Semantic Contribution-Aware Adaptive Retrieval for Black-Box Models

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) plays a critical role in mitigating hallucinations and improving factual accuracy for Large Language Models (LLMs). While dynamic retrieval techniques aim to determine retrieval timing and content based on model intrinsic needs, existing approaches struggle to generalize effectively in black-box model scenarios. To address this limitation, we propose the Semantic Contribution-Aware Adaptive Retrieval (SCAAR) framework. SCAAR iteratively leverages the semantic importance of words in upcoming sentences to dynamically adjust retrieval thresholds and filter information, retaining the top-$\alpha\%$ most semantically significant words for constructing retrieval queries. We comprehensively evaluate SCAAR against baseline methods across four long-form, knowledge-intensive generation datasets using four models. Our method achieved the highest score on each dataset with GPT-4o. Extensive experiments also analyze the impact of various hyperparameters within the framework. Our results demonstrate SCAAR's superior or competitive performance, showcasing its ability to effectively detect model retrieval needs and construct efficient retrieval queries for relevant knowledge about problem-solving in black-box scenarios. Our code is publicly available on GitHub.

1 Introduction

Large Language Models (LLMs) demonstrate impressive capabilities in various natural language processing tasks such as question-answering (QA), abstractive summarization, and machine translation (Zhao et al., 2023). The emergence of prompt tuning and in-context learning (Brown et al., 2020; Zhou et al., 2022; Chan et al., 2022) facilitates LLMs to generate convincing and human-like responses, enabling LLMs to be integrated into AI-powered intelligent assistants to support human reasoning and decision-making processes (OpenAI, 2022; Achiam et al., 2023). However, when confronting time-dependent and complex reasoning tasks, LLMs demonstrate reasoning inconsistencies and factual inaccuracies during response generation, which is referred to as the hallucination of LLMs (Huang et al., 2023).

Retrieval-Augmented Generation (RAG) (Guu et al., 2020; Lewis et al., 2020) alleviates the hallucination issue by incorporating relevant knowledge into the context during the reasoning, enhancing the model's reasoning ability (Ram et al., 2023). The conventional RAG framework implements a single retrieval operation upon a question and leverages the retrieved knowledge to assist the response generation (Izacard et al., 2022; Luo et al., 2023). However it shows limited performance in long-form generation and tasks requiring multi-step reasoning. This limitation stems from single-step retrieval, which only retrieves knowledge relevant to the initial question, neglecting the potential need during the iterative generation process.

Recent work focuses on the problem of when and what to retrieve during the generation process of LLMs. IRCoT (Trivedi et al., 2022) triggers retrieval at the end of each sentence, and Toolformer (Schick et al., 2023) triggers retrieval when seeing named entities. Meanwhile, adaptive retrieval has received increasing attention. The advantage of the adaptive retrieval lies in its ability to decide whether to trigger retrieval and determine the query for retrieval in accordance with status of the model. Adaptive retrieval avoids unnecessary retrieval overhead and reduces the interference caused by wrong retrievals, improving the quality of the query and the retrieved content. Recent work has explored different implementations of adaptive retrieval. FLARE (Jiang et al., 2023) uses the probability of the generated tokens to determine whether to retrieve and uses the model's current generation as the query, treating low-confidence tokens as hallucinations. DRAGIN (Su et al., 2024)

proposes an attention-based dynamic retrieval determination criterion assigns different significance values to content words and stopwords when building the query for retrieval. SeaKR (Yao et al., 2024) proposes a retrieval determination criterion based on self-aware uncertainty. These methods effectively enhance RAG, but they rely on models' hidden states and can't work with black-box models. Moreover, the query formulation schemes of these methods revolve around the entire sentence or part of speech, which lacks dynamism.

In this work, we focus on threshold adaptive weighting schemes that work in black-box scenarios and retrieval problem construction schemes based on these weights. Following the definition in previous works (Cífka and Liutkus, 2023; Kuang et al., 2024; Shi et al., 2023), in a black-box scenario, we can only obtain the probabilities information corresponding to the tokens returned by the model, and cannot obtain other content. We propose **S**emantic **C**ontribution-**A**ware **A**daptive **R**etrieval (SCAAR) as shown in Figure 1, which adopts an encoder model to compute the semantic contribution value of each token. The semantic contribution values are then leveraged to dynamically adjust the retrieval threshold and filter low-importance words in the query for retrieval.

We compared our SCAAR against white-box adaptive retrieval approaches and static retrieval approaches on four knowledge-intensive datasets. Experimental results show that SCAAR achieves a performance comparable to adaptive white-box retrieval approaches, indicating that SCAAR can effectively capture the value of each token and determine 'when to retrieve' in black-box settings. Additionally, the construction of contribution-based queries in SCAAR outperforms existing approaches, indicating that SCAAR can better determine 'what to retrieve'.

The contributions of our paper are as follows:

- We present a semantic contribution-based adaptive weighting (SCW) method, which accurately captures the model's inherent need of retrieval under black-box settings.
- We propose a Percentile-Filtered Query (PFQ) construction based on semantic contribution, filtering unimportant information in upcoming sentences for better retrieval.
- We empirically demonstrate that our SCAAR framework composed of SCW and PFQ achieves superior performance compared to baselines on four knowledge-intensive datasets.

2 Related Work

2.1 Adaptive Retrieval

Conventional RAG frameworks generally determine to perform retrieval at a fixed time or based on simple rules, for example, every question (Khandelwal et al., 2019), every N tokens (Borgeaud et al., 2022; Ram et al., 2023) or every N sentences (Shi et al., 2023). Such mechanisms frequently fail to match the knowledge need of models, and even weaken final performance with unrelated retrieved contents (Mallen et al., 2022).

Adaptive retrieval methods dynamically determine whether to retrieve by sensing the potential quality issues during model generation. Existing adaptive retrieval approaches can be based on question difficulty assessment (Mallen et al., 2022; Li et al., 2023; Asai et al., 2023), uncertainty qualification (Su et al., 2024; Yao et al., 2024; Jiang et al., 2023), and retrieval result postprocessing (Wang et al., 2023; Xu et al., 2023; Yao et al., 2024), among which the approaches based on uncertainty qualification are most relevant to our work.

FLARE (Jiang et al., 2023) is the fundamental work that applies uncertainty qualification to RAG. If the confidence of any token is lower than a preset threshold, FLARE triggers retrieval and uses the remaining tokens to compose a query for retrieval. FLARE effectively explores the model generation intention and requirement, but lacks flexibility due to the fixed threshold.

DRAGIN (Su et al., 2024) dynamically sets a threshold for each token based on its attention score, where tokens with higher attention scores are regarded as more significant so they are assigned higher thresholds. However, this approach cannot be generalized to black-box models.

Our mechanism assigns dynamic thresholds to different tokens by incorporating a lightweight language model to quantify token semantic significance as weighting factors of thresholds, introducing minimal computational overhead but enhancing performance metrics in both white-box and black-box scenarios.

2.2 Retrieval for Black-Box Models

Adaptive retrieval works generally focus on white-box models since the LLMs' internal states are

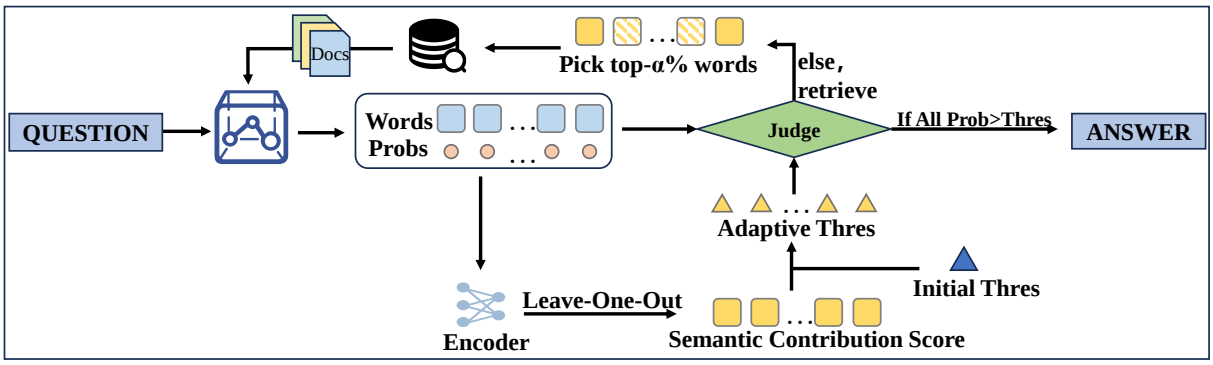


Figure 1: SCAAR dynamically adjust thresholds for words based on semantic contribution score and keep top-α words to construct query.

