Invoke Interfaces Only When Needed: Adaptive Invocation for Large Language Models in Question Answering

Jihao Zhao[1] Chunlai Zhou[1] Daixuan Li[1] Shuaishuai Zu[1] Biao Qin[1]*
[1]School of Information, Renmin University of China, Beijing, China

Abstract

The collaborative paradigm of large and small language models (LMs) effectively balances performance and cost, yet its pivotal challenge lies in precisely pinpointing the moment of invocation when hallucinations arise in small LMs. Previous optimization efforts primarily focused on post-processing techniques, which were separate from the reasoning process of LMs, resulting in high computational costs and limited effectiveness. In this paper, we propose a practical invocation evaluation metric called AttenHScore, which calculates the accumulation and propagation of hallucinations during the generation process of small LMs, continuously amplifying potential reasoning errors. By dynamically adjusting the detection threshold, we achieve more accurate real-time invocation of large LMs. Additionally, considering the limited reasoning capacity of small LMs, we leverage uncertainty-aware knowledge reorganization to assist them better capture critical information from different text chunks. Extensive experiments reveal that our AttenHScore outperforms most baselines in enhancing real-time hallucination detection capabilities across multiple QA datasets, especially when addressing complex queries. Moreover, our strategies eliminate the need for additional model training and display flexibility in adapting to various transformer-based LMs. Our code is available at https://github.com/Robot2050/AttenHScore.

1 Introduction

With the profound study of the scaling law (Kaplan et al., 2020) and the density law (Xiao et al., 2024), the development and application of language models (LMs) have exhibited a diversified pattern. In this context, the remarkable performance of large language models (LLMs) such as GPT-4o in reasoning tasks has attracted significant attention (Hosseini et al., 2023). However, due to their complex structures and massive parameter scales, these LLMs consume considerable computational resources during training and inference. Consequently, many of these LLMs are only available through paid API services, undoubtedly increasing their monetary cost. Meanwhile, small language models (SLMs), with their lightweight architectures and efficient inference capabilities (Zhang et al., 2024), demonstrate significant advantages in specific scenarios, such as real-time responses on edge devices (Khiabani et al., 2025) and rapid processing of simple tasks (Li et al., 2024). Nevertheless, when faced with higher-level tasks requiring complex semantic understanding, the capabilities of SLMs appear to be inferior compared to those of LLMs (Wang et al., 2024).

To balance performance and cost while enhancing overall efficiency, a new paradigm of collaboration between large and small LMs has emerged from the perspectives of cost-effectiveness and resource optimization. This paradigm aims to fully leverage the advantages of LLMs in handling complex tasks while exploiting the efficiency of SLMs in simple problem scenarios, thus achieving optimal resource allocation and efficient task processing. As illustrated in Figure 1, we conduct retrieval-based question answering (QA) experiments utilizing two LMs, one large and one small, across five datasets from Longbench (Bai et al., 2023), to evaluate the performance of both LMs in scenarios without retrieval, and with top-5, top-10, top-15 retrieval results. LLM exhibits overall superior performance, but the gap between it and SLM is remarkably narrow on certain datasets. Under these circumstances, researchers have mainly proposed two strategies: routing and cascading. The core mechanism of the former lies in accurately directing user queries to a specific model based on criteria provided by specially trained models (Aggarwal et al., 2023; Ding et al., 2024b). Com-

*Corresponding author: qinbiao@ruc.edu.cn

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 1518–1532
November 4-9, 2025 ©2025 Association for Computational Linguistics

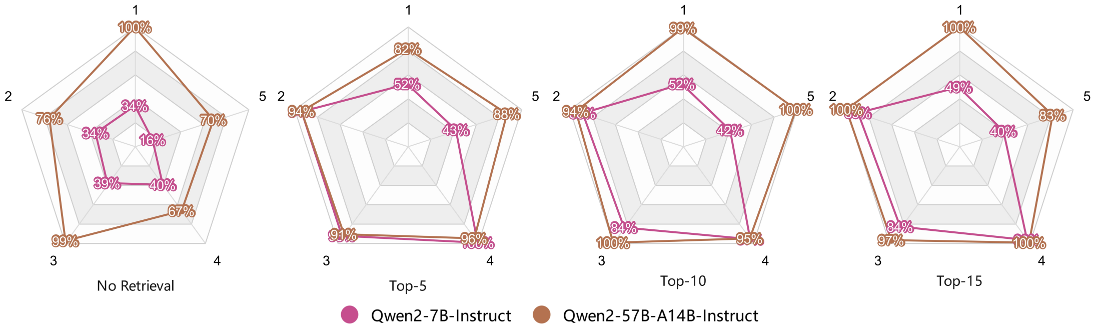

Figure 1: Performance of large and small LMs on different QA datasets in the RAG scenario. Including 1: 2WikiMultihopQA, 2: MultiFieldQA-en, 3: Qasper, 4: MultiFieldQA-zh and 5: HotpotQA.

paratively, the latter exhibits a more flexible and phased processing mode. According to this strategy, user queries are first sent to SLMs for initial processing. Then, based on the output results of these models, the system determines whether further in-depth reasoning by LLMs is necessary (Yue et al., 2023; Ramírez et al., 2024).

Based on the aforementioned research, we find that routing strategies require the introduction of auxiliary models for decision-making during implementation, which contradicts the initial goal of simplicity and efficiency. More importantly, these auxiliary models not only require specialized training but also often rely on specific datasets (Šakota et al., 2024; Ding et al., 2024b), potentially limiting their versatility across different tasks. In view of this, we have chosen to adopt a cascading strategy, where the main technical challenge lies in accurately determining when hallucinations occur in SLMs. Currently, research on hallucination detection in LMs primarily focuses on the post-reasoning phase (Manakul et al., 2023; Zhang et al., 2023; Li et al., 2023). However, such methods exhibit significant limitations when integrated into the practical LLMs applications. The primary issue is that these post-processing methods often incur high computational costs and notable delays. For instance, cutting-edge detection methods typically utilize LLMs such as ChatGPT, OPT, etc. (Zhang et al., 2023), making the cost of hallucination detection comparable to or even more expensive than LLMs reasoning tasks. What's more, post-processing methods are independent of the reasoning process (Shi et al., 2022; Wang et al., 2022), thus they cannot delve into the origins and evolution of hallucinations within each LMs.

Seeking to surmount the outlined restrictions, we shift the focus of optimizing LMs invocations towards understanding their existing available signals, rather than training and running more auxiliary models. This paper proposes a practical invocation evaluation metric, AttenHScore, designed to calculate the accumulation and propagation of hallucinations during the generation process of SLMs. By continuously amplifying potential error points, this metric enables more skillfully identify deviations between generated content and facts, thereby improving the detection accuracy of hallucinations. Furthermore, from the perspective of retrieval-augmented generation (RAG), we guide SLMs to evaluate the uncertainty between queries and different text chunks, optimizing the information arrangement by moving more relevant content from the retrieval to the front of the prompt, thereby further assisting SLMs in capturing key information and enhancing their accuracy in QA tasks.

The main contributions of this work are as follows:

- We propose a method for optimizing LMs invocation based on the uncertainty of generated text. The core technology lies in the thorough consideration of accumulation and propagation effects of hallucinations, thereby achieving unsupervised, real-time and plug-and-play invocation optimization.
- In the realm of retrieval-based QA, we fully utilize the chain-of-thought reasoning capability of generative LMs and guide text reranking through an uncertainty evaluation mechanism to precisely optimize information arrangement.
- To validate the effectiveness of our method, we test it on four QA datasets utilizing three different LLMs and conduct an in-depth anal-

ysis of the proposed method through multi-dimensional experiments.

