correct.

Results shown in Table 1 indicate that ADAPT-LLM consistently outperforms both NR-LLM and AR-LLM on the PopQA test set. As can be observed, NR-LLM exhibits the lowest performance among the models, with a significant 10-15 point accuracy gap compared to the other configurations, underscoring the limitations of relying solely on Llama's parametric memory. Although the difference between AR-LLM and ADAPT-LLM is relatively small, ADAPT-LLM consistently demonstrates a slight but meaningful improvement, with 1% higher accuracy when trained on the NQ datasets and about 1.5% higher accuracy when trained on SQuAD. Overall, these results highlight the effectiveness of the adaptive retrieval approach, which dynamically determines when context is necessary for accurate question answering, leading to improved performance compared to fixed strategies of always or never retrieving context.

Given the close performance between Llama-2 and Llama-3.1, with a slight advantage for the latter, we opted to use only Llama-3.1 for the subsequent experiments.

4.6 Contextual Retrieval Decision Analysis

In this experiment, our objective is to once again evaluate the effectiveness of the ADAPT-LLM model, this time focusing on its ability to accurately determine when additional context is needed. For this purpose, we adhere to the following steps:

1. We conduct inference on the ADAPT-LLM model using the PopQA test set, prompting it to either return an answer directly or indicate the need for additional context by returning ⟨RET⟩.

2. In the case of receiving a ⟨RET⟩ response from the ADAPT-LLM model, we proceed with the following steps:

   2.1. We conduct inference on the ADAPT-LLM model, prompting it to return an answer given the context obtained from the IR system.

   2.2. We also conduct inference on the NR-LLM model with the instruction to provide an answer directly without additional context.

3. If the ADAPT-LLM model decides to answer the question directly relying only on its parametric memory:

   3.1. We conduct inference on the ADAPT-LLM model, prompting it to return the answer without providing context.

   3.2. We conduct inference on the AR-LLM model with the instruction to provide an answer using the context retrieved by the IR system.

Table 2 presents the results of this experiment. The first thing to note is that the ADAPT-LLM model generates the ⟨RET⟩ token for approximately 83-87% of the questions in the PopQA dataset, aligning with the low performance of the NR-LLM configuration demonstrated in Table 1.

However, ADAPT-LLM consistently determines when additional context is required to answer a question accurately. Across both the NQ and SQuAD training datasets, ADAPT-LLM exhibits significantly higher accuracy when retrieving context compared to the NR-LLM model's accuracy without context (as indicated in the ⟨RET⟩ column of Table 2). Specifically, for the NQ dataset, the accuracy of the ADAPT-LLM model when requesting

Figure 2: Histograms depicting the proportion of questions where ADAPT-LLM trained on NQ (left) and ADAPT-LLM trained on SQuAD (right) ask for extra context for different popularity score intervals.

Passages	SQuAD Dev Acc.	NQ Dev Acc.
Gold	**89.85%**	**70.91%**
Contriever	23.84%	28.52%

Table 3: Performance comparison of ADAPT-LLM for the SQuAD and NQ dev sets, when using the gold passages provided by the datasets and when using the best passage retrieved by Contriever.

context is 33.89%, whereas the accuracy of the NR-LLM model without context retrieval is notably lower at 20.34%. Similarly, for the SQuAD dataset, ADAPT-LLM achieves an accuracy of 34.26% with context retrieval, whereas the NR-LLM model's accuracy without context is substantially lower at 14.32%. Finally, the last column of Table 2 (No ⟨RET⟩) shows the performance of ADAPT-LLM when answering questions based solely on its parametric memory. As can be seen, accuracies above 77% are obtained when no context is utilized, providing further evidence that ADAPT-LLM effectively discerns between retrieving context and providing direct answers to questions. Additionally, we evaluate the performance of these questions when context is added to the input, revealing significant decreases in accuracy of up to 12 absolute points. These findings provide insights into the effectiveness of the decision-making process employed by the ADAPT-LLM model in determining the necessity of additional context for accurate response generation and present empirical evidence of the necessity of performing dynamic context retrieval in improving the accuracy of question answering models. However, it is notable that the overall performance of the model when answering questions with retrieved context, as observed in Table 2 (approximately 34%), is relatively low. To further explore this observation, we conduct an additional experiment: evaluating ADAPT-LLM on the NQ and SQuAD development splits, comparing performance when using the gold passages of the dataset and the context retrieved by our IR system, Contriever (Gautier et al., 2022). Unfortunately, PopQA does not provide the gold passages, so direct evaluation there was not possible.

