

# Unified Active Retrieval for Retrieval Augmented Generation

Anonymous ACL submission

## Abstract

In Retrieval-Augmented Generation (RAG), retrieval is not always helpful and applying it to every instruction is sub-optimal. Therefore, determining whether to retrieve is crucial for RAG, which is usually referred to as Active Retrieval. However, existing active retrieval methods face two challenges: 1. They usually rely on a single criterion, which struggles with handling various types of instructions. 2. They depend on specialized and highly differentiated procedures, and thus combining them makes the RAG system more complicated and leads to higher response latency. To address these challenges, we propose Unified Active Retrieval (UAR). UAR contains four orthogonal criteria and casts them into plug-and-play classification tasks, which achieves multifaceted retrieval timing judgements with negligible extra inference cost. We further introduce the Unified Active Retrieval Criteria (UAR-Criteria), designed to process diverse active retrieval scenarios through a standardized procedure. Experiments on four representative types of user instructions show that UAR significantly outperforms existing work on the retrieval timing judgement and the performance of downstream tasks, which shows the effectiveness of UAR and its helpfulness to downstream tasks.

## 1 Introduction

With the rapid development of large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023; Zeng et al., 2023; Yang et al., 2023; Cai et al., 2024; Bai et al., 2023), AI assistants based on LLMs become ubiquitous and show remarkable abilities on various types of instructions, e.g., coding, writing and reasoning (OpenAI, 2022; Taori et al., 2023; Chiang et al., 2023; Sun et al., 2024; OpenAI, 2023; Anthropic, 2023; Anil et al., 2023). However, LLMs often generate fabricated and non-factual information (Lin et al., 2022b; Wang et al., 2023a), which is called “hallucination” and makes






Type	Instruction
 The user wants to use retrieval	Could you help me research this question?
	Retrieve information first, and then answer.
 Doesn't require factual knowledge	Write a rap about staying positive.
	Write a few lines of an original poem.
 Facts change over time	Who is the CEO of Google?
	Who is the current Prime Minister of Japan?
 Facts do not change & The model knows	Where is the capital of the United States?
	Who is the author of Harry Potter?
 Facts do not change & Model does not know	What is the name for the lump in a human throat?
	In which country was Michael J. Fox born?

Figure 1: Different types of user instructions, which can not be handled by single active retrieval criteria.

LLMs’ responses not trustworthy in real-world scenarios.

Retrieval-Augmented Generation (RAG) is a prevailing approach to address LLM’s hallucination (Guu et al., 2020; Gao et al., 2024). Given a user query, it usually first retrieves relevant documents and then uses them to augment the LLM’s factual correctness. However, retrieval is not always helpful and applying it to every instruction is sub-optimal. When faced with instructions that do not require external knowledge, RAG can impair the creativity and versatility of LLMs (Asai et al., 2023).

If irrelevant knowledge is retrieved, it will hinder the LLM from utilizing its internal knowledge effectively and make it produce low-quality responses (Shi et al., 2023; Yoran et al., 2023). Meanwhile, compared with only LLM, RAG involves an additional retrieval process and the longer LLM input, resulting in significantly longer response latency. Therefore, applying RAG for all instructions is sub-optimal and unnecessary, and determining the correct timing for retrieval is crucial for LLMs’ real-world application, which is often referred to as *Active Retrieval* (Jiang et al., 2023; Asai et al., 2023).

In general, there are two lines of active retrieval methods. One is the “knowledge-aware” method,

	<b>UAR</b> (our work)	<b>FLARE</b> (Jiang et al., 2023)	<b>Self-RAG</b> (Asai et al., 2023)	<b>SKR</b> (Wang et al., 2023b)
Intent Awareness?	✓	✗	✗	✗
Knowledge Awareness?	✓	✗	✓	✗
Time Awareness?	✓	✗	✗	✗
Self Awareness?	✓	✓	✗	✓

Table 1: Comparison of UAR to other active retrieval methods. Existing methods only consider a single active retrieval criterion, while UAR unifies four orthogonal criteria and can handle various types of user instructions.

based on the instruction’s factual relevance, e.g., Self-RAG (Asai et al., 2023). If the instruction requires factual information, the retrieval will be triggered. Another line of work is the “self-aware” method, based on the LLM’s self awareness (Wang et al., 2023b). The retrieval is only triggered when the LLM thinks that it does not know the answer, i.e., when it is uncertain. In this way, the retrieval can supplement knowledge for the LLM when necessary and avoid unnecessary retrieval cost. Although these methods can determine retrieval timing for specialized scenarios, they still face two challenges: 1. Previous work usually relies on a single criterion, which struggles with diverse scenarios. For instance, the self-aware method (Wang et al., 2023b; Liu et al., 2024; Ding et al., 2024) struggles with various instructions such as time-sensitive queries or those with user’s explicit retrieval intent. For time-sensitive questions, it is challenging for a static LLM to judge whether it possesses the correct knowledge for a rapidly changing answer. Additionally, these methods often overlook user’s intent in real-world scenarios, such as when a user seeks a verifiable answer that requires external information sources, necessitating retrieval. Therefore, correctly determining whether to retrieve requires multifaceted decision-making. 2. Existing methods rely on specialized procedures, complicating the integration within the RAG system and increasing computational load. For example, FLARE (Jiang et al., 2023) uses the confidence of generation and Rowen (Ding et al., 2024) relies on response divergence for the same question. These highly differentiated approaches are difficult to unify, making it very difficult to extend them to new scenarios.

To address these challenges, we propose **Unified Active Retrieval (UAR)**, a unified and comprehensive framework for judging whether to retrieve for various types of user instructions. UAR consists of various orthogonal criteria of retrieval timing and casts them into unified classification tasks, and thus can judge the LLM’s retrieval timing both

comprehensively and efficiently. Specifically, UAR consists of four orthogonal criteria for determining the retrieval timing: 1) **Intent-aware**: whether the user desires retrieval / external information; 2) **Knowledge-aware**: whether the question requires fact knowledge; 3) **Time-Sensitive-aware**: whether the question is time-sensitive; 4) **Self-aware**: whether the LLM has the internal knowledge. As shown in Table 1, compared with previous methods of single criterion (Jiang et al., 2023; Wang et al., 2023b; Asai et al., 2023), UAR can comprehensively handle various types of user instructions and call retrieval accurately considering multiple active retrieval criteria. To efficiently achieve judgements of multiple criteria, UAR unifies each criterion’s judgement into binary classification tasks using lightweight classifiers. For each criterion  $c_i$ , we train a plug-and-play binary classifier on the last layer’s hidden states of a fixed LLM, to judge whether the input requires retrieval according to  $c_i$ . In this way, UAR does not change LLMs’ parameters, avoiding the costly LLM fine-tuning and performance degradation (Yang et al., 2024). Meanwhile, the classifiers and LLM generation share the same input encoding, which makes UAR only need to encode the input once and thus achieves multifaceted retrieval timing judgements with negligible extra inference cost.