2 Related Works

Collaboration of SLMs and LLMs The joint application of LLMs and SLMs has recently emerged as a technological approach, achieving breakthroughs in multiple research areas (Ma et al., 2023; Ding et al., 2024a; Min et al., 2024). In the studies by Sakota et al. (2024) and Lu et al. (2023), they proposed training an auxiliary model to estimate the success rate of invoking LLMs. Chen et al. (2023) introduced a cascade strategy, utilizing an auxiliary model to predict the accuracy of outputs from SLMs. Additionally, Yue et al. (2023) suggested repeatedly invoking SLMs to perform inference tasks, while research by Ramírez et al. (2023) indicated that the margin of a knowledge-distilled model has the potential to enhance the efficiency of calls made to LLMs. Later, Ramírez et al. (2024) proposed the Margin Sampling approach, which identifies hallucinations by computing the margin between the most likely first and second tokens. However, the above methods are more suitable for short answer generation tasks, while the direct judgment of long answer generation is still a gap and more challenging.

Hallucination Detection The concept of hallucination, which originally emerged from the fields of pathology and psychology (Macpherson, 2013), has been subsequently adopted and applied in the domain of Natural Language Processing (Maynez et al., 2020). The occurrence of hallucinations is widespread in deep learning models utilized for a range of text generation tasks (Dziri et al., 2022; Su et al., 2022). It is defined as the generation of content that lacks practical significance or deviates from the provided source material (Ji et al., 2023). With the widespread adoption of LLMs in various applications, the issue of hallucinations arising from these LMs has garnered significant attention from researchers (Shen et al., 2023; Becker et al., 2024). In this context, Min et al. (2023) introduced the FactScore method, which leverages knowledge sources to verify the accuracy of each atomic fact in the generated text. Furthermore, Manakul et al. (2023) presented SelfCheckGPT in their study, a black-box technique for hallucination detection. Despite those advancements, their methods still possess certain limitations. They either rely on external knowledge bases or require the analysis of multiple responses sampled from LMs, which undoubtedly increases resource consumption and reduces efficiency.

3 Optimizing the Adaptive Invocation Interface for LLMs

Our design philosophy is to "maximize the success rate of SLMs and minimize the need for LLM invocations", thereby embodying the idea of "Invoke Interfaces Only When Needed". We achieve this goal through a two-stage strategy that combines "pre-event optimization" with "post-event quality control":

Re-ranking: This module aims to reduce the likelihood of SLMs generating hallucinations at the source. As we pointed out in the paper, SLMs have limited capabilities in processing long texts and extracting key information. By re-ranking the input information and placing the most critical content at the forefront, we can significantly optimize the working environment for SLMs, thereby proactively minimizing scenarios that might necessitate LLM invocations.

Hallucination Detection: When "pre-event optimization" is still insufficient to completely avoid errors, this module serves as a safety net. It is responsible for real-time monitoring of SLM outputs to precisely identify moments when LLM invocations are truly necessary.

3.1 Problem Definition

In this paper, we focus on predicting the mapping relationship between elements in the input space X and their corresponding labels in the output space Y. Here, $(x_1, ..., x_q) \sim X$ represents the response generated by SLMs upon a user query, while $(0, 1) \sim Y$ denotes the decision flag indicating whether to invoke LLMs. We transform the system into the predictor $f : X \to Y$. For each incoming X, we determine whether to call LLMs based on the hallucination detection strategy. The entire procedure is outlined in Algorithm 1.

3.2 Real-time Hallucination Detection

Our work is grounded in a key insight: hallucinations are not instantaneous and isolated errors but rather a conductive and gradually accumulating dynamic process. The uncertainty signals from individual tokens are often weak and ambiguous. However, when these signals appear continuously and are amplified along the inference chain, they constitute a strong warning that hallucinations are about to occur.

Based on the aforementioned in-depth analysis, we ascertain that current methods relying on post-processing or uncertainty measures are inadequate for detecting hallucinations in collaborative large and small LMs systems. Given this limitation, we address the issue from the perspective of sequence generation in SLMs. Observing the accumulation and propagation of hallucinations during the token-by-token generation process, we propose the AttenHScore evaluation metric to quantify these characteristic, thereby providing valuable guidance for hallucination detection. As illustrated in Figure 2, we define this metric as follows:

$$H = \sum_{i=1}^{K} a_i I_i = -\sum_{i=1}^{K} a_i \log p_{max}(x_i) \tag{1}$$

where $p_{max}(x_i)$ represents the maximum probability of generating token x_i at position i, I_i denotes the degree of uncertainty for that token, and a_i signifies the accumulation and propagation weight of hallucination designed for each I_i, which is specifically calculated as:

$$a_i = p_{max}(x_i)\text{Atten}(x_i) \tag{2}$$

Specifically, $\text{Atten}(x_i)$ is used in attention-based models to measure the degree of attention the LM pays to each token, reflecting which tokens are more important and relevant for answering in the current processing step. By multiplying $p_{max}(x_i)$ and the attention score, we obtain a weight that comprehensively reflects the degree of attention and confidence of the token during model processing. Therefore, the above two steps of accumulation and multiplication together highlight the hallucinations of LMs during generation more effectively.

If the generated text is long, we preset a value K, calculate an AttenHScore value for every K tokens, and take the maximum as the object to compare with the threshold to determine whether to invoke the LLM:

$$S_{RHDI} = \max\{H_1, H_2, ..., H_n\} \tag{3}$$

In addition, we conduct a comprehensive design for the computation of $\text{Atten}(x_i)$. Initially, we integrate the softmax function with the mask function to generate the attention weight matrix M:

$$M = \text{softmax}\left(\text{mask}\left(\frac{QR^\top}{\sqrt{d_k}}\right)\right) \tag{4}$$

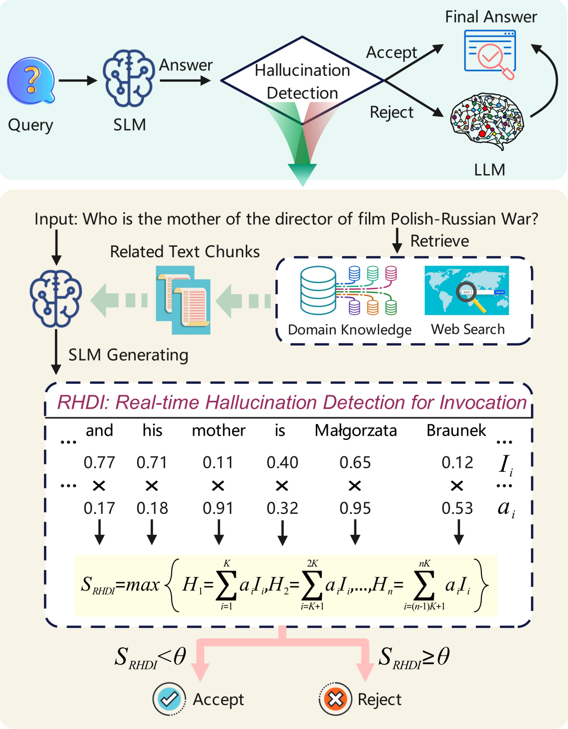
Final Answer
Query
SLM
Answer
Hallucination Detection
Accept
Reject
LLM
Input: Who is the mother of the director of film Polish-Russian War?
Retrieve
Related Text Chunks
Domain Knowledge
Web Search
SLM Generating
RHDI: Real-time Hallucination Detection for Invocation
... and his mother is Małgorzata Braunek ...
0.77 0.71 0.11 0.40 0.65 0.12 I_i
... × × × × × × ...
0.17 0.18 0.91 0.32 0.95 0.53 a_i
$S_{RHDI}=max\left\{H_1=\sum_{i=1}^{K}a_iI_i, H_2=\sum_{i=K+1}^{2K}a_iI_i, ..., H_n=\sum_{i=(n-1)K+1}^{nK}a_iI_i\right\}$
$S_{RHDI}<\theta$
$S_{RHDI}\geq\theta$
Accept
Reject