Table 3 presents the results of this experiment. A significant performance difference is observed between using the gold passage and the top passage retrieved by Contriever for both datasets (approximately 66 absolute points for SQuAD and 42 for NQ). This indicates that Contriever, and current IR systems in general, do not consistently retrieve the most relevant passage to answer a given question. This observation underscores the importance of retrieving multiple documents as context, as seen in the most successful open-domain QA systems (Izacard and Grave, 2021), and highlights its impact on the overall performance of ADAPT-LLM in PopQA. To further validate the behavior of ADAPT-LLM when requesting additional context, Figure 2 illustrates the proportion of questions for which our model generates the ⟨RET⟩ token, aggregated by popularity score intervals (left image for ADAPT-LLM trained on NQ and right image for SQuAD). Mallen et al. (2023) suggest that high-popularity questions can be adequately answered using the parametric memory of the LLM, while lower popularity scores necessitate extra context. In the figure, we observe this pattern for both versions of ADAPT-LLM, indicating that our model, despite lacking access to popularity scores during training or inference, has learned effective criteria for requesting additional context.

Additionally, we have observed that different types of questions yield significantly different results in model performance (see Appendix D).

4.7 Comparison with State-of-the-Art Methods

We conducted a comparative analysis between our ADAPT-LLM model and the current state-of-the-art approach for PopQA proposed by Mallen et al. (2023). Their methodology relies on the popularity score annotated in the PopQA dataset to determine whether a question requires additional context. To establish the optimal threshold for determining question popularity, Mallen et al. (2023) split the PopQA dataset into 75% as a development set for threshold determination and 25% as a test set. In the original paper, they apply this methodology to various LLMs available at that moment.

To ensure a fair comparison between ADAPT-LLM and the popularity-based method, we replicated their approach using the Llama-3.1 8B model to determine the best popularity score threshold (found to be 710,000) using the same PopQA development set. This allowed us to obtain results consistent with their methodology while utilizing our base LLM. Similar to the original results in Mallen et al. (2023) when using smaller models, the popularity score threshold is almost equivalent to always retrieving contextual information for Llama-3.1 8B. The IR usage is of 99.86% as presented in Table 4. This clearly shows how the popularity score method struggles with smaller size models, being GPT-3 DAVINCI-003 the only model to get a IR usage below 80% in the original paper when using adaptive retrieval with the Contriever. Subsequently, we evaluated our ADAPT-LLM configuration on the same 25% test set split and compared the outcomes with those obtained using the method described by Mallen et al. (2023). This systematic comparison enabled us to assess the efficacy of our ADAPT-LLM model in relation to the current state of the art. The results of this experiment are presented in Table 4. We observe comparable performance between the replicated approach of Mallen et al. (2023) and ADAPT-LLM when trained on NQ and SQuAD datasets and tested on the 25% subset of PopQA. It's worth mentioning that ADAPT-LLM does not utilize any information from PopQA, unlike Mallen et al. (2023), who directly use the popularity score and a 75% portion of PopQA dataset to find an optimal value for that popularity score. This methodology is not generalizable to other open-domain question answering tasks since the popularity score is a unique feature of PopQA. However, ADAPT-LLM can be applied to any similar dataset. Given these characteristics, we believe that the results obtained by ADAPT-LLM are even more significant, offering comparable performance to an approach that utilizes dataset-specific information.

Model Configuration	IR usage	Accuracy
POPULARITY SCORE	99.86%	37.23%
ADAPT-LLM (NQ)	82.93%	36.08%
ADAPT-LLM (SQUAD)	80.15%	**37.92%**

Table 4: Performance comparison of Llama-3.1 base models trained on the SQuAD and NQ datasets for the ADAPT-LLM and POPULARITY SCORE configurations.

5 Conclusions

In this paper, we introduce ADAPT-LLM, a LLM which learns to discern when additional context is necessary for answering a question, rather than relying solely on its parametric memory. ADAPT-LLM is the result of fine-tuning a base LLM on an open-domain question answering dataset that has been modified to differentiate between questions answerable with the LLM's parametric memory alone and those requiring supplementary context. To construct these training datasets, we initially subject the base LLM to zero-shot evaluation to determine its accuracy in answering questions.

For questions where the model's response is incorrect, we train the LLM to generate a special token, ⟨RET⟩, indicating the need for additional context. Through extensive experiments conducted on the PopQA dataset, we show that ADAPT-LLM performs better than its two fixed alternatives: never retrieving and always retrieving relevant context information. Furthermore, our findings highlight ADAPT-LLM's capability to effectively discern the necessity of additional context, which is the primary objective of this work.

For future investigations, we propose exploring methods to enhance performance when utilizing an IR system, such as incorporating learnable sequential retrieval techniques. Furthermore, we believe it would be valuable to conduct a more in-depth analysis of the interaction between training and testing datasets in the development of ADAPT-LLM systems.