To handle various instructions in a unified procedure, we further propose **Unified Active Retrieval Criteria (UAR-Criteria)**, which specifies priorities for multiple retrieval criteria and unifies them into a single multifaceted decision tree. As shown in Figure 2, UAR-Criteria can trigger retrieval for time-sensitive or LLM-unknown instructions, which facilitates necessary external information supplement. Meanwhile, UAR-Criteria cancels retrieval for those non-knowledge-intensive or LLM-known instructions, which avoids the negative effect of unnecessary retrieval. In this way, UAR-Criteria unifies the process to comprehensively decide whether to retrieve for various types of user instructions, which facilitates more effective RAG.

Experiments on four representative types of user instructions show that UAR significantly outperforms existing work on the retrieval timing judgment accuracy and the performance of downstream tasks, which verifies the effectiveness of UAR and its helpfulness to downstream tasks. We summarize our contributions as follows:

- We propose an active retrieval framework named Unified Active Retrieval (UAR) for Retrieval-Augmented Generation (RAG). To the best of our knowledge, UAR is the first work to propose multifaceted criteria for active retrieval and demonstrate its necessity.
- We curate the Active Retrieval benchmark (AR-Bench) for evaluating the accuracy of retrieval timing and conduct comprehensive experiments on AR-Bench and downstream tasks. The results show that UAR significantly outperforms existing work and achieves more efficient RAG.
- We release the code, data, models and relevant resources to facilitate future research.

## 2 Related Work

### 2.1 Active Retrieval

Compared to applying retrieval for every instruction (passive retrieval), active retrieval has advantages such as not hurting the versatility of the model, reducing the number of retrievals, and preventing interference from low-quality retrieval results. Self-RAG (Asai et al., 2023) construct active retrieval data using GPT-4 and teach the model to not retrieve when encounter non-knowledge-intensive instructions. FLARE (Jiang et al., 2023) proposes forward-looking active retrieval augmented generation based on model’s confidence, only retrieving information when the model’s uncertainty for the prediction is high. SKR (Wang et al., 2023b), RA-ISF (Liu et al., 2024) and Self-DC (Wang et al., 2024) first determines whether the model knows the questions and then retrieves only when the model does not know. However, current active retrieval methods mostly consider only a single scenario and are unable to adapt to complex situations in real-world applications.

### 2.2 Time-awareness of LLMs

There are some papers focus on the time awareness of large language models. Chen et al. (2021) construct a time-sensitive QA dataset called TimeQA

to evaluate the model’s ability to handle temporal questions. Fierro et al. (2024) create a benchmark named MULAN for evaluating the ability of language models to predict mutable facts. They find representations classification can distinct immutable and mutable facts, which means language models have a certain degree of temporal awareness. Zhao et al. (2024) investigate whether language models can align their internal knowledge to a target year. They construct a dataset which contains time-sensitive questions.

### 2.3 Self-awareness of LLMs

Self-awareness means that large language model can be aware of what they know and what they don’t know. Kadavath et al. (2022) find that language models can be well-calibrated when using a multiple-choice template. And they also finetune a value head to predict whether language models know the answer to the given question. Lin et al. (2022a) finetune GPT-3 to express uncertainty in words on math questions. Yin et al. (2023) collect some unanswerable questions to evaluate whether language models can express uncertainty to these unanswerable questions. Zhang et al. (2023) utilize supervised fine-tune to teach large language models to refuse questions which beyond their knowledge scope. Cheng et al. (2024) explore more alignment methods beyond supervised fine-tuning to teach language models know and express what they don’t know, like preference optimization. Results of previous work show that we can enhance language models’ self-awareness with corresponding dataset.

## 3 Methodology

UAR is a plug-and-play active retrieval framework. As shown in Figure 2, we fix the parameters of the LLM and train a lightweight classifier for each active retrieval criteria using the model’s hidden states, which is far more efficient than fine-tuning the entire model. Besides, UAR determines the need for active retrieval following the UAR-Criteria shown on the right side of Figure 2, invoking retrieval when necessary and avoiding unnecessary across various scenarios, making RAG more effective and efficient. For instructions requiring retrieval, we append the retrieved documents to the original instruction, which means that UAR does not introduce extra LLM inference cost. We introduce the details of our UAR framework in the following sections.

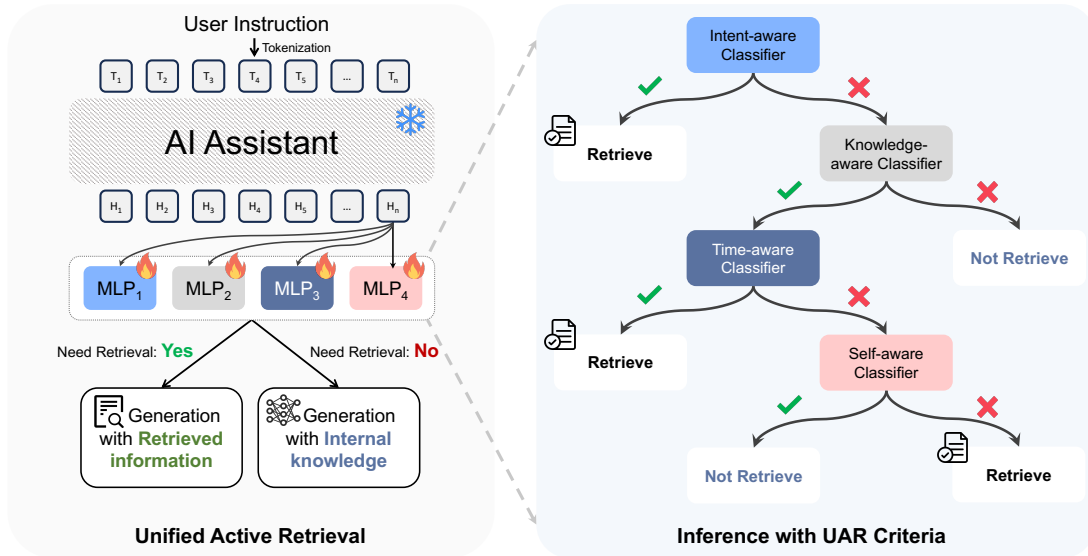


Figure 2: Overview of the UAR framework. ❄ indicates that we freeze these parameters. 🔥 indicates that we update these parameters. Each MLP is a fully connected layer, with an input dimension equal to the model’s hidden state dimension and an output dimension of 2.

### 3.1 UAR Classifiers Training

We construct distinct training data tailored to each scenario.

**Self-aware** In the self-aware scenario, the model must determine if it knows the answer to a given question. Following the methodology in Cheng et al. (2024), we create model-specific IDK (I don’t know) datasets. For example, with the Llama2-7B-chat model, we use the TriviaQA (Joshi et al., 2017) dataset, sampling ten responses for each question. If all responses are correct, the question is marked as known; otherwise, it is unknown. 10% of the TriviaQA training set is used for validation, with the rest designated as the training set.

**Time-aware** In the time-aware scenario, it is critical to determine if a user’s question is time-sensitive, meaning the answer changes over time. We utilize questions from TAQA’s (Zhao et al., 2024) training and validation sets as time-sensitive questions. In contrast, we sample an equivalent number of questions from the TriviaQA training set to represent non-time-sensitive questions, which typically have static answers.

