When to Retrieve: Teaching LLMs to Utilize Information Retrieval Effectively

Tiziano Labruna
University of Bozen-Bolzano
Fondazione Bruno Kessler
tlabruna@fbk.eu

Jon Ander Campos
Cohere
jonander@cohere.com

Gorka Azkune
HiTZ Center - Ixa
UPV/EHU
gorka.azcune@ehu.eus

Abstract

In this paper, we demonstrate how Large Language Models (LLMs) can effectively learn to use an off-the-shelf information retrieval (IR) system specifically when additional context is required to answer a given question. Given the performance of IR systems, the optimal strategy for question answering does not always entail external information retrieval; rather, it often involves leveraging the parametric memory of the LLM itself. Prior research has identified this phenomenon in the PopQA dataset, wherein the most popular questions are effectively addressed using the LLM's parametric memory, while less popular ones require IR system usage. Following this, we propose a tailored training approach for LLMs, leveraging existing open-domain question answering datasets. Here, LLMs are trained 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 the questions, (ii) using always the parametric memory of the LLM, and (iii) using a popularity threshold to decide when to use a retriever. Through our analysis, we demonstrate that ADAPT-LLM is able to generate the ⟨RET⟩ token when it determines that it does not know how to answer a question, indicating the need for IR, while it achieves notably high accuracy levels when it chooses to rely only on its parametric memory.

1 Introduction

The task of question answering (QA) remains a key focus in Natural Language Understanding research. Benchmarks such as Natural Questions (NQ) (Kwiatkowski et al., 2019), SQuAD (Rajpurkar et al., 2016), and QuAC (Choi et al., 2018) are commonly used to evaluate QA models. Large Language Models (LLMs) now consistently outperform traditional methods on these datasets.

Two primary approaches are typically used for QA with LLMs:

(i) **Closed Book Question Answering**: The model relies solely on its parametric memory, enhanced through instruction tuning (Taori et al., 2023) or few-shot prompting (Brown et al., 2020). However, parametric memory is limited to the training data and may be outdated, missing post-training information.

(ii) **Open Book Question Answering**: An LLM is paired with an Information Retriever (IR) system (Izacard and Grave, 2021; Zhu et al., 2021) to access external context for more accurate answers.

Recent work by Mallen et al. (2023) highlights the complexity of choosing between these strategies. Using the PopQA dataset—14K questions with popularity scores—they show that LLMs perform well on popular questions using only parametric memory, while IR helps with less popular ones. Their findings support a hybrid approach: rely on parametric memory for high-popularity questions and use IR for the rest, guided by a fixed popularity threshold. However, most QA datasets lack popularity scores, making this strategy non-generalizable.

Our study addresses whether LLMs can learn to decide on their own when to invoke an IR system. To investigate this, we analyze LLM performance on an open-domain QA dataset, identifying which questions it answers correctly and which it does not. For incorrect responses, we annotate questions with a special ⟨RET⟩ token to indicate the need for additional context. Using this, we build a new training dataset where the LLM learns either to answer directly or to retrieve context when uncertain (see Figure 1). We refer to this model as ADAPT-LLM. We evaluate ADAPT-LLM on PopQA, a strong testbed for hybrid retrieval strategies. Our results

Proceedings of Recent Advances in Natural Language Processing,pages 623–632
Varna, Sep 8–10, 2025
https://doi.org/10.26615/978-954-452-098-4-073

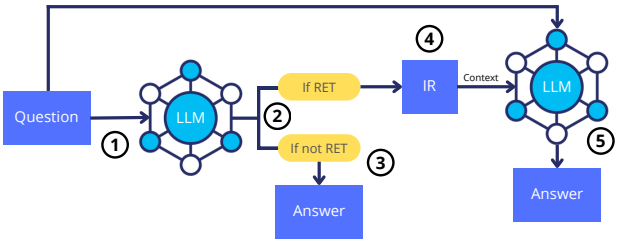

Figure 1: The inference process of ADAPT-LLM step-by-step: given a question (step 1), an LLM decides (step 2) whether to answer the question directly (step 3) or to ask for additional contextual information, generating the special ⟨RET⟩ token; for the latter, an off-the-shelf IR system is used to retrieve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

show:

- ADAPT-LLM consistently outperforms fixed strategies, such as always retrieving context or never retrieving.
- It performs comparably to methods that use popularity scores—without relying on any dataset-specific metric.
- When ADAPT-LLM chooses to retrieve context, accuracy improves significantly; when it answers directly, it also achieves high accuracy. This indicates effective judgment.
- The main performance bottleneck lies in the IR system: accuracy with gold passages is much higher than with retrieved ones.

These results highlight the value of adaptive retrieval for QA with LLMs. By training ADAPT-LLM to decide when external information is necessary, we show it is feasible to teach LLMs to use retrieval selectively and effectively.