Figure 2: Overview of our hallucination detection and collaborative framework.

where Q represents the query matrix, R stands for the key matrix, and d_k denotes the dimensionality of a key vector. Following this, for a given token x_i, we determine its corresponding maximum attention value by searching through all elements $M_{j,i}$ where $j > i$. Lastly, to further enhance the influence of attention scores in the overall evaluation, we employ an exponential function to amplify them:

$$\text{Atten}(x_i) = \exp\left(\max_{j>i} M_{j,i}\right) \tag{5}$$

Through this approach, we ensure that the attention mechanism plays a more prominent role in the evaluation. It is worth noting that in Eq. (5), we choose max to calculate attention scores, rather than basing on a specific layer or taking an average. This is because we believe that doing so may be affected by special cases and reduce the perception of key content, thereby affecting our final detection performance. This is experimentally confirmed in Section 4.6.

3.3 Dynamic Threshold

Setting a threshold for decision criteria is a common requirement across all strategies, and we introduce a dynamic threshold mechanism. Specifically, we first utilize the results of the first five queries to calculate an initial threshold. During this process, we do not evaluate whether these five queries trigger the LLM, but only obtain output results from the SLM. Subsequently, at each new query, we incorporate the hallucination score of the current query into the historical records and recalculate the average hallucination score of all processed queries, using this as the updated threshold.

3.4 Re-ranking Strategy based on Uncertainty Evaluation

In long text processing scenarios, SLMs often face challenges in extracting effective information, leading to inefficient utilization of key information. Additionally, these SLMs exhibit a significant position bias phenomenon in long texts, where they tend to focus more on the beginning of the prompt and easily overlook information in the middle (Jiang et al., 2023). Therefore, we introduce auxiliary mechanisms for SLMs to enhance their information utilization capabilities.

Given a query, we are able to retrieve multiple associated text chunks. For each text chunk, we guide SLMs to perform reverse thinking, which involves generating the corresponding query based on the text content. Afterwards, we quantify the uncertainty of this generation process using the following method:

$$G = -\sum_{x_i \in X} \mathrm{Atten}(x_i) \log p(x_i) \tag{6}$$

where X represents the token set of the known query. This approach takes full advantage of the powerful reasoning capabilities and deep understanding of structural nuances inherent in current LMs. Experimental results presented in Section 4.5 suggest that this method possesses generalization capabilities, enabling it to more accurately filter out noisy or incomplete information when compared to prevailing benchmark models.

By integrating the various strategies we proposed, real-time hallucination detection and re-ranking are achieved within a large and small LMs collaboration system without the need for additional model training. This process is unsupervised, namely, our methods do not require manual supervision or labeled data for training. More meaningfully, our methods are universally applicable to all transformer-based LMs, truly embodying the plug-and-play principle and showcasing flexibility.

Algorithm 1 Adaptive Invocation for LLMs in QA

```
Input: SLM generator $M_s$, LLM interface $M_l$,
    User query $Q_i$, Initial threshold $\theta$
Output: Decision $y \in \{0, 1\}$ for LLM invocation,
    Response $R$
while new user query $Q_i$ arrives do
    $M_s(Q_i)$ generate candidate tokens $X = \{x_1, ..., x_q\} \rightarrow$ logits, attentions
    if $i \leq 5$ then
        $y \leftarrow 0, R(Q_i) \leftarrow X$
    else
        Calculate $\mathrm{Atten}(x_i)$
        Gradually calculate $H_1, H_2, ..., H_n$
        $S_{RHDI} \leftarrow \max\{H_1, H_2, ..., H_n\}$
        if $S_{RHDI} < \theta$ then
            $y \leftarrow 0, R(Q_i) \leftarrow X$
        else
            $y \leftarrow 1, R(Q_i) \leftarrow M_l(Q_i)$
        end if
        Updadte $\theta \leftarrow \frac{\sum_{k=1}^{N} S_{RHDI}(X_k)}{n}$
    end if
end while
```

4 Experiment

4.1 Datasets and Metrics

We adopt four highly recognized QA datasets for evaluation, including two open-book conversational datasets: CoQA (Reddy et al., 2019) and SQuAD (Rajpurkar, 2016), and two closed-book QA datasets: TriviaQA (Joshi et al., 2017) and Natural Questions (NQ) (Kwiatkowski et al., 2019). CoQA is sourced from seven different domains, with each dialogue involving two crowd workers engaging in a question-and-answer exchange around a passage (Reddy et al., 2019). SQuAD is renowned for its large scale and high quality, with its origins in Wikipedia articles (Rajpurkar, 2016). TriviaQA is a reading comprehension dataset that comprises question-answer-evidence triplets (Joshi et al., 2017). NQ contains authentic queries posed by users to Google Search, along with answers sourced from Wikipedia (Kwiatkowski et al., 2019). On the other hand, the actual answers in the CoQA and SQuAD datasets are often longer, whereas answers in the TriviaQA and NQ datasets tend to be in the form of single or few-word responses. For evaluation metrics, we follow the prior work of Ren et al. (2022) and Chen et al. (2024) by utilizing the area under the receiver operator characteristic curve (AUROC) and accuracy (ACC). Specifically,