**Knowledge-aware** In the knowledge-aware scenario, identifying whether a user’s instruction requires factual knowledge is essential. We use non-retrieval instruct-following data from the Self-RAG (Asai et al., 2023) training set, which GPT-4 classifies as non-knowledge-intensive. We select 2,000 entries for our validation set and 22,801 for train-

ing. Additionally, we incorporate all entries from our time-aware data’s training and validation sets as knowledge-intensive instructions to complete the final knowledge-aware training and validation sets.

**Intent-aware** In the intent-aware scenario, it’s crucial to identify users’ intentions to use retrieval-augmented generation. Due to a lack of data with explicit retrieval intentions, we use Self-Instruct (Wang et al., 2023c) to generate 3,000 user intents from ten handwritten intents. We allocate 2,000 for training, 500 for validation, and 500 for testing. We assemble user queries by sampling 52,949 entries from Self-RAG’s non-retrieval-required data, and factual knowledge questions from TAQA and TriviaQA for the training set, with an additional 5,000 for validation. We integrate half of these data with user retrieval intents, alternating the position of intents before and after user inputs, to create inputs with retrieval intents. The remaining data are used as inputs without retrieval intents.

For each scenario, we train a single-layer MLP as the classifier, using the hidden states from the last token in the input as the input to the classification head. In this way, UAR can achieve various criteria’s judgements with negligible extra computational cost. We include details of classifiers’ training in Appendix E.

### 3.2 UAR Criteria

We further propose UAR-Criteria to unify the judgements of different types of user instructions

315 in to one unified procedure. During the inference  
316 stage, UAR sequentially utilizes four classifiers  
317 according to different priorities to determine the  
318 correct timing for retrieval calls, and we introduce  
319 its details as follows.

320 Initially, UAR checks whether the user in-  
321 tends to use retrieval augmentation. If so, re-  
322 trieval is triggered. If not, UAR evaluates  
323 whether the input is knowledge-intensive. For non-  
324 knowledge-intensive tasks, retrieval is not used.  
325 For knowledge-intensive tasks, UAR further as-  
326 sesses whether the knowledge is time-sensitive.  
327 Retrieval is necessary for time-sensitive questions.  
328 For non-time-sensitive, knowledge-intensive tasks,  
329 UAR checks whether the model already has the  
330 relevant knowledge, activating retrieval only for  
331 unfamiliar questions. In this way, UAR can han-  
332 dle various types of instructions. Specifically,  
333 UAR-Criteria activates retrieval for instructions  
334 that are time-sensitive, unknown to the model, and  
335 have explicit retrieval intent, which facilitates nec-  
336 essary external information supplement. Mean-  
337 while, UAR-Criteria cancels retrieval for those non-  
338 knowledge-intensive or LLM-known instructions,  
339 which avoids the negative effect of unnecessary re-  
340 trieval. Meanwhile, since UAR achieves the judge-  
341 ment of multifaceted criteria by linear classifiers,  
342 the introduced extra computational cost is negligi-  
343 ble.

### 344 3.3 Generation with Relevant Information

345 For instructions requiring retrieval augmentation,  
346 we append the retrieved external information with  
347 a RAG template to the original user input. Since  
348 most of the prevailing LLMs are based on the  
349 decoder-only architecture (Brown et al., 2020),  
350 UAR can avoid the need to recompute the original  
351 instruction. The retriever might fetch information  
352 irrelevant to the question, our prompt instructs the  
353 model to utilize only the information relevant to  
354 the question. This approach helps prevent irrele-  
355 vant information from misleading the model. An  
356 example of our RAG prompt is as follows:

```
357 {question}  
358 Here are some additional reference passages:  
359 {reference passages}  
360 You can refer to the content of relevant  
361 reference passages to answer the questions.  
362 Now give me the answer.
```

363 For instructions that do not require retrieval, we  
364 allow the model to generate outputs in its original  
365 format.

## 4 Experiments 366

### 4.1 Benchmarking Retrieval Timing 367

368 We curate an Active Retrieval Benchmark (AR-  
369 Bench) to evaluate the accuracy of various active  
370 retrieval methods in determining the timing of re-  
371 trieval. The AR-Bench includes four sub-tasks:  
372 intent-aware, knowledge-aware, time-aware and  
373 self-aware, covering all the active retrieval scenar-  
374 ios mentioned in this paper. Each sub-task is a bi-  
375 nary classification task comprising 8,000 samples,  
376 with a 1:1 ratio of positive to negative examples,  
377 and these samples do not overlap with the train-  
378 ing data of UAR. These four sub-tasks separately  
379 evaluate one single active retrieval criterion and we  
380 control variables to ensure that each task’s retrieval  
381 decision solely depends on one single criterion. We  
382 introduce details of AR-Bench construction in Ap-  
383 pendix A.

### 4.2 Downstream Tasks 384

385 We select six datasets to test UAR’s performance  
386 in real downstream tasks and its adaptability to  
387 different active retrieval scenarios. Since the intent-  
388 aware judgement focuses on satisfying users’ re-  
389 trieval intent, which is not reflected on the objec-  
390 tive downstream performance, the selected datasets  
391 cover the remaining three scenarios: knowledge-  
392 aware, time-aware, and self-aware. For knowledge-  
393 aware scenario, we use DROP (Dua et al., 2019)  
394 and (Cobbe et al., 2021). For time-aware scenario,  
395 we use TAQA (Zhao et al., 2024) and FreshQA  
396 (Vu et al., 2023). For self-aware scenario, we use  
397 TriviaQA (Joshi et al., 2017) and WebQuestions  
398 (WQ) (Berant et al., 2013). We provide a detailed  
399 introduction to these datasets in Appendix F. In  
400 these six datasets, we only use the training sets  
401 of TriviaQA and TAQA for UAR’s training, and  
402 thus the remaining evaluation dataset can reflect  
403 the UAR’s out-of-distribution (OOD) performance,  
404 which can further verify the effectiveness of UAR  
405 in complicated real-world scenarios.

### 4.3 Baselines 406

407 We choose three active retrieval methods as our  
408 baseline methods: FLARE (Jiang et al., 2023), Self-  
409 RAG (Asai et al., 2023), and SKR (Wang et al.,  
410 2023b), covering two main active retrieval criteria.  
411 FLARE determines whether external retrieval is  
412 needed by assessing the model’s uncertainty about  
413 the generated responses. SKR first collects model’s  
414 self-knowledge (knowns and unknowns) data, then

Scenario	Intent-aware	Knowledge-aware	Time-aware	Self-aware	Overall
<i>7B Models</i>					
FLARE	61.95	56.76	53.69	53.59	56.50
Self-RAG <sup>†</sup>	64.26	72.82	47.45	55.95	60.12
SKR	58.73	42.94	76.61	70.28	62.14
UAR	<b>91.88</b>	<b>90.38</b>	<b>86.69</b>	<b>72.32</b>	<b>85.32</b>
<i>13B Models</i>					
FLARE	65.49	53.54	55.20	54.61	57.21
Self-RAG <sup>†</sup>	67.80	64.85	54.44	52.49	59.89
SKR	59.00	43.18	79.91	68.70	62.70
UAR	<b>92.49</b>	<b>91.04</b>	<b>87.94</b>	<b>73.84</b>	<b>86.33</b>

Table 2: Comparisons of active retrieval accuracy on our active retrieval benchmark (AR-Bench). †: Self-RAG is fine-tuned from Llama2-base models. Other methods are based on Llama2-chat models.