2 Related Work

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) has improved many NLP tasks like question answering (Karpukhin et al., 2020; Izacard and Grave, 2021), truthfulness (Ji et al., 2023; Lin et al., 2022), and language modeling (Guu et al., 2020; Borgeaud et al., 2022). By grounding generation in retrieved text, smaller models can match larger ones (Catav et al., 2024), and RAG helps keep LLMs updated without costly retraining (Gao et al., 2023). However, traditional retrieval (e.g., TF-IDF, BM-25 (Robertson et al., 2009)) relies on keyword overlap and struggles with lexical gaps (Berger et al., 2000). Transformer-based dense retrieval models (Gao et al., 2021; Reimers and Gurevych, 2019; Karpukhin et al., 2020) improve performance but face challenges in zero-shot new domains (Thakur et al., 2021). Retrieval quality limits overall model performance and large indices increase latency, harming real-time user experience (Barnett et al., 2024). Meanwhile, as models scale, their parametric knowledge grows (Kaplan et al., 2020), enabling competitive open-domain QA using only internal knowledge (Liang et al., 2023; Achiam et al., 2023; Touvron et al., 2023). This motivates adaptive approaches (Schick et al., 2024; Mallen et al., 2023): use the model's parametric knowledge when possible, and augment with retrieval only when needed. For example, Schick et al. (2024)'s Toolformer learns to use external tools via self-supervised API calls, boosting performance but often overusing tools (e.g., search engine used 99.3% in QA). In contrast, ADAPT-LLM leverages parametric knowledge and reduces retrieval use to 83.99% while improving over vanilla retrieval. Similarly, Mallen et al. (2023) propose PopQA, a dataset with entity popularity scores to decide when to retrieve: below a threshold, retrieve; above it, answer directly. This outperforms vanilla retrieval but depends on unavailable popularity scores in real scenarios. Contemporaneous to our work, Erbacher et al. (2024) train an LLM to balance hallucination risk and retrieval cost. Our ADAPT-LLM similarly learns when to retrieve, but also compares against baselines of always or never retrieving, showing the benefit of adaptive retrieval. Finally, Roy et al. (2024) propose SELF-multi-RAG, which learns when to retrieve, how to rewrite conversational context for retrieval, and how to assess passage relevance in multi-turn QA, complementing our ADAPT-LLM approach that focuses on balancing retrieval and

parametric knowledge without rewriting.[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 ⟨RET⟩, 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).

The decision-making process of ADAPT-LLM enables the model to determine the necessity of context for answering questions through dynamic assessment of each prompt. This flexible behavior allows the model to strike a balance between utilizing context for enhanced understanding and delivering direct answers when sufficient.

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

Algorithm 1: Training data creation

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

```
1  DS_Adapt = init_empty()
2  for q, gold_ans, pass in (Q, A, P) do
3      ans = LLM(q)
4      if ans = gold_ans then
5          inst = build_instance('parametric_prompt', q, gold_ans)
6          DS_Adapt.add(inst)
7      end
8      else
9          inst1 = build_instance('parametric_prompt', q, "¡RET¿")
10         DS_Adapt.add(inst1)
11         inst2 = build_instance('context_prompt', q, gold_ans, pass)
12         DS_Adapt.add(inst2)
13     end
14 end
15 return DS_Adapt
```

3.1 Training ADAPT-LLM

Here, we delineate the methodology employed to train our ADAPT-LLM model. The process of crafting the training data, denoted as DS_{Adapt}, is presented in Algorithm 1.