Dataset		**CoQA**			**SQuAD**			**TriviaQA**			**NQ**	
Method	**AUCs**	**AUCr**	**ACCr**	**AUCs**	**AUCr**	**ACCr**	**AUCs**	**AUCr**	**ACCr**	**AUCs**	**AUCr**	**ACCr**
						Llama3-8B-Instruct						
Perplexity	0.5783	0.5509	0.5251	0.4745	0.4645	0.4782	**0.8431**	0.8342	**0.7525**	**0.7700**	0.7694	0.6742
Energy	0.4212	0.3797	0.4025	0.4297	0.4129	0.4497	0.7204	0.6920	0.6547	0.6551	0.6440	0.6111
AVG-Range	0.5344	0.5033	0.5068	0.4609	0.4516	0.4821	0.8277	0.8229	0.7473	0.7492	0.7555	0.7172
LN-Entropy	0.6732	0.6668	0.6280	0.6113	0.6134	0.6179	0.8189	0.8177	0.7518	0.7490	0.7553	0.6640
LexicalSimilarity	0.7602	0.7614	0.7041	0.6365	0.6341	0.5562	0.7838	0.7746	0.7412	0.7354	0.7321	**0.7172**
EigenScore	0.7910	0.8014	0.7328	0.7359	0.7417	0.6741	0.7941	0.7783	0.7410	0.7599	0.7587	0.6801
AttenHScore (Ours)	**0.8330**	**0.8706**	**0.8097**	**0.8715**	**0.9024**	**0.8176**	0.8334	**0.8388**	0.7513	0.7650	**0.7871**	0.7072
						Vicuna1.5-7B						
Perplexity	0.4701	0.3292	0.3492	0.5143	0.2610	0.3109	**0.8184**	0.8108	0.7366	0.6794	0.6794	0.6427
Energy	0.3817	0.2139	0.2307	0.4273	0.1648	0.1791	0.7316	0.7147	0.6632	0.5767	0.5613	0.4947
AVG-Range	0.4624	0.3128	0.4154	0.5164	0.2645	0.3591	0.7859	0.7820	0.7165	0.6395	0.6344	0.6615
LN-Entropy	0.5221	0.4274	0.3739	0.5672	0.4331	0.5383	0.7962	0.7974	0.7339	0.6792	0.6895	0.6593
LexicalSimilarity	0.5876	0.5518	0.4894	0.5656	0.4650	0.5530	0.7870	0.7833	0.7385	0.7279	0.7441	**0.7443**
EigenScore	0.6500	0.6648	0.5165	0.6441	0.6315	0.5309	0.7979	0.7880	0.7402	**0.7557**	0.7748	0.6825
AttenHScore (Ours)	**0.7503**	**0.8481**	**0.7840**	**0.7193**	**0.8085**	**0.7212**	0.8178	**0.8338**	**0.7467**	0.7524	**0.7949**	0.6958
						Llama2-13B-Chat-HF						
Perplexity	0.5423	0.5272	0.5108	0.4830	0.4638	0.4504	**0.8111**	0.8142	0.7422	0.6944	0.6942	0.6463
Energy	0.4380	0.3993	0.4596	0.4102	0.3890	0.4167	0.6976	0.6888	0.6545	0.6229	0.6133	0.5507
AVG-Range	0.5243	0.5075	0.5569	0.4651	0.4451	0.4612	0.7936	0.8002	0.7276	0.6570	0.6562	0.6620
LN-Entropy	0.6005	0.6018	0.5867	0.5938	0.5904	0.5778	0.7729	0.7855	0.7208	0.6849	0.6931	0.6169
LexicalSimilarity	0.7155	0.7331	0.6593	0.6536	0.6667	0.6623	0.7439	0.7466	0.7303	0.7286	0.7373	0.6928
EigenScore	0.7509	0.7809	0.7120	0.7364	0.7585	0.6670	0.7512	0.7502	0.7265	**0.7477**	0.7645	0.6717
AttenHScore (Ours)	**0.8369**	**0.8982**	**0.8320**	**0.8544**	**0.9032**	**0.8322**	0.8036	**0.8221**	**0.7442**	0.7423	**0.7785**	**0.6978**

Table 1: Main experimental results are presented in four QA datasets. The best result is in bold, and the second best result is underlined.

AUCs denotes the AUROC score with sentence similarity serving as the measure of correctness, while AUCr represents the AUROC score with the Rouge-L score as the correctness measure, and ACCr follows similarly.

4.2 Baselines

We undertake a comparative analysis of our proposed approach with the prevalent uncertainty-based techniques, namely Length-normalized Entropy (LN-Entropy) (Malinin and Gales, 2020), the consistency-based metric Lexical Similarity (Lin et al., 2022) as well as EigenScore (Chen et al., 2024), which utilizes the eigenvalues of the response covariance matrix to quantify semantic consistency or diversity in the dense embedding space. All three aforementioned methods require SLMs to generate multiple answers to the same question. In addition, we introduce three comparison methods that only require SLMs to generate an answer once. Perplexity evaluates the rationality of text generation by calculating the predictive probability distribution of SLMs (Ren et al., 2022). AVG-Range assesses credibility by measuring the average difference between the highest and lowest probabilities in the probability distribution of each token output by SLMs (Ramírez et al., 2024). Energy score (Liu et al., 2020), a popular out-of-distribution detection method, is tested for its applicability in hallucination detection. Our methodology also adheres to the paradigm of single-pass model generation.

4.3 Implementation Setting

In experiments aimed at detecting hallucinations for collaboration, we primarily employ three LMs with the following hyperparameter settings: temperature at 0.5, top-p at 0.99, top-k at 5, and the number of generations set to 10. When assessing the correctness of generated answers, we adopt two commonly used methods: Rouge-L (Lin, 2004) and semantic similarity (Reimers, 2019). The former employs the threshold of 0.5, while the latter uti-

Methods	**ERNIE-3.5**	**Qwen-Plus**	**Qwen-Turbo**	**Deepseek-v3**	**Qwen-72B**	**Qwen1.5-72B**	**Qwen2-57B**	**R1-LLama-70B**	**R1-Qwen-32B**
Initial Score with Only Vicuna-7B-v1.5: 13.25; Score After Our Re-ranking Process: 16.62.									
With Large-Small Language Model Collaboration									
Perplexity	17.82	17.01	20.31	17.27	18.76	19.14	18.02	17.51	17.69
Random	21.05	18.9	22.61	19.53	20.89	22.16	20.28	20.72	20.28
AVG_Range	22.33	20.87	23.49	20.39	21.93	22.59	21.62	21.65	21.34
AttenHScore	**24.91**	**23.27**	**26.23**	**22.82**	**24.95**	**25.71**	**23.69**	**23.76**	**24.03**
For Reference: *Scores Obtained by Exclusively Utilizing Interfaces of Various Large Language Models.*									
LLMs	25.12	22.25	27.71	22.0	26.05	27.45	23.9	23.65	23.5

Table 2: We report the metric F1 score of QA performance under three scenarios: SLM only, large-small LM collaboration, and LLM only.

lizes the nli-roberta-large model with the threshold set to 0.9. Moreover, in conducting collaborative experiments between small and large LMs, we select Vicuna-7B-v1.5 as the SLM and incorporated nine distinct LLM interfaces to participate in the experiments. We incorporate RAG techniques, using bge-large-en-v1.5 as the retriever and setting the number of retrieved text chunks to 10. Detailed experimental setup information can be found in Appendix A.3.

4.4 Main Results

In this section, we first conduct a comprehensive evaluation of the key component for detecting hallucinations in SLMs within the collaborative system of large-small LM on the hallucination benchmark (Chen et al., 2024). Subsequently, we integrate AttenHScore into the entire system and evaluate its accuracy in determining interface calls by comparing various real-time hallucination detection methods.

4.4.1 Overall Results of the Hallucination Detection Component

To comprehensively validate the effectiveness of our proposed AttenHScore, we conduct experiments exploiting three LMs and four widely-used QA datasets. In designing the experiments, we not only consider the diversity of baseline methods but also emphasize the comprehensiveness of evaluation metrics to ensure the objectivity and accuracy of assessment results. The experimental results, as presented in Table 1, demonstrate that our AttenHScore achieves significant performance improvements on both CoQA and SQuAD datasets. Specifically, our method outperforms other baseline methods across various evaluation metrics and exhibits stable improvements across different LMs. On TriviaQA and NQ datasets, we observe that the methods based on perplexity and AVG-Range exhibit larger variations in performance compared to their performance on CoQA and SQuAD. This is related to the fact that answers in the TriviaQA and NQ datasets are generally simpler and shorter. Our proposed method exhibits superior performance when handling complex questions. With respect to simpler questions, its performance is comparable to that of state-of-the-art methods.

4.4.2 Collaborative Performance of LLMs and SLMs in QA

By integrating our proposed model hallucination discrimination method and re-ranking strategy into the large-small LMs collaboration system, we conduct further experiments on the MultiFieldQA-zh from the Longbench benchmark (Bai et al., 2023), with the specific setup detailed in Appendix A.3. The results in Table 2 show that simply reordering the retrieved content before inputting it into SLMs achieves significant performance improvement of 3.37. This indicates that SLMs encounter information overload issues when processing lengthy contexts, and optimizing the semantic relevance of the input sequence can effectively alleviate the limitations of their attention mechanisms.