Table 1: Comparison of different methods.

Method	Adaptive	Dynamic	Scenario white-box	Scenario black-box
FS-RAG (2022)	×	×	✓	✓
FLARE (2023)	✓	×	✓	✓
DRAGIN (2024)	✓	✓	✓	×
SCAAR(Ours)	✓	✓	✓	✓

considered to be significant in hallucination detection (Chen et al., 2024). However, some powerful models such as GPT-4 do not provide any information of the internal states, posing a challenge to perform RAG based on these models. Existing black-box approaches focus on the consistency between multiple responses for the question to assist retrieval determination. The more consistent answers are, the more likely the model is to know the correct answer. Otherwise, the model tend to give hallucinated responses with high semantic diversity. Fomicheva et al. (Fomicheva et al., 2020) employ Meteor score to quantify the consistency of multiple responses. Lin et al. (Lin et al., 2023) propose to use semantic sets and graph Laplacian eigenvalues to estimate the uncertainty and confidence from the Jaccard similarities over multiple generations. Manakul et al. (Manakul et al., 2023) consider the similarities adopted in the above two approaches. Farquhar et al. (Farquhar et al., 2024) construct different queries for the specific idea generated by the LLM and determine the factuality of the idea by the consistency of the final results over different queries. These approaches facilitate hallucination detection in black-box models and achieve effective performances, but still introduces much computational complexity due to the need for a large amount of extra generations.

The comparison of the characteristics of different methods is shown in the Table 1.

3 Methodology

3.1 Formulation of Adaptive Retrieval

Given a language model M and a user question $\mathbf{q}$, the generated response of the language model can be denoted as $\mathbf{y} = M(\mathbf{q})$. Here, the response $\mathbf{y}$ can be regarded as a sequence of sentences, i.e., $\mathbf{y} = [\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n]$, where each sentence $\mathbf{s}_i$ can be regarded as a sequence of words, i.e., $\mathbf{s}_i = [w_{i,1}, w_{i,2}, \cdots, w_{i,m}]$.

A knowledge base in an RAG framework can be denoted as a set of general Wikipedia or customized documents $\mathcal{D} = \{\mathbf{d}_i\}_{i=1}^{|\mathcal{D}|}$, where $\mathbf{d}_i$ is a single document. The RAG framework is able to retrieve the k documents most relevant to the user question $\mathbf{q}$ from the knowledge base $\mathcal{D}$. The set of the retrieved k documents is referred to as the context knowledge, denoted as $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_k\}$, where $\mathbf{c}_i \in \mathcal{D}$. The model M then performs augmented generation $\mathbf{y}' = M(\mathcal{C}, \mathbf{q})$ using the context knowledge $\mathcal{C}$ and the original user question $\mathbf{q}$. Generally, when given relevant retrieved context, the quality of output $\mathbf{y}'$ is superior to that of $\mathbf{y}$.

Adaptive retrieval approaches perform retrieval determination and query construction based on the information generated by the model itself. Given a question, the model needs additional knowledge to answer if it is not confident, which can be determined by comparing the probability of the currently generated token y_t with a threshold θ. If $y_t < \theta$, the RAG framework will trigger retrieval at timestep t for more knowledge. Query construction is the problem of determining what to retrieve, i.e., a query $\mathbf{q}_r$ should be constructed to retrieve the most relevant knowledge from the knowledge base. The query is generally constructed based on the original question $\mathbf{q}$ and the already generated response $\mathbf{y}_{<t} = [y_1, y_2, \cdots, y_{t-1}]$ through a query construc-

tion function qry, denoted as $\mathbf{q}_r = \text{qry}(\mathbf{q}, \mathbf{y}_{<t})$.

3.2 Semantic Contribution-Aware Retrieval Determination

We propose a novel semantic contribution-aware retrieval determination method to address the problem "when to retrieve" in an RAG framework. It consists of 3 steps: (1) compute the word contribution, (2) scale the preset threshold based on the contribution, (3) compare the word probability with the threshold to determine whether to retrieve.

Word Contribution. As words differ in semantic contribution, their importance should be evaluated accordingly(Duan et al., 2024). We compute the contribution of a specific word using the leave-one-out method, which involves comparing the semantic change before and after removing the word.

SAR(Duan et al., 2024) estimated sentence uncertainty while we studied the feasibility of assigning threshold weights to individual tokens. Besides, we consider word-level instead of token-level contributions. Specifically, given a question $\mathbf{q}$ and a specific sentence $\mathbf{s}_t$ from response $\mathbf{y}$, we first remove word $w_{t,i}$ from $\mathbf{s}_t$, obtaining a corrupted response sentence $\mathbf{s}_t \backslash w_{t,i}$. We then compute the similarity between $[\mathbf{q}, \mathbf{s}_t]$ and $[\mathbf{q}, \mathbf{s}_t \backslash w_{t,i}]$ through an external cross-encoder model $f_{\text{x-enc}}$ (e.g., RoBERTa (Liu, 2019)), as shown in Eq. 1:

$$r(w_{t,i}; \mathbf{q}, \mathbf{s}_t) = 1 - f_{\text{x-enc}}\left([\mathbf{q}, \mathbf{s}_t], [\mathbf{q}, \mathbf{s}_t \backslash w_{t,i}]\right). \tag{1}$$

We treat it as the semantic contribution of $w_{t,i}$.

Threshold Scaling. The contribution $r(w_{t,i}; \mathbf{q}, \mathbf{s}_t)$ falls between 0 and 1. We need to normalize the overall sum of the weights to the length of sentence, so we normalize the contribution value along sentence $\mathbf{s}_t$, as shown in Eq. 2, which we denote it as semantic-contribution-weighting (SCW). A value lower or greater than 1 indicates that the contribution of the word is under or above average. Then, we scale the threshold for the specific word by the exponential of the contribution value, as shown in Eq. 3, where $\theta(w_{t,i}; \mathbf{q}, \mathbf{s}_t)$ denotes the original threshold (generally a constant value) of $w_{t,i}$.

$$r'(w_{t,i}; \mathbf{q}, \mathbf{s}_t) = \frac{|\mathbf{s}_t| \cdot r(w_{t,i}; \mathbf{q}, \mathbf{s}_t)}{\sum_{w_{t,i} \in \mathbf{s}_t} r(w_{t,i}; \mathbf{q}, \mathbf{s}_t)} \tag{2}$$

$$\theta_{\text{scaar}}(w_{t,i}; \mathbf{q}, \mathbf{s}_t) = e^{r(w_{t,i})} \cdot \theta(w_{t,i}; \mathbf{q}, \mathbf{s}_t) \tag{3}$$

Retrieval Determination. During generation, the probability of a word is computed as the product of the probabilities of all its tokens in Eq. 4:

$$P(\mathbf{w}_{t,i} | \mathcal{C}, \mathcal{W}, \mathbf{w}_{t,<i}) = \prod_{k=m}^{n} P(\mathbf{T}_{t,k} | \mathcal{C}, \mathcal{W}, \mathbf{T}_{t,<k}), \tag{4}$$

where $\mathbf{T}_{t,k}$ denotes the corresponding tokens that make up $\mathbf{w}_{t,i}$, m, n are the beginning and end indexes of a word, $\mathcal{W}$ is composed of question $\mathbf{q}$ and content generated previously $\mathbf{s}_{<\mathbf{t}}$. For the description of how to identify words from tokens,please refer to the Appendix B.

However, this computation results in lower probability values for words with more tokens. Therefore, we perform length normalization as shown in Eq. 5:

$$P'(\mathbf{w}_{t,i} | \mathcal{C}, \mathcal{W}, \mathbf{w}_{t,<i}) = P(\mathbf{w}_{t,i} | \mathcal{C}, \mathcal{W}, \mathbf{w}_{t,<i})^{\frac{1}{|w_{t,i}|}}, \tag{5}$$

Then, the normalized word probability is compared with the scaled word threshold. If the normalized probability of any word $w_{t,i}$ in the response sentence $\mathbf{s}_t$ is lower than the corresponding scaled threshold $\theta_{\text{scaar}}(w_{t,i}; \mathbf{q}, \mathbf{s}_t)$, the response sentence $\mathbf{s}_t$ should trigger retrieval.