We begin by selecting an open-domain question answering dataset containing questions Q, associated context passages P, and corresponding answers A. We initialize DS_{Adapt} to an empty set (line 1 of the algorithm). For each question in Q, we leverage the base LLM without any retrieval mechanism to perform a zero-shot inference (line 3). This step allows us to differentiate questions for which the model generates correct answers from those where its responses are inaccurate. This process can be understood as a way to discover what the base LLM *knows* due to its parametric memory. For questions where the model's response is accurate (line 4), we build a training set instance incorporating the following prompt, which we call *parametric_prompt*:

```
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 (line 5), which is subsequently appended to the DS_{Adapt} dataset (line 6). This prompt is always used without context and serves to teach the model to either answer directly or to signal that it needs context. In contrast, if the LLM fails to produce a correct response to the question (line 8), we build two different instances. The first employs the same *parametric_prompt* as previously described, with ⟨RET⟩ designated as the answer (line 9), indicating the necessity for additional context. The second prompt, termed *context_prompt*, encompasses 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 (line 11). This two-stage supervision helps the model decide when it needs external knowledge and how to use it effectively once retrieved.

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 prepared for the subsequent fine-tuning phase. The fine-tuning process entails training the base LLM on our dataset, resulting in the ADAPT-LLM model.

This approach ensures that the model effectively learns to discern when context is necessary for answering questions, or to provide a direct response when it suffices, as well as answer directly when provided with context.

3.2 Inference

In the inference phase, 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 ⟨RET⟩ if it is unsure of the answer.

If the model returns ⟨RET⟩, 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.

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:

1. Evaluation of ADAPT-LLM performance compared to the following baseline models: (i) an LLM that retrieves contextual information for all questions, and (ii) an LLM that exclusively relies on its parametric memory without using an IR system for any question (Section 4.5).

2. Analysis of ADAPT-LLM's ability to determine when extra context is necessary to answer a question (Section 4.6).

3. Comparison with the state-of-the-art approach for PopQA (Section 4.7).

4.1 Datasets

To ensure comprehensive training and evaluation of our models, we specifically selected three diverse question answering datasets. For training, we chose NQ (Kwiatkowski et al., 2019) and SQuAD (Rajpurkar et al., 2016), as they are widely recognized datasets that assess factual knowledge and are based on Wikipedia. For evaluation, we opted for PopQA (Mallen et al., 2023). Below are brief descriptions of each dataset:

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

Training Set	Model configuration	Accuracy
NQ	NEVER RETRIEVE	21.43%
	ALWAYS RETRIEVE	35.86%
	ADAPT-LLM (ours)	**36.77%**
SQUAD	NEVER RETRIEVE	21.22%
	ALWAYS RETRIEVE	36.59%
	ADAPT-LLM (ours)	**38.15%**

Table 1: Performance comparison of Llama-2 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. Exact match accuracy is reported for all models.

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 Model

In our experiments, we employ Llama-2 (Touvron et al., 2023) as our base LLM. Llama-2 is an open-source instruction-based LLM, which comes in versions of 7B, 13B, and 70B parameters. The model is pretrained on an expanded corpus sourced from publicly available online data sources. This corpus offers a 40% increase in size compared to its predecessor, contributing to the model's enhanced performance and capabilities.

Additionally, Llama-2 features an extended context length, effectively doubling its capacity to process and comprehend longer sequences of text. These enhancements significantly improve the model's effectiveness across various natural language understanding tasks. Specifically, for our experiments, we utilize the Llama-2 model with 7B parameters, leveraging its robust capabilities for our specific research objectives.

4.3 Model Configurations

We conduct the experiments using three different model configurations, corresponding to the three

	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 2: 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.

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 (Izacard 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 (Izacard 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,

Training	⟨RET⟩ Usage	⟨RET⟩		No ⟨RET⟩	
		Acc. w/ context	Acc. w/o context	Acc. w/ context	Acc. w/o context
NQ	87.26%	33.04%	14.65%	55.72%	62.36%
SQuAD	83.93%	33.40%	9.94%	57.73%	62.92%

Table 3: 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 performance between answering those questions with and without context. The last column (No ⟨RET⟩) is for questions which ADAPT-LLM decides to answer directly. We also compare the performance with and without the context retrieved by the IR system.

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

To assess the effectiveness of our adaptive approach (ADAPT-LLM) compared to the NR-LLM and AR-LLM baselines, we fine-tuned the Llama-2 model on both the NQ and SQuAD datasets using all three configurations. For NR-LLM and AR-LLM, we constructed training samples using question-answer pairs with instruction prompts: NR-LLM was prompted to answer without context, while AR-LLM received both question and context. In contrast, ADAPT-LLM was trained using the two-step approach described in Section 3.1, resulting in 74.72% of questions marked with the ⟨RET⟩ token for NQ and 87.49% for SQuAD. All models were evaluated on the PopQA dataset. During inference, NR-LLM and AR-LLM followed their respective prompting strategies, while ADAPT-LLM used the procedure from Section 3.2. We used **Exact Match Accuracy** as the evaluation metric, comparing generated answers to the annotated gold answers in PopQA. Table 1 shows that across both training sets, ADAPT-LLM consistently outperforms NR-LLM and AR-LLM. NR-LLM yields the lowest scores, with an accuracy gap of around 14 points, confirming that Llama-2's parametric memory alone is insufficient for PopQA.