415 trains a BERT-based (Devlin et al., 2019) classifier to determine whether the model knows a certain question. For questions the model does not know, retrieval augmentation is used. Self-RAG gathers a large amount of knowledge-intensive and instruction-following data (no fact knowledge required), then trains the pre-trained model to only use retrieval augmentation for knowledge-intensive tasks. For downstream tasks, we also include generation with never-retrieval and always-retrieval as baseline methods. The original SKR and FLARE are not based on Llama2, so we re-implement these methods on the Llama2 model. The details of our re-implementation are provided in Appendix B.

#### 4.4 Retrievers

430 For time-sensitive datasets TAQA and FreshQA, we follow the settings in FreshQA Vu et al. (2023) and use Google Search. For other datasets, following the settings in Self-RAG, we use off-the-shelf Contriever-MS MARCO (Izacard et al., 2022) and retrieve up to ten documents for each input. During generation, we use the top five retrieved documents. For other datasets, following the settings in Self-RAG, we adopt off-the-shelf Contriever-MS MARCO (Izacard et al., 2022) and use the top-5 documents.

#### 4.5 Evaluation Metrics

442 Following previous work (Asai et al., 2023; Mallen et al., 2023; Schick et al., 2023), we check whether gold answers are included in model’s generations to evaluate performance on the DROP, TriviaQA, and WQ datasets, instead of strictly requiring exact matching. For GSM8K, we use the prompts for answer extraction in Kojima et al. (2022) to extract model’s answers and then use exact matching to calculate the accuracy. For TAQA and FreshQA, since

451 the golden answers are too long to conduct lexical matching, we use ChatGPT to evaluate whether the model’s answers are correct. Details of ChatGPT evaluation are included in Appendix C. For AR-Bench, we use accuracy as the metric. Since AR-Bench is a binary classification task with an equal number of positive and negative samples, accuracy and micro F1 score are equivalent.

#### 4.6 Comparisons on AR-Bench

460 We show the results in Table 2. We observe that UAR outperforms existing active retrieval methods across all AR-Bench scenarios, demonstrating its versatility and effectiveness. Since baseline methods depend on a single criterion, they struggle with various active retrieval scenarios, which demonstrates the limitation of single criterion and the necessity of multifaceted decision for active retrieval. Additionally, we find FLARE struggle with self-aware scenario, which it is targeted at. We think it is because its uncertainty estimation heavily depends on model calibration and this leads to its poor performance on less calibrated models like chat models (He et al., 2023) or those with fewer parameters. Self-RAG uses the knowledge-intensive nature of tasks as the retrieval criterion, performing well in knowledge-aware scenarios but poorly in others. SKR bases retrieval on the model’s knowledge of an answer, excelling in self-aware and time-aware scenarios but failing in others. Additionally, since SKR uses BERT as the classifier, whose internal knowledge has a significant gap with Llama, it underperforms UAR with value heads based on the Llama’s representation, in the self-aware scenario.

#### 4.7 Comparisons on Downstream Tasks

485 For Self-RAG, we use inference scripts provided by the authors. For FLARE, SKR, UAR, and always-

Dataset	Drop	GSM8K	TriviaQA	WQ	TAQA	FreshQA	Overall
<i>7B Models</i>							
Never-Ret	57.67 <sub>(0%)</sub>	26.91 <sub>(0%)</sub>	62.15 <sub>(0%)</sub>	59.79 <sub>(0%)</sub>	16.43 <sub>(0%)</sub>	35.64 <sub>(0%)</sub>	43.10
Always-Ret	49.57 <sub>(100%)</sub>	23.65 <sub>(100%)</sub>	68.73 <sub>(100%)</sub>	53.99 <sub>(100%)</sub>	34.49 <sub>(100%)</sub>	65.35 <sub>(100%)</sub>	49.23
<i>Active Retrieval</i>							
Self-RAG <sup>†</sup>	39.17 <sub>(5.7%)</sub>	16.07 <sub>(4.9%)</sub>	61.68 <sub>(53.5%)</sub>	43.01 <sub>(61.9%)</sub>	11.09 <sub>(42.1%)</sub>	44.88 <sub>(51.2%)</sub>	35.98
SKR	53.00 <sub>(61.4%)</sub>	26.38 <sub>(35.3%)</sub>	65.39 <sub>(48.9%)</sub>	58.96 <sub>(26.8%)</sub>	30.63 <sub>(79.9%)</sub>	48.84 <sub>(39.3%)</sub>	47.17
FLARE	<b>56.98</b> <sub>(9.6%)</sub>	26.76 <sub>(45.8%)</sub>	65.98 <sub>(58.8%)</sub>	55.46 <sub>(67.9%)</sub>	28.08 <sub>(63.5%)</sub>	57.76 <sub>(57.4%)</sub>	48.50
UAR	52.55 <sub>(49.7%)</sub>	<b>26.91</b> <sub>(0.1%)</sub>	<b>69.02</b> <sub>(50.1%)</sub>	<b>60.53</b> <sub>(25.0%)</sub>	<b>34.46</b> <sub>(99.7%)</sub>	<b>59.74</b> <sub>(78.5%)</sub>	<b>50.49</b>
<i>13B Models</i>							
Never-Ret	58.76 <sub>(0%)</sub>	40.64 <sub>(0%)</sub>	63.18 <sub>(0%)</sub>	57.63 <sub>(0%)</sub>	11.14 <sub>(0%)</sub>	34.98 <sub>(0%)</sub>	44.39
Always-Ret	54.16 <sub>(100%)</sub>	37.68 <sub>(100%)</sub>	71.02 <sub>(100%)</sub>	54.08 <sub>(100%)</sub>	34.20 <sub>(100%)</sub>	62.05 <sub>(100%)</sub>	52.09
<i>Active Retrieval</i>							
Self-RAG <sup>†</sup>	44.68 <sub>(0.1%)</sub>	21.00 <sub>(0.0%)</sub>	62.53 <sub>(30.0%)</sub>	42.37 <sub>(51.9%)</sub>	15.42 <sub>(37.0%)</sub>	39.60 <sub>(39.3%)</sub>	37.60
SKR	56.58 <sub>(50.9%)</sub>	39.35 <sub>(27.6%)</sub>	67.21 <sub>(49.2%)</sub>	56.20 <sub>(31.5%)</sub>	31.66 <sub>(89.2%)</sub>	50.17 <sub>(45.9%)</sub>	50.16
FLARE	58.12 <sub>(17.5%)</sub>	38.05 <sub>(61.2%)</sub>	68.00 <sub>(54.9%)</sub>	53.64 <sub>(69.6%)</sub>	25.40 <sub>(60.9%)</sub>	50.17 <sub>(55.8%)</sub>	48.90
UAR	<b>58.55</b> <sub>(3.7%)</sub>	<b>40.64</b> <sub>(0.0%)</sub>	<b>71.71</b> <sub>(48.5%)</sub>	<b>59.20</b> <sub>(31.2%)</sub>	<b>34.14</b> <sub>(99.6%)</sub>	<b>55.45</b> <sub>(73.3%)</sub>	<b>53.26</b>

Table 3: Comparisons of downstream tasks performance. Never-Ret means that retrieval augmentation is never used during generation, while Always-Ret means that retrieval augmentation is used in every generation. †: Self-RAG is fine-tuned from Llama2-base models. Other methods are based on Llama2-chat models.

retrieval methods, we use the same prompts to generate responses by incorporating the retrieved information. We introduce the details of generation in Appendix D.