Under the condition of limiting the total number of LLMs calls to 40%, we compare the impact of four real-time detection and calling methods on performance improvement and find that AttenHScore method performs more prominently in terms of enhancing performance. It is worth noting that in the four columns, we find the performance of model collaboration to be slightly better than using the LLM alone. This finding is consistent with the observation results presented in Figure 1. It also indicates that when dealing with certain RAG problems, the performance of SLMs is comparable to or even better than that of LLMs.

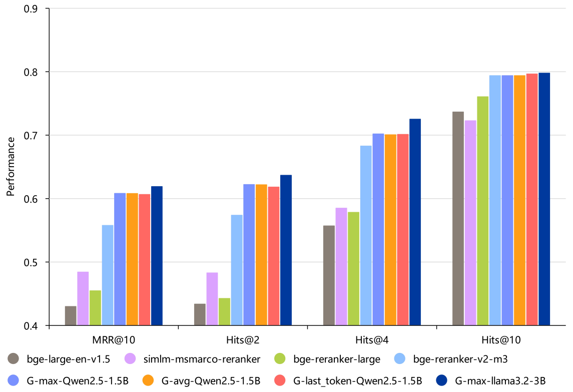

Figure 3: Performance comparison between the re-ranking method based on uncertainty evaluation and commonly used re-ranking models. Among them, the one starting with G represents our approach, and the rest of the models are all from huggingface.

4.5 Comparison and Reflection on Re-ranking

In long text scenarios, SLMs struggle with information extraction and show a position bias, thus we introduce auxiliary mechanisms to enhance their capabilities. As shown in Figure 3, after retrieving the top-15 text chunks, we evaluate the relevance of the rearranged top-10 content. By comparing our proposed re-ranking method based on uncertainty G with four existing re-ranking models, experimental results clearly demonstrate the excellent performance of our approach on the MRR@10, Hits@2, and Hits@4 metrics, indicating that our uncertainty can fully utilize the reasoning capabilities of LMs to more accurately identify texts relevant to the question. On the Hits@10 metric, our method slightly outperforms the most advanced re-ranking model, which is due to the incomplete retrieval results of top 15. In addition, we find that there is little difference in performance between using max, avg, and last-token to calculate attention scores, with max performing slightly better. Meanwhile, stronger LMs assisting uncertainty can further improve the performance of rearrangement.

4.6 Ablation studies

The calculation methods of attention scores exhibit diversity, and we specifically test three methods listed in Table 3. Experimental results reveal that the performance achieved using the max method surpasses that of the last-token and avg methods in both types of LMs. This superiority is primarily attributed to the fact that the max method is more effective in capturing the most prominent and critical information within the text sequence. In contrast, the last-token method tends to overly focus on the tail information of the sequence while neglecting other important elements, and the avg method tends to dilute the significance of key information due to averaging processing. This finding aligns with our proposed approach of detecting from the perspective of hallucination accumulation and transmission.

Dataset	CoQA			SQuAD		
Attention	AUCs	ACCr	AUCr	AUCs	ACCr	AUCr
			Llama3-8B-Instruct			
last-token	0.8226	0.8564	0.7948	0.8580	0.8864	0.8050
avg	0.8308	0.8673	0.8065	0.8678	0.8980	0.8176
max	**0.8330**	**0.8706**	**0.8097**	**0.8715**	**0.9024**	**0.8176**
			Vicuna1.5-7B			
last-token	0.7473	0.8412	0.7675	0.7176	0.8014	**0.7279**
avg	0.7491	0.8454	0.7792	0.7190	0.8059	0.7181
max	**0.7503**	**0.8481**	**0.7840**	**0.7193**	**0.8085**	0.7212

Table 3: Analysis of differences in three attention score calculation methods under different models.

AUCs	SentenceSimilarity			Rouge-L		
Method	0.7	0.8	0.9	0.3	0.5	0.7
			Llama3-8B-Instruct			
Perplexity	0.5178	0.4898	0.4745	0.5528	0.5078	0.4937
Energy	0.4702	0.4423	0.4297	0.4885	0.4462	0.4333
AVG-Range	0.5016	0.4749	0.4609	0.5369	0.4957	0.4819
LN-Entropy	0.6185	0.6087	0.6113	0.6490	0.6288	0.6231
LexicalSimilarity	0.6549	0.6442	0.6365	0.6821	0.6640	0.6507
EigenScore	0.7303	0.7327	0.7359	0.7433	0.7397	0.7381
AttenHScore	**0.8207**	**0.8498**	**0.8715**	**0.8373**	**0.8618**	**0.8733**

Table 4: Impact of correctness thresholds on hallucination detection performance.

4.7 Hyper-parameter Sensitivity Analysis

Utilizing the Llama3-8B-Instruct model, we execute comprehensive ablation experiments on the SQuAD dataset. The experimental results, shown in Table 4, clearly demonstrate that different thresholds for correctness metrics have a significant impact on the final performance of hallucination detection. More importantly, our proposed AttenHScore exhibits superior performance compared to other baseline methods across various threshold settings.

On the other hand, we also carry out experiments on the decoding sampling hyperparameters of LMs, with specific results presented in Figures 4 and 5. Experimental data reveals that our approach shows remarkable robustness across a wide range of parameter configurations.

Furthermore, considering the variability in the length of answers generated by SLMs, we introduce a preset token count K during the calculation of hallucinatio, as specifically illustrated in Figures 6 and 7. Our approach involves calculating an AttenHScore value for every K tokens, and then selecting the maximum AttenHScore computed from the entire answer generated by SLMs as the basis for evaluation. Through observation, we find that system performance reaches an optimum when K is set between 10 and 20. Further details of the experimental design and analysis are provided in Appendix A.4.

5 Conclusion

Amidst the drive for efficiency and resource optimization, this study delves into the challenges of hallucination detection and prompt re-ranking within the collaboration of large and small LMs. We introduce a novel invocation discriminant metric, AttenHScore, which quantifies the accumulation and propagation of hallucinations in SLMs generations, enabling more precise detection of potential reasoning errors. Additionally, within a retrieval-based QA context, we steer SLMs to assess the uncertainty of queries relative to various text chunks, thereby achieving superior re-ranking and enhanced accuracy. Extensive experiments across four datasets reveal that our proposed real-time, plug-and-play detection methodology and re-ranking strategy strike an effective balance between cost and performance, eliminating the need for domain-specific knowledge or model training. We anticipate that our insights will inspire further researches into hallucination detection and re-ranking, ultimately promoting the development of collaboration between large and small LMs.

Limitations

We acknowledge certain limitations, particularly in relying on the internal states of the LLM for hallucination detection. While this approach can identify hallucinations to some extent, there is still room for improvement in its accuracy. Future work will focus on deeper exploration of the LLMs' internal states to further enhance the precision and reliability of hallucination detection. Additionally, despite demonstrating good performance in complex query tasks, there may still be deficiencies in handling extremely complex tasks or those requiring deep semantic understanding. For instance, tasks involving multi-hop reasoning or strong domain relevance may not be fully addressed by the current invocation strategy. The primary objective of this paper is to further enhance the performance of the current large-small LM collaboration system through more accurate hallucination detection techniques. We will next concentrate on overcoming the limitations of existing methods to achieve a more efficient and reliable collaboration system.

References

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, and 1 others. 2023. Automix: Automatically mixing language models. *arXiv preprint arXiv:2310.12963*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. 2024. Mars: Meaning-aware response scoring for uncertainty estimation in generative llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7752–7767.

Jonas Becker, Jan Philip Wahle, Bela Gipp, and Terry Ruas. 2024. Text generation: A systematic literature review of tasks, evaluation, and challenges. *arXiv preprint arXiv:2405.15604*.

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*.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.