By introducing an external cross-encoder model for word contribution computation, our retrieval determination approach can be generalized to black-box LLMs. The additional overhead introduced by the cross-encoder model is slight since it is generally a lightweight model compared to the LLM. For more details, please refer to Appendix G

3.3 Semantic Contribution-Aware Query Construction

To address the problem "what to retrieve", we propose a novel query construction approach based on the computed word contribution through α-percentile filtering policy, which we name it Percentile-Filtered Query (PFQ). Given the question $\mathbf{q}$, we say $\mathbf{s}_t$ is a hallucination sentence if retrieval is triggered at time step t. Given the hallucination sentence $\mathbf{s}_t = [w_{t,1}, w_{t,2}, \cdots, w_{t,n}]$, we sort the words in $\mathbf{s}_t$ by their semantic contribution decendingly and only keep the words with top $\alpha\%$ contribution values. The remaining words after α-percentile filtering may still contain hallucination words whose probabilities are below their thresholds. Therefore, we further remove the hallucination words and concatenate the question $\mathbf{q}$ with the remaining words to obtain the final query $\mathbf{q}_r$. The

complete algorithm of semantic contribution-aware query construction is shown in Algorithm 1. We denote the query as a function of the question and the response sentence, i.e., $\mathbf{q}_r = \text{qry}_{\text{scaar}}(\mathbf{q}, \mathbf{s_t})$.

Algorithm 1: Query construction

Data: Question $\mathbf{q}$, hallucination response sentence $\mathbf{s}_t$
Input: Percentage to keep α
Result: a constructed query $\mathbf{q}_r$

Sort $\mathbf{s}_t$ as $\mathbf{s}'_t$ descendingly of word contributions;
Let r_α be the α-percentile of contributions in $\mathbf{s}'_t$;
Initialize the query as the question: $\mathbf{q}_r \leftarrow \mathbf{q}$;
for $w_{t,i} \in \mathbf{s}'_t$ **do**
  $r_{t,i} \leftarrow r'(w_{t,i}; \mathbf{q}, \mathbf{s}_t)$;
  $\theta_{t,i} \leftarrow \theta_{\text{scaar}}(w_{t,i}; \mathbf{q}, \mathbf{s}_t)$;
  if $r_{t,i} > \theta_{t,i}$ ***and*** $r_{t,i} > r_\alpha$ **then**
    $\mathbf{q}_r \leftarrow \text{concat}(\mathbf{q}_r, w_{t,i})$;
  end
end
return $\mathbf{q}_r$

The α-percentile filtering policy provides a relative criterion to remove low-semantic-contributory words that may interfere with qualities of retrieval results. Intuitively, when confronted with unevenly distributed word semantics, the criterion based on α-percentile can better control the query length and quality compared to absolute filtering criteria. Like retrieval determination, the remaining high-semantic-contributory words are determined as hallucinated or not by comparing their generation probabilities with their adaptive thresholds, where higher-contributory words are assigned with higher thresholds, as shown in Eq. 3. This effectively addresses cases where the semantic contribution distribution of the remains has a large variance.

3.4 Generation Refinement

The SCAAR framework adopts a refinement idea of generating refinement with retrieved knowledge. Given the response sentence $\mathbf{s}_t$ generated from M, if $\mathbf{s}_t$ does not trigger retrieval, we use it as the output of timestep t. Otherwise, we perform query construction given question $\mathbf{q}$ and response sentence $\mathbf{s}_t$ to obtain the query $\text{qry}_{\text{scaar}}(\mathbf{q}, \mathbf{s}_t)$. Then, we retrieve the context knowledge $\mathcal{C}_t$ from knowledge base $\mathcal{D}$, denoted by Eq. 6. Finally, model M regenerate for a better response $\mathbf{s}'_t$ based on the context knowledge $\mathcal{C}_t$, the original question $\mathbf{q}$, and the outputs of previous timesteps $\mathbf{s}'_{<t}$, denoted by Eq. 7. Note that we only use the knowledge $\mathcal{C}_t$ retrieved at the current timestep t. The refined response sentence $\mathbf{s}'_t$ will replace the hallucination sentence $\mathbf{s}_t$.

$$\mathcal{C}_t \sim \mathcal{D}|_{\text{query}=\text{qry}_{\text{scaar}}(\mathbf{q},\mathbf{s}_t)} \quad (6)$$

$$\mathbf{s}'_t = M(\mathcal{C}_t, \mathbf{q}, \mathbf{s}'_{<t}) \quad (7)$$

4 Experiment

In this section, we first demonstrated and compared the performance of the SCAAR method with other baselines on the evaluation data, and then analyzed the effectiveness of different components in SCAAR through ablation studies.

4.1 Experiment Setup

Baselines. We compared SCAAR with methods including non-retrieval method (w/o RAG), fix-sentence RAG (FS-RAG) (Trivedi et al., 2022), which retrieves every sentence, alongside the adaptive retrieval methods FLARE (Jiang et al., 2023) and DRAGIN (Su et al., 2024). The original FLARE perform retrieval determination based on token-level probabilities. We adapted it to word-level by computing a geometric mean probability of all tokens in a word, in line with other methods.
Datasets. We tested on four open-source datasets: 2WikiMultiHopQA (Ho et al., 2020), HotpotQA (Yang et al., 2018), IIRC (Ferguson et al., 2020), and StrategyQA (Geva et al., 2021).
Evaluation Metrics. We randomly selected 300 samples from each dataset for evaluation. We incorporated Chain-of-Thought (Wei et al., 2022) and few-shot prompting (Brown et al., 2020) into the prompt to guide the model's reasoning process and generate correct answers for evaluation. The prompt we used is shown in Appendix A. For StrategyQA, we evaluated the exact match (EM) score since the answer is in "yes/no" format. For the other three datasets, we adopted both EM and F1 scores as evaluation metrics since the answers are phrases. Moreover, to evaluate the retrieval efficiency, we measured the average improvement brought by each retrieval. Given the average number of retrievals N_R and the improvement in F1 or EM score ΔS compared to the non-RAG baseline, the retrieval efficiency is computed as $S_{\text{eff}} = \Delta S / N_R$. We evaluated the efficiency in EM score improvement for StrategyQA and evaluated in F1 score improvement for other three.
Models. We utilized the instruct version of open-source Llama-2-7B, Llama-2-13B (Touvron et al., 2023), and Llama-3.1-8B (Dubey et al., 2024) for white-box evaluation. For SCAAR, these models

Table 2: Overall results on four datasets with the highest values of each model highlighted in bold are shown. The best performance of each method across all experimental settings is reported.