The results are shown in Table 3. We see that UAR leads to the best overall performance across different downstream task scenarios, which indicates its effectiveness. We analyze each scenario as follows.

**UAR does not invoke retrieval when factual knowledge is not needed.** The DROP and GSM8K dataset do not require fact knowledge, and using retrieval enhancement will interfere with the model. The results of always-retrieval are worse than never-retrieval. UAR only invokes a small amount of retrieval, while SKR and FLARE incorrectly invoke retrieval extensively. And since UAR avoid unnecessary retrieval<sup>1</sup> and thus prevents affecting the original capabilities of the LLM, it achieves the best results among all active retrieval methods on DROP and GSM8K, coming close to the results of never-retrieval. Although Self-RAG does not incorrectly invoke retrieval, its final performance is not very good because it is fine-tuned based on the base model rather than leveraging the capabilities of the chat model.

<sup>1</sup>UAR based on the 7B model incorrectly invokes retrieval 50% of the time on the DROP dataset. We speculate that this may be due to the limited representation capacity of the 7B model’s hidden states. In contrast, the 13B model only incorrectly invokes retrieval 3.7% of the time.

**UAR accurately invokes retrieval for time-sensitive questions.** Since the questions in TAQA and FreshQA are time-sensitive and their answers keep changing, each question requires the retrieval of the latest information. It is evident that the always-retrieval method based on Google Search performs significantly better than the never-retrieval method. For TAQA, UAR almost perfectly invokes retrieval. For FreshQA, UAR also invokes retrieval for most of the questions. In contrast, other methods invoke retrieval less frequently and therefore do not use the latest information for responses, resulting in lower accuracy compared to UAR.

**UAR accurately assesses the model’s knowledge, avoiding poor retrieval impacts.** For questions in TriviaQA and WQ whose answers do not change over time, always-retrieval is sub-optimal and the reason is two-fold: 1. For questions which model knows, retrieval increases unnecessary latency. 2. Potential incorrect external information will interfere correct internal knowledge. Retrieving information only for knowledge that the model does not know can mitigate this issue. Compared to SKR, UAR can more accurately determine whether the model knows a particular piece of knowledge. Although SKR and UAR use a comparable number of retrieval calls, the accuracy of SKR’s answers is lower than that of UAR, indicating that SKR’s retrieval calls are less precise than UAR’s. We

believe this is because SKR uses independent models, whereas our approach uses hidden states of the original model, resulting in better generalization. Moreover, UAR outperforms always-retrieval with fewer retrieval calls, demonstrating the superiority of the Active Retrieval method.

## 5 Analysis

### 5.1 Single Classifiers vs UAR

Scenario	Single Classifier	UAR
Intent-aware	98.29	91.88
Knowledge-aware	99.66	90.38
Time-aware	99.41	86.69
Self-aware	<u>72.56</u>	<u>72.32</u>

Table 4: Comparison between single classifiers and UAR based on Llama2-7B-chat.

Different scenarios have varying levels of discrimination difficulty. As shown in Table 4, the single classifier for the self-aware scenario has the lowest accuracy, which implies that determining whether the model is self-aware is a relatively challenging task. We can also observe that the accuracy of each single classifier is higher than UAR in their respective scenarios. The self-aware classifier may become the bottleneck restricting the performance of UAR, which also results in the accuracy of UAR on the AR-Bench being lower than the accuracy of using a single classifier alone.

### 5.2 Using the Whole LLM as Classifier

Self-aware	Only Value Head	Whole LLM
Llama2-7B-chat	72.56	75.65
Llama2-13B-chat	73.48	76.28

Table 5: Comparison of the performance between training a value head as the classifier and training an entire large language model as the classifier.

To improve the performance bottleneck of the self-aware classifier, we attempt to fine-tune the entire large language model as the classifier. From the results in Table 5, we can observe that on both 7B and 13B models, fine-tuning the entire model only achieves slight higher accuracy compared to just fine-tuning a lightweight value head. Using a whole LLM as the classifier, UAR’s inference latency and required parameters will significantly increase. Therefore, we use lightweight value heads as classifiers, ensuring the efficiency of the entire framework with minimal performance loss.

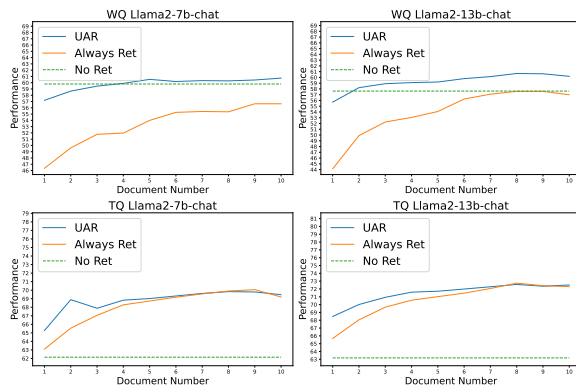


Figure 3: The impact of the number of reference documents on model performance.

### 5.3 The Impact of Document Number

We evaluate performance on the TriviaQA (TQ) and WebQuestions (WQ) datasets by varying the number of reference documents from 1 to 10. The results, shown in Figure 3, indicate that on the WQ dataset, the always-retrieval method performs worse than the never-retrieval method, possibly because some documents disrupt the correct knowledge within the model. UAR reduces retrieval frequency, enabling more precise retrieval calls and outperforming the never-retrieval method. On the TQ dataset, always-retrieval outperforms never-retrieval, and performance improves with more documents, suggesting useful information might be in lower-ranked documents. UAR performs best with fewer documents. With more documents, it matches the performance of always-retrieval, although it requires significantly fewer retrieval calls.

## 6 Conclusion

In this paper, we introduce UAR, a unified active retrieval framework for retrieval-augmented generation. Unlike existing methods that rely on a single criterion, UAR incorporates four orthogonal criteria into plug-and-play classification tasks, enabling comprehensive retrieval timing judgments with minimal inference cost and no loss of model capabilities. We also introduce UAR-Criteria for processing various active retrieval scenarios uniformly. We curate the Active Retrieval Benchmark (AR-Bench) to assess the retrieval timing accuracy of active retrieval methods across different scenarios. Experimental results demonstrate that UAR significantly outperforms existing methods on AR-Bench and downstream tasks, highlighting its effectiveness and benefits to downstream applications.