Bowen Ding, Qingkai Min, Shengkun Ma, Yingjie Li, Linyi Yang, and Yue Zhang. 2024a. A rationale-centric counterfactual data augmentation method for cross-document event coreference resolution. *arXiv preprint arXiv:2404.01921*.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. 2024b. Hybrid llm: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*.

Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022. On the origin of hallucinations in conversational models: Is it the datasets or the models? *arXiv preprint arXiv:2204.07931*.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, and 1 others. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. *arXiv preprint arXiv:2403.04696*.

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

Mohammad Hosseini, Catherine A Gao, David M Liebovitz, Alexandre M Carvalho, Faraz S Ahmad, Yuan Luo, Ngan MacDonald, Kristi L Holmes, and Abel Kho. 2023. An exploratory survey about using chatgpt in education, healthcare, and research. *Plos one*, 18(10):e0292216.

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

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *arXiv preprint arXiv:2310.06839*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

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

Yahya Sowti Khiabani, Farris Atif, Chieh Hsu, Sven Stahlmann, Tobias Michels, Sebastian Kramer, Benedikt Heidrich, M Saquib Sarfraz, Julian Merten, and Faezeh Tafazzoli. 2025. Optimizing small language models for in-vehicle function-calling. *arXiv preprint arXiv:2501.02342*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. *arXiv preprint arXiv:2305.11747*.

Tianlin Li, Qian Liu, Tianyu Pang, Chao Du, Qing Guo, Yang Liu, and Min Lin. 2024. Purifying large language models by ensembling a small language model. *arXiv preprint arXiv:2402.14845*.

Wei Li, Wenhao Wu, Moye Chen, Jiachen Liu, Xinyan Xiao, and Hua Wu. 2022. Faithfulness in natural language generation: A systematic survey of analysis, evaluation and optimization methods. *arXiv preprint arXiv:2203.05227*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Zi Lin, Jeremiah Zhe Liu, and Jingbo Shang. 2022. Towards collaborative neural-symbolic graph semantic parsing via uncertainty. *Findings of the Association for Computational Linguistics: ACL 2022*.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2023. Routing to the expert: Efficient reward-guided ensemble of large language models. *arXiv preprint arXiv:2311.08692*.

Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! *arXiv preprint arXiv:2303.08559*.

Fiona Macpherson. 2013. The philosophy and psychology of hallucination: an introduction.

Fiona Macpherson and Dimitris Platchias. 2013. *Hallucination: Philosophy and psychology*. MIT Press.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

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*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Qingkai Min, Qipeng Guo, Xiangkun Hu, Songfang Huang, Zheng Zhang, and Yue Zhang. 2024. Synergetic event understanding: A collaborative approach to cross-document event coreference resolution with large language models. *arXiv preprint arXiv:2406.02148*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023.

Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *Advances in Neural Information Processing Systems*, 37:8901–8929.

P Rajpurkar. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Guillem Ramírez, Alexandra Birch, and Ivan Titov. 2024. Optimising calls to large language models with uncertainty-based two-tier selection. *arXiv preprint arXiv:2405.02134*.

Guillem Ramírez, Matthias Lindemann, Alexandra Birch, and Ivan Titov. 2023. Cache & distil: Optimising api calls to large language models. *arXiv preprint arXiv:2310.13561*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. 2022. Out-of-distribution detection and selective generation for conditional language models. In *The Eleventh International Conference on Learning Representations*.

Marija Šakota, Maxime Peyrard, and Robert West. 2024. Fly-swat or cannon? cost-effective language model choice via meta-modeling. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 606–615.

Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. In chatgpt we trust? measuring and characterizing the reliability of chatgpt. *arXiv preprint arXiv:2304.08979*.

Freda Shi, Daniel Fried, Marjan Ghazvininejad, Luke Zettlemoyer, and Sida I Wang. 2022. Natural language to code translation with execution. *arXiv preprint arXiv:2204.11454*.

Ben Snyder, Marius Moisescu, and Muhammad Bilal Zafar. 2024. On early detection of hallucinations in factual question answering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2721–2732.

Dan Su, Xiaoguang Li, Jindi Zhang, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. 2022. Read before generate! faithful long form question answering with machine reading. *arXiv preprint arXiv:2203.00343*.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *arXiv preprint arXiv:2307.03987*.

Fali Wang, Zhiwei Zhang, Xianren Zhang, Zongyu Wu, Tzuhao Mo, Qiuhao Lu, Wanjing Wang, Rui Li, Junjie Xu, Xianfeng Tang, and 1 others. 2024. A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness. *arXiv preprint arXiv:2411.03350*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Chaojun Xiao, Jie Cai, Weilin Zhao, Guoyang Zeng, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. Densing law of llms. *arXiv preprint arXiv:2412.04315*.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. 2023. Large language model cascades with mixture of thoughts representations for cost-efficient reasoning. *arXiv preprint arXiv:2310.03094*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Tianhang Zhang, Lin Qiu, Qipeng Guo, Cheng Deng, Yue Zhang, Zheng Zhang, Chenghu Zhou, Xinbing Wang, and Luoyi Fu. 2023. Enhancing uncertainty-based hallucination detection with stronger focus. *arXiv preprint arXiv:2311.13230*.

Jihao Zhao, Zhiyuan Ji, Yuchen Feng, Pengnian Qi, Simin Niu, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024. Meta-chunking: Learning efficient text segmentation via logical perception. *arXiv preprint arXiv:2410.12788*.

A Appendix

A.1 Hallucination Detection and Uncertainty Evaluation

The concept of "hallucination" originally stems from the research domains of pathology and psychology, where it is defined as the perception of entities or events that do not exist in reality (Macpherson and Platchias, 2013). In the field of natural language processing (NLP), hallucination typically manifests as the generation that appears nonsensical or contradicts the original content (Maynez et al., 2020). Broadly speaking, hallucinations arising in NLP tasks can be classified into two major categories: intrinsic hallucination and extrinsic hallucination (Li et al., 2022; Ji et al., 2023). The former refers to the conflict between the output content of LLMs and the original input information, while the latter refers to the generated content that cannot be verified by the original content.

As LLMs become increasingly adept at generating human-like text, distinguishing between accurate and hallucinated content has become a critical issue. Current research on hallucination detection requires access to the model's output content, latent states, or distributional features, and uncertainty assessment strategies based on the latter two have become an important research direction.

Fadeeva et al. (2024) introduce token-level and claim-conditioned uncertainty for fact-checking and entity -level detection. Varshney et al. (2023) detect hallucinations by identifying tokens with low confidence, utilizing an active detection and mitigation pipeline. The analysis of Snyder et al. (2024) involves examining softmax output probabilities, attention mechanisms, and gradients to identify early signs of hallucinations. The following approaches estimate uncertainty regarding meaning, rather than surface form, by considering entropy or semantic similarity over output distributions or samples. Semantic entropy (Farquhar et al., 2024), representing uncertainty at the meaning level, is introduced to robustly detect confabulations. MARS (Bakman et al., 2024), a method that weights tokens based on semantic context in uncertainty scoring, is employed. Nikitin et al. (2024) propose a semantic similarity-based uncertainty quantification method for LLMs, where kernel language entropy is exploited to assess uncertainty via von Neumann entropy over semantically-clustered model outputs. This field acknowledges high-certainty hallucinations and calibration as key unresolved challenges, pushing for a deeper introspective and semantics-based analysis. Our detection pipelines integrate probability features, content perception, and attention mechanisms to form a comprehensive signal.