	2WikiMultiHopQA				HotpotQA				IIRC				StrategyQA		
	EM	F1	N_R	S_{eff}	EM	F1	N_R	S_{eff}	EM	F1	N_R	S_{eff}	EM	N_R	S_{eff}
Llama-2-13B															
w/o RAG	0.1658	0.2779	-	-	0.1623	0.2736	-	-	0.1111	0.1454	-	-	0.6710	-	-
FS-RAG	0.3389	0.4701	3.48	5.52	0.2500	0.3724	2.73	3.62	0.2291	0.2813	4.03	3.38	0.6667	4.22	-0.10
FLARE	0.3910	0.4912	2.71	**7.88**	0.3244	0.4339	3.80	4.22	0.2484	0.3078	3.98	**4.08**	0.6749	5.57	0.07
DRAGIN	0.3400	0.4637	2.65	7.01	**0.3415**	**0.4490**	3.16	**5.54**	0.2385	0.2806	3.75	3.61	0.7069	4.59	**0.78**
SCAAR (Ours)	**0.3918**	**0.4973**	3.14	6.99	0.3333	0.4369	3.39	4.81	**0.2490**	**0.3091**	4.20	3.90	**0.7090**	5.56	0.68
Llama-2-7B															
w/o RAG	0.2367	0.3099	-	-	0.2033	0.3158	-	-	0.1367	0.1665	-	-	0.6455	-	-
FS-RAG	0.2214	0.3106	2.48	0.03	0.1979	0.3014	1.74	-0.83	0.1483	0.1937	1.85	1.47	0.5933	3.49	-1.49
FLARE	0.2644	0.3509	2.31	1.78	0.2510	0.3628	2.34	2.01	**0.2000**	0.2358	1.82	3.81	0.6651	4.50	0.44
DRAGIN	0.2761	**0.3751**	2.86	2.28	0.2258	0.3310	1.69	0.90	0.1937	**0.2431**	1.95	**3.92**	0.6888	3.44	1.26
SCAAR (Ours)	**0.2778**	0.3677	2.36	**2.45**	**0.2680**	**0.3762**	1.69	**3.57**	0.1964	0.2361	1.92	3.63	**0.6944**	3.78	**1.29**
Llama-3-8B															
w/o RAG	0.3211	0.3907	-	-	0.2238	0.3354	-	-	0.2089	0.2500	-	-	0.7615	-	-
FS-RAG	0.4034	0.4950	4.05	2.57	0.3581	0.4661	3.25	4.02	0.2734	0.3223	3.92	1.84	0.7912	4.86	0.61
FLARE	0.5000	0.5812	3.09	6.16	0.4181	0.5347	3.27	6.10	0.2929	0.3496	3.27	3.05	0.7963	4.44	0.78
DRAGIN	0.3605	0.4236	0.77	4.28	0.2630	0.3761	1.07	3.81	0.1886	0.2120	1.58	-2.40	**0.8048**	1.38	**3.14**
SCAAR (Ours)	**0.5246**	**0.6026**	2.70	**7.84**	**0.4460**	**0.5570**	3.40	**6.52**	**0.3203**	**0.3694**	3.31	**3.60**	0.7799	4.35	0.42
GPT-4o															
w/o RAG	0.5452	0.6811	-	-	0.4407	0.5798	-	-	0.2972	0.3743	-	-	0.8490	-	-
FLARE	0.5556	0.6846	1.95	0.17	0.5272	0.6640	2.43	3.46	0.4000	0.4678	3.69	4.85	0.8941	2.57	1.70
SCAAR(Ours)	**0.5784**	**0.6980**	1.88	**0.89**	**0.5551**	**0.6794**	2.40	**4.15**	**0.4183**	**0.4965**	1.83	**6.67**	**0.8978**	2.58	**1.89**

were encapsulated into an API designed to simulate a black-box scenario. We also conducted a black-box method comparison of the GPT-4o (OpenAI, 2024) model with FLARE and SCAAR. We used the RoBERTa-large as our semantic encoder.

Knowledge Base and Retriever. We used Wikipedia (Karpukhin et al., 2020) as the external knowledge base, splitting the text into blocks of length 100 for retrieval. Each retrieval returned the top 3 documents most relevant to the question, using BM25 (Robertson et al., 2009).

For more details, refer to the Appendix A.

4.2 Overall Result Analysis

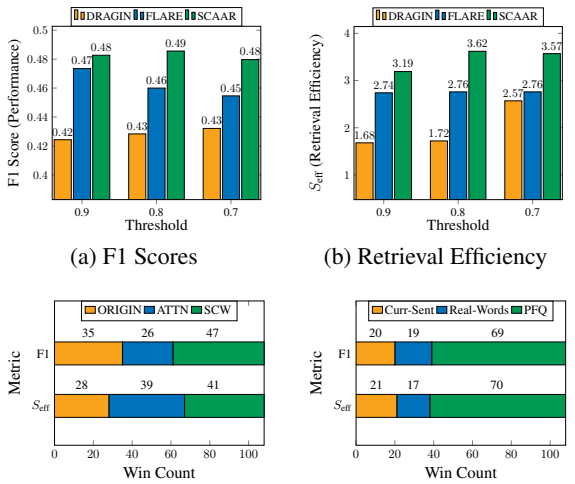
We compared SCAAR with baselines on evaluation data, as shown in Table 2, we found that: (1) Our SCAAR approach outperforms FLARE, and DRAGIN in most cases without models' internal states. It proves that our retrieval determiniation and query construction approach based on semantic contribution, effectively perceive the model's behavioral intentions and knowledge gaps, resulting in relevant retrievals. Experiments on GPT-4o demonstrate that SCAAR outperforms the static threshold black-box method FLARE by introducing dynamic thresholds. (2)FS-RAG underperforms adaptive retrieval methods and sometimes even underperforms the non-retrieval approach. This is because when retrieved content is similar to but irrelevant to the question, even if the model could inherently derive the correct answer, its over-reliance on context leads it to use incorrect information in reasoning and response. (3)DRAGIN fails to surpass FS-RAG with Llama-3.1-8B. We contribute it to the fact that model assigns higher probabilities to tokens, leading to fewer triggered retrievals compared to other models and the degraded performance. (4)Adaptive retrieval methods demonstrates significantly higher performance and retrieval efficiency compared to static methods, indicating that the adaptive retrieval determination based on model confidence works effectively.

Results of more methods and different granularities are in Appendix C, D.

For ablation, we combined the threshold weighting pipeline and query construction pipeline (three methods for each) of adaptive methods and evaluated over 3 models and 3 thresholds (0.9, 0.8, 0.7) on 4 datasets, resulting in 108 settings in total.

4.3 Initial Threshold Ablation

As shown in Equation 3, the variation of the initial threshold will alter the dynamic threshold, thereby affecting the final performance. Existing works only report best results over a range of initial thresholds of corresponding approaches, ignoring comparisons under a common initial threshold. We evaluate the performance of FLARE, DRAGIN, and SCAAR at initial threshold of 0.9, 0.8, and 0.7, 108 settings for each respectively. We believe



(a) F1 Scores (b) Retrieval Efficiency

(c) Comparison of different adaptive weighting methods. (d) Comparison of different query formulation methods.

Figure 2: Comparison under same initial thresholds and Win count of adaptive weighting methods and query formulation methods.

that an excessively low initial threshold has little practical significance. We reported the mean performance of different methods at different thresholds in Figure 2a and 2b. Different thresholds result in different generation performance (F1 score) and retrieval efficiencies (S_{eff}), but SCAAR consistently outperforms FLARE and DRAGIN in both generation performance and retrieval efficiency under all threshold configurations.

4.4 Adpative Weight and Query Formulation

In SCAAR, the SCW method determines the thresholds for each words, and the PFQ formulation method constructs queries for retrieval. We evaluated these two pipelines on four datasets using the Llama-2-7B model. We replaced SCW with ORIGIN and ATTN, where ORIGIN assigns a weight of "1" to all words and ATTN computes weights of words based on attention scores. As for the latter, we replace it with Curr-Sent and Real-Words, where Curr-Sent directly uses the high-confidence words in current sentence as the query and Real-Words uses real words. As shown in Table 3, Under various weighting methods, our PFQ achieves the best performance compared to Curr-Sent and Real-Words in most cases. However, we cannot infer which combination of adaptive weighting method and query formulation method achieves best performance (i.e., having the most underlined scores) from Table 3, since 4 out of 9 combinations achieve the best performance on at least one task.

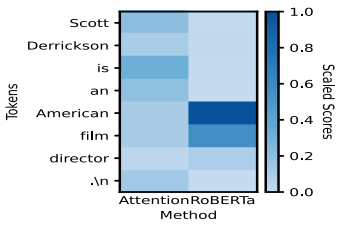


Figure 3: Visualization of word significance for answer of "Were Scott Derrickson and Ed Wood of the same nationality?" in ATTN and SCW.

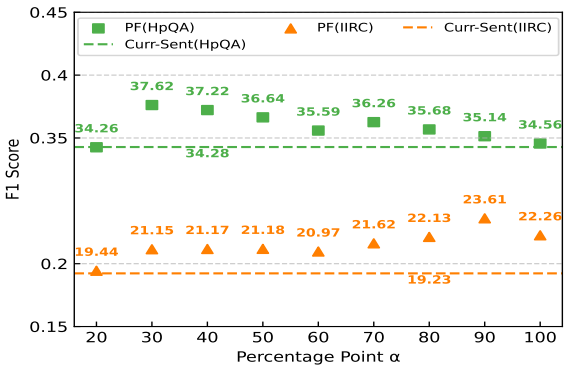


Figure 4: F1 scores of PFQ with different α against Curr-Sent pipeline. We set threshold to be 0.9.