## 611 Limitations

612 We summarize limitations of our work as follows:

- 613 • Our experiments primarily focus on the gen-  
614 eration of short texts, such as in knowledge-  
615 based question answering, and involve only a  
616 single retrieval call. How to implement mul-  
617 tiple active retrieval calls within longer text  
618 responses remains an area for future investiga-  
619 tion.
- 620 • Our active retrieval criteria are primarily de-  
621 rived from our experience in practical appli-  
622 cations, which may overlook some active re-  
623 trieval scenarios.
- 624 • Our classifier is based on a single-layer MLP  
625 network. Whether using a deeper network can  
626 further enhance performance remains to be  
627 explored.

## 628 References

629 Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-  
630 Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan  
631 Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Mil-  
632 lican, David Silver, Slav Petrov, Melvin Johnson,  
633 Ioannis Antonoglou, Julian Schrittwieser, Amelia  
634 Glaese, Jilin Chen, Emily Pitler, Timothy P. Lilli-  
635 crap, Angeliki Lazaridou, Orhan Firat, James Molloy,  
636 Michael Isard, Paul Ronald Barham, Tom Hennig-  
637 an, Benjamin Lee, Fabio Viola, Malcolm Reynolds,  
638 Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens  
639 Meyer, Eliza Rutherford, Erica Moreira, Kareem  
640 Ayoub, Megha Goel, George Tucker, Enrique Pi-  
641 queras, Maxim Krikun, Iain Barr, Nikolay Savinov,  
642 Ivo Danihelka, Becca Roelofs, Anaïs White, Anders  
643 Andreassen, Tamara von Glehn, Lakshman Yagati,  
644 Mehran Kazemi, Lucas Gonzalez, Misha Khalman,  
645 Jakub Sygnowski, and et al. 2023. *Gemini: A fam-  
646 ily of highly capable multimodal models*. *CoRR*,  
647 abs/2312.11805.

648 Anthropic. 2023. *Introducing claude*.

649 Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and  
650 Hannaneh Hajishirzi. 2023. *Self-rag: Learning to  
651 retrieve, generate, and critique through self-reflection*.  
652 *CoRR*, abs/2310.11511.

653 Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang,  
654 Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei  
655 Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin,  
656 Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu,  
657 Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren,  
658 Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong  
659 Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang  
660 Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian  
661 Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi

662 Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang,  
663 Yichang Zhang, Zhenru Zhang, Chang Zhou, Jin-  
664 gren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023.  
665 *Qwen technical report*. *CoRR*, abs/2309.16609.

666 Jonathan Berant, Andrew Chou, Roy Frostig, and Percy  
667 Liang. 2013. *Semantic parsing on freebase from  
668 question-answer pairs*. In *Proceedings of the 2013  
669 Conference on Empirical Methods in Natural Lan-  
670 guage Processing, EMNLP 2013, 18-21 October  
671 2013, Grand Hyatt Seattle, Seattle, Washington, USA,  
672 A meeting of SIGDAT, a Special Interest Group of the  
673 ACL*, pages 1533–1544. ACL.

674 Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie  
675 Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind  
676 Neelakantan, Pranav Shyam, Girish Sastry, Amanda  
677 Askell, Sandhini Agarwal, Ariel Herbert-Voss,  
678 Gretchen Krueger, Tom Henighan, Rewon Child,  
679 Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu,  
680 Clemens Winter, Christopher Hesse, Mark Chen, Eric  
681 Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess,  
682 Jack Clark, Christopher Berner, Sam McCandlish,  
683 Alec Radford, Ilya Sutskever, and Dario Amodei.  
684 2020. *Language models are few-shot learners*. In *Ad-  
685 vances in Neural Information Processing Systems 33:  
686 Annual Conference on Neural Information Process-  
687 ing Systems 2020, NeurIPS 2020, December 6-12,  
688 2020, virtual*.

689 Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen,  
690 Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi  
691 Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan,  
692 Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe  
693 Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He,  
694 Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao,  
695 Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li,  
696 Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hong-  
697 wei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu,  
698 Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv,  
699 Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang  
700 Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai  
701 Shang, Yunfan Shao, Demin Song, Zifan Song, Zhi-  
702 hao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang,  
703 Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang,  
704 Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen  
705 Weng, Fan Wu, Yingtong Xiong, and et al. 2024.  
706 *Internlm2 technical report*. *CoRR*, abs/2403.17297.

707 Wenhu Chen, Xinyi Wang, and William Yang Wang.  
708 2021. *A dataset for answering time-sensitive ques-  
709 tions*. In *Proceedings of the Neural Information Pro-  
710 cessing Systems Track on Datasets and Benchmarks  
711 1, NeurIPS Datasets and Benchmarks 2021, Decem-  
712 ber 2021, virtual*.

713 Qinyuan Cheng, Tianxiang Sun, Xiangyang Liu, Wen-  
714 wei Zhang, Zhangyue Yin, Shimin Li, Linyang Li,  
715 Zhengfu He, Kai Chen, and Xipeng Qiu. 2024. *Can  
716 AI assistants know what they don't know?* *CoRR*,  
717 abs/2401.13275.

718 Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng,  
719 Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan  
720 Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion

721	Stoica, and Eric P. Xing. 2023. <a href="#">Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.</a>	
722		
723		
724	Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. <a href="#">Training verifiers to solve math word problems.</a> <i>CoRR</i> , abs/2110.14168.	
725		
726		
727		
728		
729		
730	Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. <a href="#">BERT: pre-training of deep bidirectional transformers for language understanding.</a> In <i>Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)</i> , pages 4171–4186. Association for Computational Linguistics.	
731		
732		
733		
734		
735		
736		
737		
738		
739		
740	Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, and Xueqi Cheng. 2024. <a href="#">Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models.</a> <i>Preprint</i> , arXiv:2402.10612.	
741		
742		
743		
744		
745	Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. <a href="#">DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs.</a> In <i>Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)</i> , pages 2368–2378. Association for Computational Linguistics.	
746		
747		
748		
749		
750		
751		
752		
753		
754		
755	Constanza Fierro, Nicolas Garneau, Emanuele Bugliarello, Yova Kementchedjhiya, and Anders Søgaard. 2024. <a href="#">Mulan: A study of fact mutability in language models.</a> <i>CoRR</i> , abs/2404.03036.	
756		
757		
758		
759	Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. <a href="#">Retrieval-augmented generation for large language models: A survey.</a> <i>Preprint</i> , arXiv:2312.10997.	
760		
761		
762		
763		
764	Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. <a href="#">Realm: Retrieval-augmented language model pre-training.</a> <i>Preprint</i> , arXiv:2002.08909.	
765		
766		
767		
768	Guande He, Peng Cui, Jianfei Chen, Wenbo Hu, and Jun Zhu. 2023. <a href="#">Investigating uncertainty calibration of aligned language models under the multiple-choice setting.</a> <i>CoRR</i> , abs/2310.11732.	
769		
770		
771		
772	Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. <a href="#">Unsupervised dense information retrieval with contrastive learning.</a> <i>Trans. Mach. Learn. Res.</i> , 2022.	
773		
774		
775		
776		
	Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. <a href="#">Active retrieval augmented generation.</a> In <i>Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023</i> , pages 7969–7992. Association for Computational Linguistics.	777
		778
		779
		780
		781
		782
		783
		784
	Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. <a href="#">Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension.</a> In <i>Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers</i> , pages 1601–1611. Association for Computational Linguistics.	785
		786
		787
		788
		789
		790
		791
		792
	Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. <a href="#">Language models (mostly) know what they know.</a> <i>CoRR</i> , abs/2207.05221.	793
		794
		795
		796
		797
		798
		799
		800
		801
		802
		803
		804
		805
	Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. <a href="#">Large language models are zero-shot reasoners.</a> In <i>Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022.</i>	806
		807
		808
		809
		810
		811
		812
	Stephanie Lin, Jacob Hilton, and Owain Evans. 2022a. <a href="#">Teaching models to express their uncertainty in words.</a> <i>Trans. Mach. Learn. Res.</i> , 2022.	813
		814
		815
	Stephanie Lin, Jacob Hilton, and Owain Evans. 2022b. <a href="#">Truthfulqa: Measuring how models mimic human falsehoods.</a> In <i>Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022</i> , pages 3214–3252. Association for Computational Linguistics.	816
		817
		818
		819
		820
		821
		822
	Yanming Liu, Xinyue Peng, Xuhong Zhang, Weihao Liu, Jianwei Yin, Jiannan Cao, and Tianyu Du. 2024. <a href="#">RA-ISF: learning to answer and understand from retrieval augmentation via iterative self-feedback.</a> <i>CoRR</i> , abs/2403.06840.	823
		824
		825
		826
		827
	Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. <a href="#">When not to trust language models: Investigating effectiveness of parametric and non-parametric memories.</a> In <i>Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada,</i>	828
		829
		830
		831
		832
		833
		834