The performance gap between AR-LLM and ADAPT-LLM is smaller. ADAPT-LLM achieves 36.77% and 38.15% accuracy on PopQA when trained on NQ and SQuAD, respectively, compared to 35.86% and 36.59% for AR-LLM. The best results are obtained with SQuAD.

Although both NQ and SQuAD are Wikipedia-based like PopQA, we examine which training set better aligns with the evaluation set. Table 2 compares their characteristics.

NQ is closer in question and answer length to PopQA, but SQuAD's larger size (∼87K vs. ∼58K questions) may explain its better results. While further analysis is needed to fully understand dataset suitability—beyond the scope of this work—these findings suggest that scale plays a key role.

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. We follow these steps:

1. Run inference on ADAPT-LLM over the PopQA test set, prompting it to either return an answer directly or output ⟨RET⟩ if more context is needed.
2. If ⟨RET⟩ is returned:
   - 2.1. Provide the retrieved context to ADAPT-LLM and collect its final answer.
   - 2.2. Run the same question on NR-LLM without context.
3. If ADAPT-LLM answers directly:
   - 3.1. Run ADAPT-LLM without context.
   - 3.2. Run AR-LLM with context retrieved by the IR system.

Table 3 presents the results of this experiment. The first thing to note is that the ADAPT-LLM model generates the ⟨RET⟩ token for approximately 82-83% of the questions in the PopQA dataset, with similar ratios observed across both training datasets. This observation aligns 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.

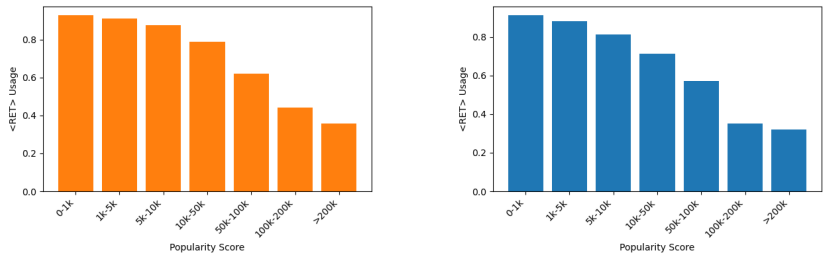

Figure 2: Histograms depicting the proportion of questions where ADAPT-LLM trained on NQ (left) and ADAPT-LLM trained on SQuAD (right) ask for extra context for different popularity score intervals.

Passage Type	SQuAD Acc. (%)	NQ Acc. (%)
Gold	**89.42**	**69.76**
Contriever	22.49	27.04

Table 4: Accuracy of ADAPT-LLM on SQuAD and NQ dev sets using gold vs. Contriever-retrieved passages.

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 3). Specifically, for the NQ dataset, the accuracy of the ADAPT-LLM model when requesting context is 33.04%, whereas the accuracy of the NR-LLM model without context retrieval is notably lower at 14.65%. Similarly, for the SQuAD dataset, ADAPT-LLM achieves an accuracy of 33.40% with context retrieval, whereas the NR-LLM model's accuracy without context is substantially lower at 9.94%.

Finally, the last column of Table 3 (No ⟨RET⟩) shows the performance of ADAPT-LLM when answering questions based solely on its parametric memory. As can be seen, accuracies above 62% 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 7 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 3 (approximately 33%), is relatively low.

To further explore this observation, we conduct an additional experiment: evaluating ADAPT-LLM (both versions trained on NQ and SQuAD) 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 (Izacard et al., 2022). Unfortunately, PopQA does not provide the gold passages, so direct evaluation there was not possible.

Table 4 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 67 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 Figure 2, 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.

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 (Llama-2 was not released yet). To ensure a fair comparison between ADAPT-LLM and the popularity-based method, we replicated their approach using the Llama-2 7B model to determine the best popularity score threshold (found to be 707,000) 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-2 7B. The IR usage is of 99.86% as presented in Table 5. 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 5. 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. These findings substantiate the validity of our approach, demonstrating its effectiveness even when trained on datasets different from the one used for testing.

Model	IR (%)	Acc. (%)
POPULARITY SCORE	99.86	36.81
ADAPT-LLM (NQ)	87.22	35.30
ADAPT-LLM (SQUAD)	83.99	**37.29**

Table 5: Accuracy and IR usage of ADAPT-LLM (trained on NQ and SQuAD) vs. a POPULARITY SCORE-based strategy using Llama-2, following Mallen et al. (2023).

5 Conclusions

In this paper, we introduce ADAPT-LLM, an LLM trained to decide 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. Future work could explore 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.

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 B. M. Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2206–2240. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Amnon Catav, Roy Miara, Ilai Giloh, Nathan Cordeiro, and Amir Ingber. 2024. Rag makes llms better and equal. *Pinecone Blog*. https://www.pinecone.io/blog/rag-study/.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*.

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*.

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

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

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.

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 *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

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. *arXiv preprint arXiv:1611.09268*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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.

Nirmal Roy, Leonardo FR Ribeiro, Rexhina Blloshmi, and Kevin Small. 2024. Learning when to retrieve, what to rewrite, and how to respond in conversational qa. *arXiv preprint arXiv:2409.15515*.

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.

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.

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. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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*.