Different from experiments in Table 3, we counted the number of times each weighting method achieves the best performance and efficiency given the specific query formulation method across 108 configurations and reported results in Figure 2c. SCW achieves the highest win count in F1 and S_{eff} scores. Besides, results of query formulations in Figure 2d show PFQ achieves the highest win count in F1 and S_{eff} scores.

To more intuitively analyze the difference between ATTN and SCW, we visualize the word significance computed by the two methods. Given a specific question, the first sentence of the response is “Scott Derrickson is an American film director.” with 7 words. Figure 3 demonstrates the significance score of each word computed by ATTN and SCW, where SCW effectively captures “American” and “film”, the two words that contributes most to the semantics of the sentence. These two words are indeed potential hallucinations since they describe some factual and knowledgeable content, therefore need to be assigned with a stricter threshold.

4.5 Percentile Ablation

In PFQ, we keep words with top $\alpha\%$ contribution values. To clarify the influence of α, we perform ablation experiment on Llama-2-7B model with

Table 3: Best results of Llama-2-7B across various thresholds. The bold values indicates the best query formulation method, and the underlined values indicates the best combination of weighting method and query formulation methods. Note that we only controlled the pipeline variable and did not control any other variables.

Weighting	Query	2WikiMultiHopQA			HotpotQA			IIRC			StrategyQA	
		EM	F1	S_{eff}	EM	F1	S_{eff}	EM	F1	S_{eff}	EM	S_{eff}
ORIGIN	Curr-Sent	0.2644	0.3509	1.78	0.2510	0.3628	2.01	0.2000	0.2358	3.81	0.6651	0.44
ORIGIN	Real-Words	0.2534	0.3434	1.44	**0.2696**	**0.3693**	**2.93**	0.1952	0.2432	3.08	0.6632	0.43
ORIGIN	PFQ (Ours)	**0.2838**	**0.3707**	**2.48**	0.2625	0.3544	1.73	**0.2218**	**0.2576**	**4.35**	**0.6986**	**1.22**
ATTN	Curr-Sent	0.2795	0.3675	2.02	0.2198	0.3357	1.19	0.1918	0.2370	3.26	0.6429	-0.07
ATTN	Real-Words	0.2761	0.3751	2.28	0.2258	0.3310	0.90	0.1937	0.2431	3.92	0.6118	-0.93
ATTN	PFQ (Ours)	**0.3014**	**0.3787**	**2.41**	**0.2313**	**0.3471**	**1.86**	**0.2082**	**0.2520**	**4.48**	**0.6485**	**0.08**
SCW (Ours)	Curr-Sent	0.2664	0.3562	1.93	0.2556	0.3505	2.05	0.1906	0.2234	3.11	0.6844	0.96
SCW (Ours)	Real-Words	0.2525	0.3425	1.58	0.2609	0.3532	2.09	0.1713	0.2239	2.57	0.6655	0.41
SCW (Ours)	PFQ (Ours)	**0.2778**	**0.3677**	**2.45**	**0.2680**	**0.3762**	**3.57**	**0.1964**	**0.2361**	**3.63**	**0.6944**	**1.29**

Table 4: EM score of Llama-2-7B-chat on 2WikimultihopQA with different amounts of documents.

Method	Number of Documents			
	2	3	4	5
FLARE	0.2391	0.2644	0.2383	0.2375
DRAGIN	0.2742	0.2761	0.2341	0.2609
SCAAR	**0.2755**	**0.2778**	**0.2508**	**0.2752**

Table 5: Performance on Llama2-7B-chat over datasets with DPR.

Dataset	Method	EM	F1	S_{eff}
HotpotQA	FLARE	0.2068	0.2695	-2.94
	DRAGIN	0.1773	0.2678	-3.15
	SCAAR	**0.2162**	**0.3245**	**0.56**
IIRC	FLARE	0.1204	0.1373	-1.36
	DRAGIN	0.1313	0.1663	-0.01
	SCAAR	**0.1370**	**0.1689**	**0.13**
StrategyQA	FLARE	0.6469	0.6469	0.03
	DRAGIN	0.6566	0.6566	0.31
	SCAAR	**0.6763**	**0.6763**	**0.65**

different α values and the same weighting method. Results in Figure 4 show that for HotpotQA and IIRC, mostly α values outperform the Curr-Sent approach with SCW. Even with small α values, the filter effectively eliminates extraneous information from current generations while maintaining focus on important information. We find that for different datasets, the optimal α varies, and the results on either side of the peak decay as the distance from the peak increases.

4.6 Impact of Encoder

We replaced RoBERTa with DeBERTa (He et al.) to analyze the impact of the encoder on the framework's performance. Results of Llama-3-8B on 2WikimultihopQA and HotpotQA are shown in the Appendix H. DeBERTa outperforms RoBERTa, we attributed it to its better performance in semantic similarity calculation.

4.7 Impact of Num of Documents

To compare performance as the number of documents changes, we vary the number from 2 to 5 (performance remains stable when it exceeds 5). Results of Llama-2-7B on the 2WikimultihopQA are presented in Table 4. The best performance is achieved when the number is set to 3. Across all experiments, our SCAAR outperform FLARE and DRAGIN, demonstrating the effectiveness. Additional results are provided in the Appendix F.

4.8 Impact of Retriever

There are two types retrieval: lexical matching and dense retrieval. We also employed the DPR model (Karpukhin et al., 2020) as dense retriever and conducted tests on the Llama2-7B model. For more detail, please refer to Appendix E. Results as shown in Table 5, indicate that the SCAAR scheme outperforms the FLARE and DRAGIN across test datasets with the DPR model. However, the performance of three methods is lower than that of the BM25-based retriever and baseline methods even underperform the non-retrieval method on the HotpotQA and IIRC. We assume that this is due to the lack of semantics brought by the short length of the following sentence.

5 Conclusion

In this paper, we propose an adaptive RAG framework incorporating a dynamic weight adjustment mechanism based on semantic contribution and a percentile-filtered query construction method for black-box scenarios. Extensive experiments demonstrate the effectiveness of our framework. Furthermore, ablation study results show the contributions of individual pipeline components to the enhanced performance.

6 Limitations

We acknowledge that there remains significant room for enhancement on the following directions:

- **Enhancing Semantic Weight Representativeness.** As SCAAR with DeBERTa-base surpasses that with RoBERTa-large, domain-specific fine-tuning of the encoder during application may strengthen the representativeness of the weight coefficients
- **Learnable Quantile Filtering.** Our percentile filtering method relies on heuristic constants. We argue that training a classifier for percentile prediction is a necessary step
- **Optimizing Dense Passage Retrieval.** Experiment results indicate that dpr still has substantial potential for improvement. A key challenge in adaptive retrieval scenarios is capturing the semantics of up-coming sentences with limited word counts
- **LLM-based metrics.** We ensured the feasibility of using EM and F1 as evaluation metrics by controlling the output format of the model. However, it would be valuable to include experimental results using LLM-based metrics to assess retrieval accuracy more comprehensively.

7 Ethics Statement

In our research and experimental endeavors, we adhere strictly to ethical guidelines to ensure that our development and application of artificial intelligence technology are conducted responsibly. Throughout our research process, we have refrained from utilizing data that relies on personal information or manual annotations. Moreover, we have employed open-source models for our experiments without any additional training, thereby ensuring that we do not introduce bias or other harmful knowledge into them. In addition, we have made our code and data publicly available on the GitHub community. This allows the community to verify the performance of our proposed method and to further enhance and optimize it.

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*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

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

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.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 35:18878–18891.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. Inside: Llms' internal states retain the power of hallucination detection. *arXiv preprint arXiv:2402.03744*.

Ondřej Cífka and Antoine Liutkus. 2023. Black-box language model explanation by context length probing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, page 1067–1079. Association for Computational Linguistics.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063.

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

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

James Ferguson, Matt Gardner, Hannaneh Hajishirzi, Tushar Khot, and Pradeep Dasigi. 2020. Iirc: A dataset of incomplete information reading comprehension questions. *arXiv preprint arXiv:2011.07127*.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 1(2):4.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1. Minneapolis, Minnesota.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

Xiaojun Kuang, CL Philip Chen, Shuzhen Li, and Tong Zhang. 2024. Multi-scale prompt memory-augmented model for black-box scenarios. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1743–1757.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

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.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jingyuan Wang, Jian-Yun Nie, and Ji-Rong Wen. 2023. The web can be your oyster for improving language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 728–746.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Hongyin Luo, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. Sail: Search-augmented instruction learning. *arXiv preprint arXiv:2305.15225*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

OpenAI. 2022. Introducing chatgpt. https://openai.com/index/chatgpt/.