835		July 9-14, 2023, pages 9802–9822. Association for Computational Linguistics.		892
836				893
837	OpenAI.	2022. <a href="#">Introducing chatgpt.</a>		894
838	OpenAI.	2023. <a href="#">Gpt-4 technical report.</a>		895
839	Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom.	2023. <a href="#">Toolformer: Language models can teach themselves to use tools.</a> In <i>Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.</i>		896
840				897
841				898
842				899
843				900
844				901
845				902
846				903
847	Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen.	2023. <a href="#">Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy.</a> In <i>Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023,</i> pages 9248–9274. Association for Computational Linguistics.		904
848				905
849				906
850				907
851				908
852				909
853				910
854				911
855	Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou.	2023. <a href="#">Large language models can be easily distracted by irrelevant context.</a> In <i>International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA,</i> volume 202 of <i>Proceedings of Machine Learning Research,</i> pages 31210–31227. PMLR.		912
856				913
857				914
858				915
859				916
860				917
861				918
862				919
863	Tianxiang Sun, Xiaotian Zhang, Zhengfu He, Peng Li, Qinyuan Cheng, Xiangyang Liu, Hang Yan, Yunfan Shao, Qiong Tang, Shiduo Zhang, Xingjian Zhao, Ke Chen, Yining Zheng, Zhejian Zhou, Ruixiao Li, Jun Zhan, Yunhua Zhou, Linyang Li, Xiaogui Yang, Lingling Wu, Zhangyue Yin, Xuanjing Huang, Yungang Jiang, and Xipeng Qiu.	2024. <a href="#">Moss: An open conversational large language model.</a> <i>Machine Intelligence Research.</i>		920
864				921
865				922
866				923
867				924
868				925
869				926
870				927
871				928
872	Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto.	2023. <a href="#">Stanford alpaca: An instruction-following llama model.</a> <a href="https://github.com/tatsu-lab/stanford_alpaca">https://github.com/tatsu-lab/stanford_alpaca.</a>		929
873				930
874				931
875				932
876				933
877	Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten,			934
878				935
879				936
880				937
881				938
882				939
883				940
884				941
885				942
886				943
887				944
888				945
889				946
890				947
891				948
				949
				950
				951
				952
				953
				954
				955
				956
				957
				958
				959
				960
				961
				962
				963
				964
				965
				966
				967
				968
				969
				970
				971
				972
				973
				974
				975
				976
				977
				978
				979
				980
				981
				982
				983
				984
				985
				986
				987
				988
				989
				990
				991
				992
				993
				994
				995
				996
				997
				998
				999
				1000

951 Self-distillation bridges distribution gap in language  
952 model fine-tuning. *CoRR*, abs/2402.13669.

953 Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu,  
954 Xipeng Qiu, and Xuanjing Huang. 2023. Do large  
955 language models know what they don’t know? In  
956 *Findings of the Association for Computational Lin-*  
957 *guistics: ACL 2023, Toronto, Canada, July 9-14,*  
958 *2023*, pages 8653–8665. Association for Computa-  
959 tional Linguistics.

960 Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan  
961 Berant. 2023. Making retrieval-augmented lan-  
962 guage models robust to irrelevant context. *CoRR*,  
963 abs/2310.01558.

964 Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang,  
965 Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu,  
966 Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma,  
967 Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan  
968 Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023.  
969 *GLM-130B: an open bilingual pre-trained model.* In  
970 *The Eleventh International Conference on Learning*  
971 *Representations, ICLR 2023, Kigali, Rwanda, May*  
972 *1-5, 2023*. OpenReview.net.

973 Hanning Zhang, Shizhe Diao, Yong Lin, Yi R. Fung,  
974 Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji,  
975 and Tong Zhang. 2023. R-tuning: Teaching large lan-  
976 guage models to refuse unknown questions. *CoRR*,  
977 abs/2311.09677.

978 Bowen Zhao, Zander Brumbaugh, Yizhong Wang, Han-  
979 naneh Hajishirzi, and Noah A. Smith. 2024. Set the  
980 clock: Temporal alignment of pretrained language  
981 models. *CoRR*, abs/2402.16797.

## 982 A Details of AR-Bench Construction

983 For the self-aware task, we employ the same  
984 method as described in Section 3.1 to construct test  
985 samples on the TriviaQA validation set. Questions  
986 the model does not know are marked as requiring  
987 retrieval. The test set comprise 4000 questions the  
988 model knows and 4000 questions it does not.

989 For the time-aware task, we use 4000 time-  
990 sensitive questions from the TAQA test set as inputs  
991 requiring retrieval, and 4000 questions the model  
992 knows from the TriviaQA validation set as inputs  
993 not requiring retrieval.

994 For the knowledge-aware task, we use 4000 sam-  
995 ples from the Self-RAG non-retrieval training data  
996 as inputs not requiring retrieval, and combine 2000  
997 time-sensitive questions from the TAQA test set  
998 with 2000 questions the model does not know from  
999 the TriviaQA validation set as inputs requiring re-  
1000 trieval.

1001 For the intent-aware task, we use 4000 questions  
1002 the model knows from the TriviaQA validation

set and 4000 instructions from the Self-RAG non-  
retrieval training data, half of which are concate-  
nated with user retrieval intents as inputs requiring  
retrieval, and the other half as inputs not requiring  
retrieval.