6 Limitations

In this work, we introduce a method to enhance LLMs with retrieval capabilities. The use of a retriever reduces the hallucination rate in ADAPT-LLM by providing relevant external information when necessary. However, when the model opts to generate answers without retrieval, there remains a risk of producing factually incorrect or ungrounded responses.

Our results show that training an LLM to learn when to retrieve context improves performance on general domain datasets such as NQ. While these datasets cover a broad range of topics, they may not fully capture the complexities of real-world scenarios, particularly in specialized domains. Evaluating ADAPT-LLM's generalization across diverse and domain-specific contexts is beyond the scope of this work, and future research should explore the model's adaptability to various domains to ensure robustness in practical applications.

Additionally, our analysis focused on a limited number of models, selected for their open-source nature and strong performance, making them particularly valuable to the scientific community. Expanding this analysis to include a broader range of models could provide further insights into the generalizability and limitations of our approach.

7 Ethical Considerations

ADAPT-LLM aims to reduce the number of factually incorrect answers by retrieving contextual information when the model predicts that additional context is needed. While retrieving from trusted sources has been shown to enhance the factuality of LLMs (Li et al., 2024), our method sometimes relies on the model's parametric knowledge, which can potentially generate factually incorrect answers.

This could lead to the spread of misinformation, underscoring the importance of implementing robust safeguards, such as confidence scoring and human oversight to mitigate these risks and ensure the responsible deployment of the model.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. 2024. Seven failure points when engineering a retrieval augmented generation system. *arXiv preprint arXiv:2401.05856*.

Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. 2000. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–199.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Amnon Catav, Roy Miara, Ilai Giloh, Nathan Cordeiro, and Amir Ingber. 2024. Rag makes llms better and equal.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Pierre Erbacher, Louis Falissar, Vincent Guigue, and Laure Soulier. 2024. Navigating uncertainty: Optimizing api dependency for hallucination reduction in closed-book question answering. *arXiv preprint arXiv:2401.01780*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 6894–6910. Association for Computational Linguistics (ACL).

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Izacard Gautier, Caron Mathilde, Hosseini Lucas, Riedel Sebastian, Bojanowski Piotr, Joulin Armand, and Grave Edouard. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *EACL 2021-16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 874–880. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Jiarui Li, Ye Yuan, and Zehua Zhang. 2024. Enhancing llm factual accuracy with rag to counter hallucinations: A case study on domain-specific queries in private knowledge-bases. *arXiv preprint arXiv:2403.10446*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252.

Alex Troy Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

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.

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.

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.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 3982–3992.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

Yeon Seonwoo, Juhee Son, Jiho Jin, Sang-Woo Lee, Ji-Hoon Kim, Jung-Woo Ha, and Alice Haeyun Oh. 2022. Two-step question retrieval for open-domain qa. In *60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*, pages 1487–1492. Association for Computational Linguistics.

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 (2023). *URL https://github. com/tatsu-lab/stanford_alpaca*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Raja Vavekanand and Kira Sam. 2024. Llama 3.1: An in-depth analysis of the next-generation large language model.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

A Datasets Analysis

Table 5 provides insights into the characteristics of the three datasets involved in our experimental procedure, including the total number of questions and the average number of words per question and answer. While NQ appears to be closer to PopQA in terms of question and answer lengths, the key factor influencing the better results of training ADAPT-LLM on SQuAD may be the number of questions in the training dataset (~87K in SQuAD and ~58K in NQ). Further analyses are required to elucidate the factors that render a training dataset more suitable for a given target dataset (which is beyond the scope of our study), but these results suggest that scale may play once again a crucial role.

	NQ	**SQuAD**	**PopQA**
Questions	58,880	87,599	14,282
Words/question	9.20	10.06	6.62
Words/answer	2.26	3.16	2.04

Table 5: Comparison of the three datasets we use for our experiments, i.e. SQuAD, NQ and PopQA. For each of them we provide the number of questions, and the average number of words per question and answer.