OpenAI. 2024. Gpt-4o. Accessed: 2024-08-01.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

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

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. Dragin: Dynamic retrieval augmented generation based on the real-time information needs of large language models. *arXiv preprint arXiv:2403.10081*.

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023. Learning to filter context for retrieval-augmented generation. *arXiv preprint arXiv:2311.08377*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Weichuan Liu, Lei Hou, and Juanzi Li. 2024. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. *arXiv preprint arXiv:2406.19215*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

A More Details about Experiment Setup

Datasets. We test on four knowledge-intensive datasets: 2WikiMultiHopQA (Ho et al., 2020), HotpotQA (Yang et al., 2018), IIRC (Ferguson et al., 2020), and StrategyQA (Geva et al., 2021).

2WikimultihopQA. A multi-hop question answering dataset designed to advance complex reasoning tasks, especially multi-step reasoning tasks. The dataset contains about 20,000 questions that involve a large number of reasoning steps and information synthesis tasks. Each question has multiple candidate answers, and the model needs to select the correct answer from them.

HotpotQA. A large multi-hop question answering dataset designed to advance the ability of machines to understand complex questions. The dataset contains 113,000 questions, which are characterized by the fact that it contains questions that require multi-step reasoning and information across multiple documents to answer, requiring the model to not only extract information from a single article, but also conduct comprehensive analysis across multiple documents. The answer to a question in HotpotQA is usually a short entity (such as a person's name, a place name, etc.) or a concise fact.

IIRC. The IIRC dataset is a collection of incomplete information reading comprehension questions. It comprises 13,441 questions based on 5,698 paragraphs sourced from English Wikipedia. These questions were crafted by crowdworkers who had no access to any linked documents. As a result, the contexts in which the questions and answers appear exhibit minimal lexical overlap. This unique approach not only makes the dataset more reflective of real-world information-seeking scenarios but also significantly increases the complexity of the task. Many questions in the dataset are either

unanswerable or require discrete reasoning, posing substantial challenges for models attempting to navigate and retrieve information from multiple sources.

StrategyQA. A dataset comprises 2,780 meticulously crafted samples, each encompassing a strategic policy question, its detailed decomposition steps, and a corresponding evidence paragraph. Utilizing a robust crowdsourcing pipeline, the dataset employs terminology guidance to inspire annotators, enforces strict control over the annotator group, and implements adversarial filtering to eliminate reasoning shortcuts. This comprehensive approach ensures the questions are both creative and challenging, demanding implicit reasoning steps that are not explicitly stated within the questions themselves.

HotpotQA and 2WikiMultihopQA are multi-hop reasoning datasets where models need to extract information from multiple documents to answer questions through basic analysis. IIRC is a conversational dataset that presents greater challenges than HotpotQA and 2WikiMultihopQA, as models must not only acquire document information but also understand and execute instruction-based interactions. StrategyQA aims to evaluate and enhance models' ability to solve problems requiring strategic thinking and reasoning, where models must combine textual information with common sense and logical inference.

Prompt Settings. The few-shots COT prompt we use in experiments are as shown:

```
[1] Context 1
[2] Context 2
...
[N] Context N
Answer the question by reasoning step-by-step and response
result with "So the answer is " format.
Question: Q1
Answer: A1
...
Question: Qn
Answer: An
Question: <<<the question to be evaluated>>>
Answer:
```

Knowledge Base and Retriever. We use Wikipedia (Karpukhin et al., 2020) as the external knowledge base, which contains various topics and information to support us to obtain the context knowledge relevant to test questions. There are 21,015,324 passages in the database which is sufficient for assisting models to answer questions. We employ BM25 (Robertson et al., 2009), which as the retriever following FLARE and most existing works.

Table 6: Average performance of word-level and token-level thresholding with different models.

Model	Word-Level			Token-Level		
	EM	F1	S_{eff}	EM	F1	S_{eff}
Llama-2-13B	0.3890	0.4599	3.54	0.3886	0.4579	3.65
Llama-2-7B	0.3242	0.3840	1.20	0.3203	0.3787	1.02
Llama-3-8B	0.4874	0.5559	3.47	0.4845	0.5539	3.41
Overall	0.4002	0.4666	2.73	0.3978	0.4635	2.69

B Tokens identification

Our token-to-word alignment process operates as follows: First, we decode the full token sequence to obtain a complete word list. We then employ an adaptive window sizing mechanism to progressively expand a sliding window over the token sequence. When the decoded result of substring matches any lexical item in the remaining list, we register the corresponding token span, reset the window and move the beginning to the end of span. This iterative process continues until full sequence coverage is achieved. The pseudocode below formalizes this procedure:

Algorithm 2: Word Alignment Algorithm

Initialize two empty lists: A: temporary token buffer, WL: word alignment list
Initialize index $i1 = 0$, $i2 = 0$
Initialize list $W = \text{Decode}(T)$ where T is the output token sequence
Initialize $L1 = \text{len}(W)$, $L2 = \text{len}(T)$
while $i2 < L2$ **do**
  Append T_{i2} to list A **if** $Decode(A) == W_{i1}$ **then**
    Append $[A[0], A[-1]]$ to WL
    Reset A to empty list
    Update $i2 = i2 + 1$
  end
  Update $i1 = i1 + 1$
end
After loop, append $[A[0], A[-1]]$ to WL

C Comparison of single-round RAG and fix-length RAG

In the experiment section, limited by page length, we mainly compare our method with other adaptive methods, so we show all comparison results between adaptive methods and static methods here, including single-round RAG (Lewis et al., 2020) and fix-length RAG (Ram et al., 2023).

Table 7: Overall results of SCAAR and baselines on four datasets.

	2WikiMultiHopQA				HotpotQA				IIRC				StrategyQA		
	EM	F1	N_R	S_{eff}	EM	F1	N_R	S_{eff}	EM	F1	N_R	S_{eff}	EM	N_R	S_{eff}
Llama-2-13B															
w/o RAG	0.1658	0.2779	-	-	0.1623	0.2736	-	-	0.1111	0.1454	-	-	0.6710	-	-
SR-RAG	0.1971	0.3451	1.00	6.72	0.2838	0.4016	1.00	12.80	0.1711	0.2173	1.00	7.19	0.6750	1.00	0.40
FL-RAG	0.2535	0.3674	2.06	4.35	0.2947	0.4151	3.42	4.14	0.1711	0.2314	2.81	3.06	0.6643	5.34	-0.13
FS-RAG	0.3389	0.4701	3.48	5.52	0.2500	0.3724	2.73	3.62	0.2291	0.2813	4.03	3.38	0.6667	4.22	-0.10
FLARE	0.3910	0.4912	2.71	**7.88**	0.3244	0.4339	3.80	4.22	0.2484	0.3078	3.98	**4.08**	0.6749	5.57	0.07
DRAGIN	0.3400	0.4637	2.65	7.01	**0.3415**	**0.4490**	3.16	**5.54**	0.2385	0.2806	3.75	3.61	0.7069	4.59	**0.78**
SCAAR (Ours)	**0.3918**	**0.4973**	3.14	6.99	0.3333	0.4369	3.39	4.81	**0.2490**	**0.3091**	4.20	3.90	**0.7090**	5.56	0.68
Llama-2-7B															
w/o RAG	0.2367	0.3099	-	-	0.2033	0.3158	-	-	0.1367	0.1665	-	-	0.6455	-	-
SR-RAG	0.1945	0.2920	1.00	-1.79	0.1466	0.2427	1.00	-7.31	0.1672	0.2250	1.00	5.85	0.6230	1.00	-2.25
FL-RAG	0.1620	0.2608	1.56	-3.15	0.1554	0.2573	1.18	-4.95	0.1418	0.1865	1.06	1.89	0.6421	1.61	-0.21
FS-RAG	0.2214	0.3106	2.48	0.03	0.1979	0.3014	1.74	-0.83	0.1483	0.1937	1.85	1.47	0.5933	3.49	-1.49
FLARE	0.2644	0.3509	2.31	1.78	0.2510	0.3628	2.34	2.01	**0.2000**	0.2358	1.82	3.81	0.6651	4.50	0.44
DRAGIN	0.2761	**0.3751**	2.86	2.28	0.2258	0.3310	1.69	0.90	0.1937	**0.2431**	1.95	**3.92**	0.6888	3.44	1.26
SCAAR (Ours)	**0.2778**	0.3677	2.36	**2.45**	**0.2680**	**0.3762**	1.69	**3.57**	0.1964	0.2361	1.92	3.63	**0.6944**	3.78	**1.29**
Llama-3-8B															
w/o RAG	0.3211	0.3907	-	-	0.2238	0.3354	-	-	0.2089	0.2500	-	-	0.7615	-	-
SR-RAG	0.3115	0.4193	1.00	2.86	0.3345	0.4640	1.00	12.86	0.2641	0.3377	1.00	8.77	0.7249	1.00	-3.66
FL-RAG	0.3684	0.4679	1.94	3.97	0.3825	0.4903	1.95	7.94	0.2918	0.3337	2.20	3.81	0.7181	2.19	-1.98
FS-RAG	0.4034	0.4950	4.05	2.57	0.3581	0.4661	3.25	4.02	0.2734	0.3223	3.92	1.84	0.7912	4.86	0.61
FLARE	0.5000	0.5812	3.09	6.16	0.4181	0.5347	3.27	6.10	0.2929	0.3496	3.27	3.05	0.7963	4.44	**0.78**
DRAGIN	0.3605	0.4236	0.77	4.28	0.2630	0.3761	1.07	3.81	0.1886	0.2120	1.58	-2.40	**0.8048**	1.38	3.14
SCAAR (Ours)	**0.5246**	**0.6026**	2.70	**7.84**	**0.4460**	**0.5570**	3.40	**6.52**	**0.3203**	**0.3694**	3.31	**3.60**	0.7799	4.35	0.42