It is important to note that the self-aware data for  
different models may vary, leading to different AR-  
Benches for different models. In our experiments,  
we curate two separate AR-Benches for Llama2-  
7B-chat and Llama2-13B-chat respectively.

## B Details of Baselines Re-implementation

### B.1 FLARE

In implementing FLARE, we make two modifica-  
tions. First, we conduct experiments based on the  
Llama2-chat series of models, rather than using  
text-davinci-003. Second, we eliminate the initial  
retrieval step in FLARE since our setting is active  
retrieval rather than passive retrieval. We find that  
FLARE based on Llama2 struggle to achieve sat-  
isfactory results, which we suspect may be due  
to poor calibration of the Llama2-7B-chat and  
Llama2-13B-chat models. The uncertainty estima-  
tion in FLARE heavily relies on model calibration,  
making it challenging to adapt to poorly calibrated  
models. Therefore, on the AR-Bench, we conduct  
a direct search for the best retrieval thresholds for  
FLARE, ultimately setting them at 0.006 and 0.02  
for the Llama2-7B-chat and Llama2-13B-chat mod-  
els, respectively.

### B.2 SKR

Training Hyper-parameters	
Optimizer	AdamW
Warmup Steps	0
Learning Rate	2e-5
Batch Size	32
Train Epochs	5
LR Scheduler	Linear
Max-seq-length	512

Table 6: Training hyper-parameters of SKR.

In implementing SKR, we first use the 849 origi-  
nal pieces of data provided by the authors of SKR  
and collect self-knowledge data for the Llama2-7B-  
chat model according to the scripts in SKR’s code  
repository. We obtain 15 questions that the model  
does not know and 143 questions that it knows, and  
find that these data are not sufficient to train an ef-  
fective BERT classifier. Therefore, we use the data  
from our training data of the self-aware classifier

1042	to train the BERT classifier for SKR. Our training		
1043	hyper-parameters are shown in Table 6.		
1044	<b>C ChatGPT Evaluation</b>		
1045	We use gpt-3.5-turbo-instruct as the evaluator. Dur-		
1046	ing the evaluation, we input the correct answer and		
1047	the answer to be evaluated into gpt-3.5, and then		
1048	let the model compare the correct answer with the		
1049	answer to be evaluated to determine if the latter is		
1050	correct. Following Shao et al. (2023), we use the		
1051	following prompt for evaluation.		
1052	In the following task, you are given a Question,		
1053	a model Prediction for the Question, and a		
1054	Ground-truth Answer to the Question. You should		
1055	decide whether the model Prediction implies the		
1056	Ground-truth Answer.		
1057			
1058	Question:		
1059	{question}		
1060			
1061	Prediction:		
1062	{predicted answer}		
1063			
1064	Ground-truth Answer:		
1065	{ground-truth answer}		
1066	Does the Prediction imply the Ground-truth		
1067	Answer? Output Yes or No:		
1068	<b>D Details of Generation</b>		
1069	<b>D.1 Self-RAG</b>		
1070	We use the inference script provided by the Self-		
1071	RAG authors for generation. We determine the		
1072	need for retrieval by whether the retrieval special to-		
1073	ken appears in the generated response. For datasets		
1074	using Contriever-MS MARCO as the retriever, we		
1075	provide all 10 documents retrieved to Self-RAG for		
1076	generation.		
1077	<b>D.2 Generation without Retrieval</b>		
1078	For the DROP dataset, we use the following		
1079	prompt:		
1080	Please answer the question based on the given		
1081	passage.		
1082	Passage: {passage in the dataset}		
1083	Question: {question}		
1084	Now give me the answer.		
1085	For the GSM8K dataset, we use the following		
1086	prompt:		
1087	Answer the math word question step by step. Your		
1088	answer needs to end with 'The answer is'.		
1089	Question: {question}		
1090	Let's think step by step and give me the answer.		
1091	For other datasets, we directly input the question		
1092	to the model:		
1093	{question}		
	<b>D.3 Generation with Retrieval</b>		1094
	For the DROP dataset, we use the following		1095
	prompt:		1096
	Please answer the question based on the given		1097
	passage.		1098
	Passage: {passage in the dataset}		1099
	Question: {question}		1100
			1101
	Here are some additional reference passages:		1102
	{retrieved documents}		1103
			1104
	You can refer to the content of relevant		1105
	reference passages to answer the questions.		1106
	Now give me the answer.		1107
	For the GSM8K dataset, we use the following		1108
	prompt:		1109
	Answer the math word question step by step. Your		1110
	answer needs to end with 'The answer is'		1111
	Question: {question}		1112
			1113
	Here are some additional reference passages:		1114
	{retrieved documents}		1115
			1116
	You can refer to the content of relevant		1117
	reference passages to answer the questions.		1118
	Let's think step by step and give me the answer.		1119
	For other datasets, we use the following prompt:		1120
	{question}		1121
			1122
	Here are some additional reference passages:		1123
	{retrieved documents}		1124
			1125
	You can refer to the content of relevant		1126
	reference passages to answer the questions.		1127
	Now give me the answer.		1128
	<b>E Details of UAR Training</b>		1129
	When training the UAR classifiers, we set the batch		1130
	size to 32 and train for a total of 10 epochs, saving		1131
	after each epoch and selecting the checkpoint that		1132
	perform best on the validation set. We conduct a		1133
	grid search on the validation set and ultimately de-		1134
	termine the learning rate to be 5e-5. Our classifier		1135
	is a fully connected layer with an input dimension		1136
	equal to the hidden state dimension and an output		1137
	dimension of 2.		1138
	<b>F Downstream Task Datasets</b>		1139
	For knowledge-aware scenario, we use the valida-		1140
	tion set of DROP (Dua et al., 2019) and the test		1141
	set of GSM8K (Cobbe et al., 2021) as the test		1142
	sets. DROP is a reading comprehension bench-		1143
	mark, which needs the model to answer questions		1144
	based on given paragraphs. GSM8K is a dataset		1145
	containing diverse grade school math word prob-		1146
	lems, primarily used to assess the reasoning ability		1147
	of models. These two datasets evaluate the model's		1148

1149 abstract abilities, e.g., reading comprehension and  
1150 math reasoning, and thus do not require extra fact  
1151 knowledge. Therefore, they can measure the abil-  
1152 ity of active retrieval methods to avoid unnecessary  
1153 retrieval for scenarios that requires little fact knowl-  
1154 edge.

1155 For time-aware scenario, we use the test set of  
1156 TAQA (Zhao et al., 2024) and questions whose an-  
1157 swers will change over time from FreshQA (Vu  
1158 et al., 2023) (We remove questions with false  
1159 premises). Since these questions are time-sensitive,  
1160 the active retrieval system need to retrieve real-time  
1161 information for every question.

1162 For self-aware scenario, we use the validation set  
1163 of TriviaQA (Joshi et al., 2017) and the test set of  
1164 WebQuestions (WQ) (Berant et al., 2013). These  
1165 test samples are non-time-sensitive questions. The  
1166 active retrieval system only needs to retrieve ques-  
1167 tions which the model does not know, and try to  
1168 achieve high answer accuracy with an appropriate  
1169 number of retrieval calls.