A.2 Analysis of Real-Time Capability

The calculation of AttenHScore is based on the attention weights and generation probabilities produced by the model itself during the generation process. This information is naturally generated during inference and requires no additional computation. We simply leverage this readily available information for judgment, and the process is nearly instantaneous, thus the method introduces no additional time delays. Furthermore, our method is significantly more efficient compared to approaches that necessitate model training or multiple generations.

We highlight the following advantages exhibited by our method: (1) Unsupervised: As an evaluation metric for invocation, AttenHScore can be directly calculated without relying on any detector training process, simplifying the evaluation workflow. (2) Real-time: Compared to current post-processing methods, AttenHScore, as a real-time invocation detection metric, ensures the efficient evaluation process. (3) Plug-and-play: Designed as a lightweight algorithm, AttenHScore can be easily integrated into any existing Transformer-based LMs.

A.3 Detailed Experimental Setup for Reproducibility

All language models utilized in this paper employ the chat or instruct versions where multiple versions exist, and are loaded in full precision (Float32). The vector database is constructed using Milvus, where the embedding model for English texts is bge-large-en-v1.5[1], and bge-base-zh-v1.5[2] for Chinese texts. To more effectively verify the effectiveness of the component designed for detecting small-model hallucinations in the collaborative system of large-small LMs, we utilize three SLMs of different types and sizes: Llama3-8B-Instruct[3], Vicuna1.5-7B[4], and Llama2-13B-Chat-HF[5]. The sentence embeddings of model generation and the ground truth answer are extracted by the nli-roberta-large model[6].

In Table 2, we employ nine different LLM interfaces to conduct large-small LM collaborative experiments with Vicuna1.5-7B. These interfaces are as follows: ERNIE-3.5[7], Qwen-Plus[8], Qwen-Turbo[8], Deepseek-v3[9], Qwen-72B[10], Qwen1.5-72B[11], Qwen2-57B[12], DeepSeek-R1-

[1] https://huggingface.co/BAAI/bge-large-en-v1.5

[2] https://huggingface.co/BAAI/bge-base-zh-v1.5

[3] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[4] https://huggingface.co/lmsys/vicuna-7b-v1.5

[5] https://huggingface.co/meta-llama/Llama-2-13b-chat-hf

[6] https://huggingface.co/sentence-transformers/nli-roberta-large

[7] https://console.bce.baidu.com/qianfan

[8] https://bailian.console.aliyun.com/

[9] https://platform.deepseek.com/

[10] https://huggingface.co/Qwen/Qwen-72B-Chat

[11] https://huggingface.co/Qwen/Qwen1.5-72B-Chat

[12] https://huggingface.co/Qwen/Qwen2-57B-A14B-Instruct

Distill-Llama-70B[13], and DeepSeek-R1-Distill-Qwen-32B[13]. Our experimental setup involves retrieving 10 relevant documents for each query and having the SLM to generate responses accordingly. Subsequently, different hallucination detection methods are utilized to monitor the generation status of the SLM in real-time. If it is determined that the SLM's output contains hallucinations, the corresponding LLM interface is invoked to answer the question. Regarding text chunking operations, we adopt the LLM-based chunking method (Zhao et al., 2024).

A.4 Exploring Hyperparameter Settings for Optimal Performance

Different hyperparameter settings may not only serve as critical factors influencing model performance, but also exert differential impacts on the sensitivity of various detection methods. Consequently, we conduct a systematic analysis of hyperparameters including temperature, top-k and K.

Experimental data reveals that various detection methods exhibit relatively low sensitivity to the top-k, whereas LN-Entropy, LexicalSimilarity, and EigenScore demonstrate higher sensitivity to the temperature. Extensive experiments in Figures 4 and 5 confirm that our approach shows remarkable robustness across a wide range of parameter configurations.

In the experimental section described in Figures 6 and 7, we conduct a detailed comparative analysis of the performance across different values of K. The results indicate that the system achieves optimal performance when K is set between 10 and 20 tokens.

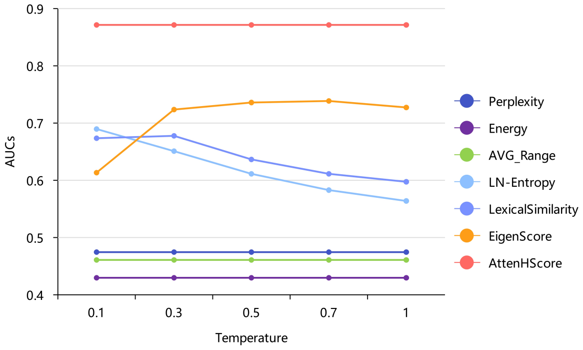

Figure 4: Performance sensitivity to temperature on Dataset SQuAD.

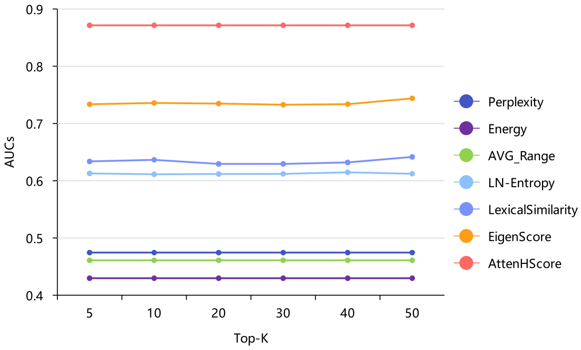

Figure 5: Performance sensitivity to top-k on Dataset SQuAD.

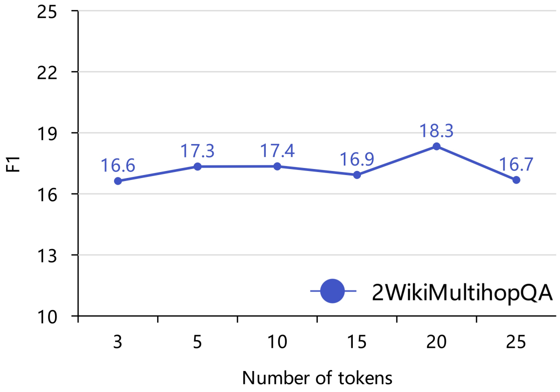

Figure 6: Performance sensitivity to K (Number of tokens) on Dataset 2WikiMultihopQA.

A.5 Setting Method for Dynamic Threshold

We adopt an adaptive strategy for threshold setting. Specifically, we first calculate the initial threshold using the average hallucination score of the first five queries. Subsequently, for each new query, we incorporate the current query's hallucination score into the historical records and recalculate the average hallucination score of all processed queries, using this as the updated threshold.

$$\theta = \frac{\sum_{i=1}^{n} S_{RHDI}(X_i)}{n}$$

In real-world production environments, systems are typically reused multiple times. We utilize the outputs from the first five queries to calculate the initial threshold. As each query is processed, the system records and dynamically computes the average hallucination score of previously generated answers in real time, thereby continuously adjusting the threshold. The update mechanism of dynamic threshold is independent upon the dataset.

For a strong model, which tends to produce lower average scores, its dynamic threshold will

[13] https://huggingface.co/deepseek-ai

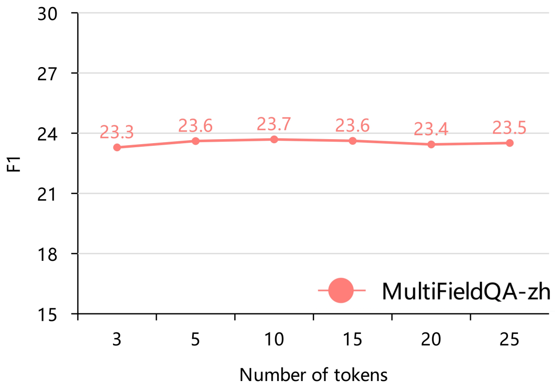

Figure 7: Performance sensitivity to K (Number of tokens) on Dataset MultiFieldQA-zh.

naturally be lower as well. Only when a question's score significantly exceeds this low baseline is it classified as a hallucination. For a weak model, which generates higher average scores, its dynamic threshold will be correspondingly higher. The system determines hallucations based on whether the current score significantly surpasses this high baseline.