In all cases, the static retrieval schemes' final performance falls short of ours, and in most instances, it also lags behind the dynamic schemes'. It is noteworthy that, in some scenarios, the single-round scheme boasts the highest retrieval efficiency among all schemes. For example, on the HotpotQA dataset, the Llama2-13B-chat and Llama3.1-8B-chat models exhibit superior efficiency. We posit that this finding underscores the strong correlation between retrieval efficiency and both the model and the question scenario. Therefore, it is imperative to integrate an adaptive scheme that leverages the model's internal knowledge with external knowledge, such as question difficulty and type, as the basis for triggering retrieval. Additionally, we observe that our retrieval efficiency index declines as the reasoning length increases. Hence, developing a more comprehensive retrieval efficiency evaluation index represents a promising direction for future research.

D Comparison of Different Granularity

We analyze the impact of different configurations in the SCAAR framework on performance through ablation studies. SCAAR computes the semantic-based adaptive weights at word level to ensure sementic integrity and generation efficiency. Intuitively, using the word-level probability may hinder the distinctness of the token probability to a certain extent. Specifically, if the initial threshold is 0.8, and the probabilities of the two tokens that make up the word are 0.7 and 1.0 respectively. At token-level, it will trigger retrieval since the probability of the first token 0.7 is lower than the threshold 0.8. However, at word-level, it will not trigger retrieval since the word probability is the geometric mean of 0.8 and 1.0, i.e., 0.83, which is greater than the threshold 0.8. To clarify the impact of different thresholding granularities, we evaluate the performance of using token-level and word-level thresholding under the vanilla RAG framework with a fixed threshold. The average performance over the aforementioned four datasets on different models is shown in Table 6, where overall indicates the average scores over all three models. The results shows that word-level thresholding slightly outperforms token-level thresholding in EM and F1 scores overl all model configurations.

E DPR Model Settings

In order to test our method in a dense passage retrieval senarior, we choose the encoder released by Karpukhin et al.(Karpukhin et al., 2020). The question encoder and text encoder used in our experiments use the BERT-base (Kenton and Toutanova, 2019) as backbones and are further trained on Natural Questions (NQ) dataset (Lee et al., 2019; Kwiatkowski et al., 2019). For a question, we ob-

tain a dense embedding of the special token [CLS] which is obtained by applying a linear transformation followed by a tanh activation function to the hidden state of the [CLS] token from the last layer. We used Faiss, a vector database, to load pre-encoded external knowledge. Then, we utilized full-precision indexing based on L2 (Euclidean distance) for matching. This approach is faster than using cosine similarity for calculations, though it may result in a slight loss of accuracy.

F Comparison of Different Num of Documents

We conduct experiments on baseline methods and SCAAR methods using different num of retrieved documents. We pick [3, 5, 7] for Llama-3.1-8B and Llama-2-13B, and pick [2,3,4,5,7] for Llama-2-7B. Results are shown in Table 10, 11, 12 respectively. We can draw several conclusions: (1)In all experiments, the setting of doc_num=3 yields the best results in most cases. Having too many or too few retrieved documents may interfere with the model's reasoning ability and cause errors. (2)There is no consistently obvious relationship between the number of documents and performance across all models. We believe this is due to the fixed retrieval number scheme lacking post-retrieval assessment of the quality of retrieved documents. This inspires us to further verify the quality of retrieved documents or the answers generated before and after model retrieval. (3)In most experimental settings, our SCAAR scheme can surpass DRAGIN to achieve the best performance, further proving that our scheme is not only suitable for black-box scenarios but also has performance advantages in white-box scenarios.

Table 8: The actual time required for SCAAR and DRAGIN methods to complete reasoning on 300 2WikimultihopQA questions on Llama3-8B and Llama2-13B.

model	DRAGIN	SCAAR
Llama3-8B	40min	60min
Llama2-13B	150min	200min

G Additional overhead

We estimated the contribution of each word before we adjusted the thresholds of every words, therefore it is important to consider the computational efficiency of our method. Our method does not involve model fine-tuning, so we only compare with self-consistency-based methods and probability-based methods.

Table 9: Comparisons of using different cross-encoders. The bold results are the best among experiments. R for RoBERTa, and D for DeBERTa

Encoder	2WikimultihopQA		HotpotQA		StrategyQA
	EM	F1	EM	F1	F1
RoBERTa	0.5246	0.6026	0.4460	0.5570	0.7799
DeBERTa	**0.5387**	**0.6083**	**0.4484**	**0.5589**	**0.7989**

G.1 Compare with self-consistency-based method

Assuming that we have a reasoning model of size N B and an cross-encoder model of size M B, and that the average length of the answer to question Q is L. For a self-consistency scheme on this model with n samples, the total overhead can be considered as $N * L * n$. In contrast, our approach involves an overhead of $N * L * 1 + M * L * 1$. The ratio of the overheads is $N * n : N + M$. Since $M < N$ and $n \geq 2$, our approach has less overhead compared to the multi-sampling black-box scheme.

G.2 Compare with Dragin

In our experiments, we used a RoBERTa-large model of size 340M to complete the calculation of semantic contribution.

The resource overhead ratio of DRAGIN:SCAAR is: $2 * N : 2 * N + 0.7$ if all models are loaded in bf16.

The actual running time (in minutes) of the DRAGIN and SCAAR schemes on the 2WikimultihopQA dataset is shown in Table 8.

The additional time overhead is about 1/2 of DRAGIN's for the 8B model and about 1/3 for the 13B model. Notably, as the inference model becomes larger, the relative overhead of our auxiliary model diminishes.