B Training Data Algorithm

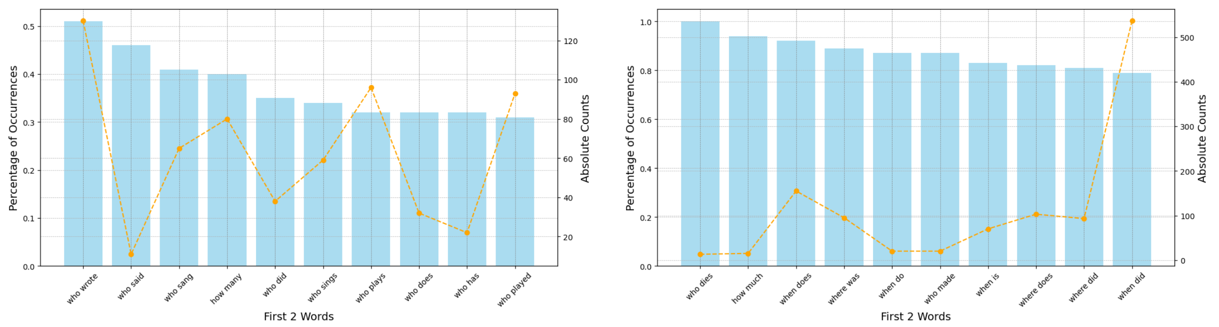
The following algorithm outlines the process used to generate the training data, as detailed thoroughly in Section 3.1.

Input: Q: questions, A: answers, P: passages, LLM
Output: DS_{Adapt}: A training dataset for Adaptive Retrieval

```
DS_Adapt = init_empty()
for q, gold_ans, pass in (Q, A, P) do
    ans = LLM(q)
    if ans = gold_ans then
        inst =
          build_instance('parametric_prompt',
          q, gold_ans)
        DS_Adapt.add(inst)
    end
    else
        inst1 =
          build_instance('parametric_prompt',
          q, "<RET>")
        DS_Adapt.add(inst1)
        inst2 =
          build_instance('context_prompt',
          q, gold_ans, pass)
        DS_Adapt.add(inst2)
    end
end
return DS_Adapt
```

C Example Prompts

In the following examples, we illustrate the process used to interact with ADAPT-LLM for question answering tasks. Initially, the model is prompted to answer a question or return ⟨RET⟩ if it is uncertain about the correct answer. In the first example, the model returns ⟨RET⟩, indicating that it requires additional context. Then, a second prompt is sent



(a) Most Accurate Questions

(b) Most Inaccurate Questions

Figure 3: Analysis of the first two words of questions and their correlation with model accuracy.

to provide the necessary context, and the model successfully answers the question. This prompt schema has been used throughout all our experiments when doing inference on ADAPT-LLM.

C.1 Prompt 1

Listing 1: Prompt 1

```
Below is an instruction that describes a
     task, paired with an input that
    provides further context. Write a
    response that appropriately
    completes the request.

### Instruction:
answer the question Q. If you need help
    answer <RET> to get the context

### Input:
Q: In what city was Aarno Maliniemi born
    ?

### Response:
```

Model Output:

Listing 2: Model Output 1

```
<RET>
```

C.2 Prompt 2

Listing 3: Prompt 2

```
Below is an instruction that describes a
     task, paired with an input that
    provides further context. Write a
    response that appropriately
    completes the request.

### Instruction:
answer the question Q given the context
    C

### Input:
Q: In what city was Aarno Maliniemi born
    ?
C: Aarno Raninen Aarno Raninen was a
    Finnish singer, songwriter and
    musician. His main instrument was
    piano but he has also mastered
    violin, cello and accordion. Born in
     Kotka, Raninen began his musical
    studies at a young age. After he
    moved to Helsinki in 1966 he got a
    job as a studio conductor at
    Musiikki-Fazer. While working there
    he made a lot of cooperative work
    with the likes of songwriter Juha
    Vainio. Later on Raninen went to
    work in Discophon where he wrote
    lyrics for many Finnish musicians,
    such as Seija Simola, Carola, Tauno

### Response:
```

Model Output:

Listing 4: Model Output 2

```
Helsinki. Aarno Maliniemi was born in
    Helsinki.
```

D Analysis of Question Accuracy

In this appendix, we present an analysis of the first two words of questions and their correlation with the model's performance. For both correctly and incorrectly answered questions, we identified the top 10 word pairs that appear most frequently. These absolute counts were then normalized by dividing them by the total occurrences of each word pair, resulting in the percentage of occurrences in the dataset. The instances with less then 10 occurrences were filtered out. This analysis was conducted using our ADAPT-LLM model, trained on the NQ train set and tested on the NQ dev set, as NQ offers better variability and representativeness in question types. The two figures below[2] illustrate these top 10 word pairs for accurate (Figure 3a) and inaccurate (Figure 3b) questions, ranked

[2]The Matplotlib library has been used to create the charts.

by their normalized values (shown as blue bars), with the absolute counts also depicted (represented by the yellow line). From this analysis, we can observe distinct patterns in the types of questions that correlate with correct versus incorrect answers. Correctly answered questions often seek specific information; for instance, 9 times out of 10, they ask for the name of one particular person. In contrast, incorrectly answered questions tend to be more vague; 7 times out of 10 they begin with "when" (which could be answered with a specific year, month, day, or a broad period of time) or "where" (which could be answered with a specific city or country), leading to less precise answers.