Our dynamic threshold is not a fair yardstick aimed at measuring the absolute difficulty of all problems. Instead, it is an adaptive benchmark that reflects the typical performance of the current model when handling a recent stream of tasks. If the system encounters a series of complex problems in the initial stage, resulting in a relatively high average score, then a higher threshold is precisely reasonable as it accurately reflects the model's current operational state. The system's goal is to determine whether the score of the current problem is abnormal relative to the model's own average performance, rather than based on its absolute value.

The threshold is calculated based on the rolling average of all processed problems. This means that the influence of any early, occasional extreme high scores will be continuously diluted as the number of processed problems increases. In real-world scenarios where the system operates over an extended period , the bias introduced by the initial few problems will become negligible.

A.6 Further Exploration of Large-Small LM Collaboration

We conduct a more in-depth analysis and visualization of the experiments on the collaboration between large and small LLMs presented in Table 2. As shown in Figure 8, we accumulate the performance of SLMs, re-ranking, four real-time collaboration strategies, and LLMs, where each color represents the performance of a method under the corresponding LM interface. The scores of LLMs called separately and the collaboration system using AttenHScore as the hallucination detection component are relatively similar, indicating that our metric is more effective in identifying hallucinated information generated by SLMs. In Figure 9, we also demonstrate the performance trends of different methods under some LLMs through line charts. It can be observed that the overall data displays an upward trend, and two charts even have higher points at AttenHScore than when using only the LLM, which more directly illustrates the superiority of our method.

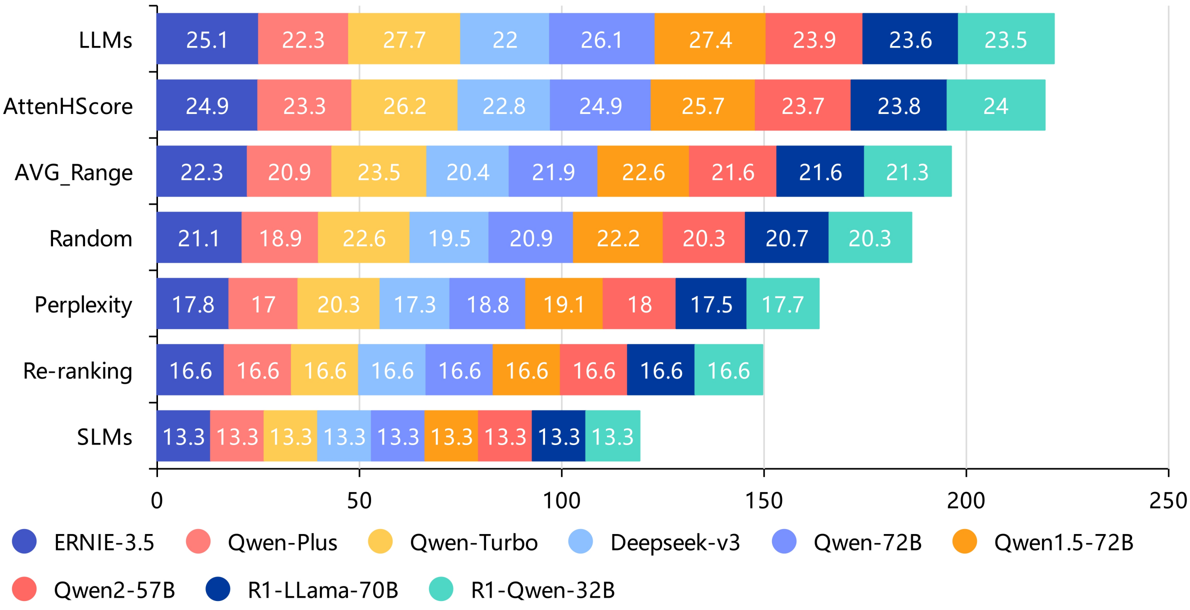

Figure 8: Comparative snalysis of AttenHScore and other methods in large-small LM collaboration system.

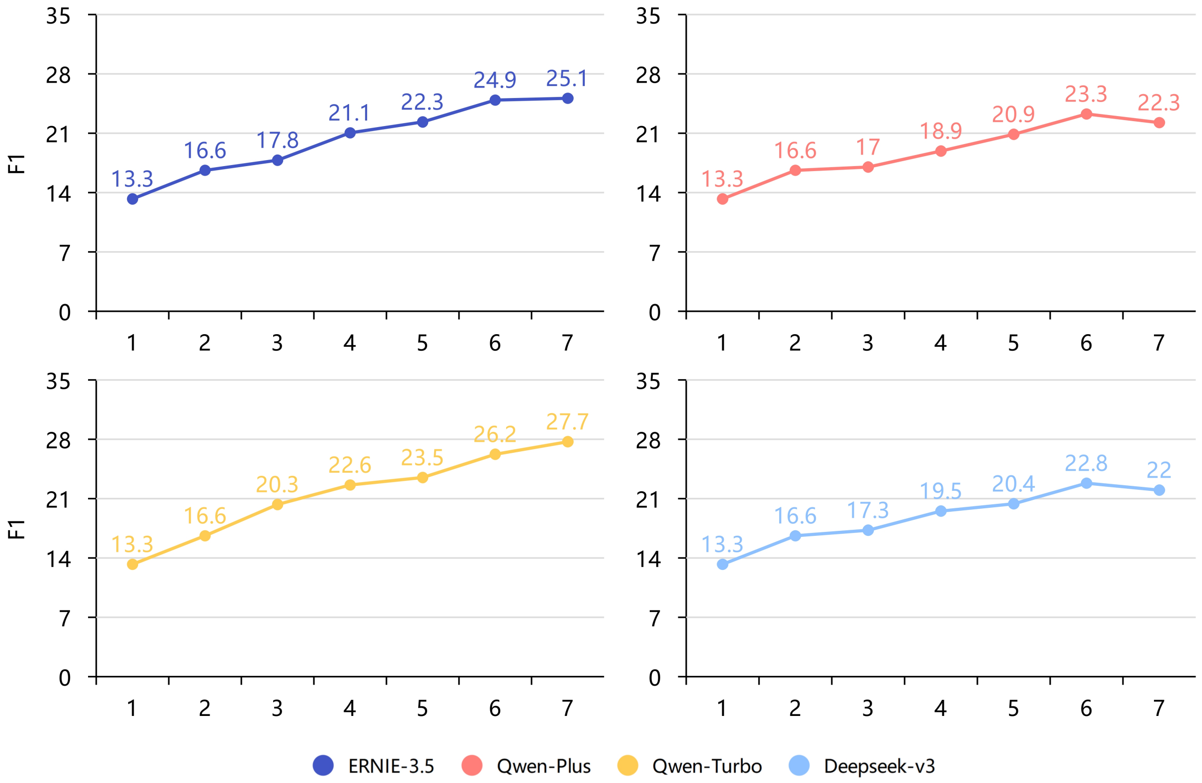

Figure 9: Performance variation trends of various large-small LM collaboration methods Under different LLM interfaces. The approaches include: 1: SLMs, 2: Re-ranking, 3: Perplexity, 4: Random, 5: AVG_Range, 6: AttenHScore and 7: LLMs.