H Impact of Semantic Encoder

We evaluated the impact on performance when using other pre-trained cross-encoders for semantic contribution calculation. We used deberta-v3-base as model as the encoder and the Llama-3-8B chat model as the reasoning model, conducting experiments on 2WikimultihopQA and HotpotQA. Results are shown in Table 9. Using two different Encoders can both achieve performance surpassing the baseline on the test set, which demonstrates

Table 10: Ablation results of doc_num for comparison of different methods on Llama-2-13B, 4 datasets. We bold the best result of each method under the dataset. When the results of different doc_num are the same, we bold the result with fewer doc_num. We denote the best result on each dataset with an asterisk.

method	**doc_num**	2WikiMultiHopQA				HotpotQA				IIRC				StrategyQA		
		EM	**F1**	N_R	S_{eff}	**EM**	**F1**	N_R	S_{eff}	**EM**	**F1**	N_R	S_{eff}	**F1**	N_R	S_{eff}
w/o RAG	0	0.1658	0.2779	0	0.00	0.1623	0.2736	0	0.00	0.1111	0.1454	0	0.00	0.6710	0	0.00
FLARE	3	**0.3910**	**0.4912**	2.71	**7.88***	**0.3244**	**0.4339**	3.80	**4.22**	0.2484	0.3078	3.98	4.08	0.6749	5.57	0.07
	5	0.3664	0.4835	2.90	7.10	0.2984	0.4172	3.85	3.73	**0.2744***	**0.3356**	4.25	**4.47**	**0.6846**	4.92	**0.28**
	7	0.3664	0.4835	2.90	7.10	0.2984	0.4172	3.85	3.73	0.2744	0.3356	4.25	4.47	0.6846	4.92	0.28
DRAGIN	3	**0.3400**	**0.4637**	2.65	7.01	**0.3415**	**0.4490**	3.16	5.54	0.2385	0.2806	3.75	3.74	**0.7069**	4.59	**0.78***
	5	0.3200	0.4384	2.24	**7.17**	0.3088	0.4187	2.37	**6.12***	**0.2586**	**0.3131**	3.73	**4.50***	0.6937	5.12	0.44
	7	0.3200	0.4384	2.24	7.17	0.3088	0.4187	2.37	6.12	0.2586	0.3131	3.73	4.50	0.6937	5.12	0.44
SCAAR	3	**0.3918***	0.4973	3.14	6.99	0.3333	0.4369	3.39	4.81	0.2490	0.3091	4.20	3.90	**0.7090***	5.56	**0.68**
	5	0.3870	**0.5037***	3.20	**7.07**	**0.3674***	**0.4639***	3.33	**5.71**	**0.2612**	**0.3276**	4.19	**4.34**	0.7024	5.04	0.62
	7	0.3870	0.5037	3.20	7.07	0.3674	0.4639	3.33	5.71	0.2612	0.3276	4.19	4.34	0.7024	5.04	0.62

Table 11: Ablation results of doc_num for comparison of different methods on Llama-2-7B, 4 datasets. We bold the best result of each method under the dataset. When the results of different doc_num are the same, we bold the result with fewer doc_num. We denote the best result on each dataset with an asterisk.

method	**doc_num**	2WikiMultiHopQA				HotpotQA				IIRC				StrategyQA		
		EM	**F1**	N_R	S_{eff}	**EM**	**F1**	N_R	S_{eff}	**EM**	**F1**	N_R	S_{eff}	**F1**	N_R	S_{eff}
w/o RAG	0	0.2367	0.3099	0	0.00	0.2033	0.3158	0	0.00	0.1367	0.1665	0	0.00	0.6455	0	0.00
FLARE	2	0.2391	0.3280	2.33	0.78	0.2730	0.3736	1.94	2.98	0.1690	0.1979	2.29	1.37	0.6421	5.59	0.00
	3	**0.2644**	**0.3509**	2.31	**1.78**	0.2510	0.3628	2.34	2.01	**0.2000***	**0.2358**	1.82	**3.05**	0.6651	4.50	0.44
	4	0.2383	0.3166	1.55	0.43	**0.2886***	**0.3780**	1.53	4.07	0.1831	0.2193	1.80	2.94	**0.6678**	3.59	**0.62**
	5	0.2375	0.3326	1.64	1.38	0.2635	0.3767	1.47	**4.15**	0.1684	0.2056	1.91	2.05	0.6531	4.83	0.16
	7	0.2375	0.3326	1.56	1.46	0.2685	0.3709	1.34	4.12	0.1684	0.2056	1.91	2.05	0.6531	4.83	0.16
DRAGIN	2	0.2742	0.3657	2.40	2.33	0.2575	0.3603	1.75	2.54	0.1800	0.2185	2.46	2.11	0.6296	3.88	-0.04
	3	**0.2761**	**0.3751***	2.86	2.28	0.2258	0.3310	1.69	0.90	**0.1937**	**0.2431**	1.95	**3.92***	**0.6888**	3.44	**0.13**
	4	0.2341	0.3387	1.77	1.63	0.2609	0.3489	1.38	2.40	0.1911	0.2379	2.13	3.35	0.6576	4.01	0.03
	5	0.2609	0.3505	1.56	**2.61**	**0.2676**	**0.3545**	1.37	**2.82**	0.1886	0.2375	2.38	2.99	0.6712	3.32	0.08
	7	0.2609	0.3505	1.56	2.61	0.2676	0.3545	1.37	2.82	0.1886	0.2375	2.38	2.99	0.6712	3.32	0.08
SCAAR	2	0.2755	0.3627	2.48	2.12	0.2709	0.3652	1.76	2.81	0.1706	0.2066	2.19	1.83	0.6679	5.29	0.04
	3	**0.2778***	**0.3677**	2.36	2.45	0.2680	0.3762	1.69	3.57	**0.1964**	**0.2361***	1.92	**3.63**	**0.6944***	3.78	**1.3***
	4	0.2508	0.3239	1.54	0.91	0.2727	0.3814	1.40	4.68	0.1757	0.2246	1.91	3.05	0.6713	4.55	0.06
	5	0.2752	0.3626	1.41	**3.75***	0.2635	0.3525	1.38	2.66	0.1741	0.2126	1.89	2.43	0.6761	4.49	0.07
	7	0.2752	0.3626	1.41	3.75	**0.2852**	**0.3828***	1.23	**5.43***	0.1741	0.2126	1.89	2.43	0.6761	4.49	0.07

Table 12: Ablation results of doc_num for comparison of different methods on Llama-3.1-8B, 4 datasets. We bold the best result of each method under the dataset. When the results of different doc_num are the same, we bold the result with fewer doc_num. We denote the best result on each dataset with an asterisk.

method	**doc_num**	2WikiMultiHopQA				HotpotQA				IIRC				StrategyQA		
		EM	**F1**	N_R	S_{eff}	**EM**	**F1**	N_R	S_{eff}	**EM**	**F1**	N_R	S_{eff}	**F1**	N_R	S_{eff}
w/o rag	0	0.3211	0.3907	0	0.00	0.2238	0.3354	0	0.00	0.2089	0.2500	0	0.00	0.7615	0	0.00
FLARE	3	**0.5000**	**0.5812**	3.09	**6.16**	0.4181	**0.5347**	3.27	**6.10**	0.2929	0.3496	3.27	3.05	0.7963	4.44	**0.08**
	5	0.4680	0.5693	3.34	5.35	**0.4225**	0.5344	3.60	5.53	**0.3536**	**0.3940**	3.93	**3.67**	**0.7951**	5.06	0.07
	7	0.4680	0.5693	3.34	5.35	0.4225	0.5344	1.78	11.19	0.3536	0.3940	3.93	3.67	0.7951	5.06	0.07
DRAGIN	3	**0.3605**	**0.4236**	0.77	**4.28**	**0.2630**	**0.3761**	1.07	**3.81**	0.1886	0.2120	1.58	-2.40	**0.8048***	1.38	0.31
	5	0.3311	0.4062	0.87	1.78	0.2571	0.3667	1.78	1.76	**0.2359**	**0.2593**	2.05	**0.45**	0.7759	2.08	**0.69***
	7	0.3311	0.4062	0.87	1.78	0.2571	0.3667	3.60	0.87	0.2359	0.2593	2.05	0.45	0.7759	2.08	0.69
SCAAR	3	**0.5246***	**0.6026***	2.70	**7.84***	**0.4460***	**0.5570***	3.40	**6.52***	0.3203	0.3694	3.31	3.60	**0.7799**	4.35	**0.04**
	5	0.4880	0.5729	3.27	5.58	0.4240	0.5412	3.35	6.14	**0.3759***	**0.4279***	3.57	**4.99***	0.7705	4.73	0.02
	7	0.4880	0.5729	3.27	5.58	0.4456	0.5632	3.53	6.45	0.3759	0.4279	3.57	4.99	0.7705	4.73	0.02

the effectiveness of the dynamic weighting strategy based on semantic contribution. Since DeBERTa outperforms RoBERTa in semantic similarity calculation, the experimental results using DeBERTa are also superior to those using RoBERTa.