DvD: Dynamic Contrastive Decoding for Knowledge Amplification in Multi-Document Question Answering

Anonymous ACL submission

Abstract

Large language models (LLMs) are widely used in question-answering (QA) systems but often generate information with hallucinations. Retrieval-augmented generation (RAG) offers a potential remedy, yet the uneven retrieval quality and irrelevant contents may distract LLMs. In this work, we address these issues at the generation phase by treating RAG as a multi-document QA task. We propose a novel decoding strategy, **D**ynamic Contrasti**v**e **D**ecoding (**DvD**), which dynamically amplifies knowledge from selected documents during the generation phase. DvD involves constructing inputs batchwise, designing new selection criteria to identify documents worth amplifying, and applying contrastive decoding with a specialized weight calculation to adjust the final logits used for sampling answer tokens. Zero-shot experimental results on ALCE-ASQA and NQ benchmark show that our method outperforms other decoding strategies. Additionally, we conduct experiments to validate the effectiveness of our selection criteria, weight calculation, and general multi-document scenarios. Our method requires no training and can be integrated with other methods to improve the RAG performance. Our codes are submitted with the paper and will be publicly available.

1 Introduction

The emergence of large language models (LLMs) has significantly advanced various natural language processing tasks (Touvron et al., 2023; Achiam et al., 2023). However, despite their extensive knowledge base and linguistic capabilities, LLMs frequently struggle with handling new knowledge and are susceptible to producing outdated content and hallucinations (Huang et al., 2023; Jiang et al., 2024). A straightforward resolution involves the continue updating of LLM's knowledge via training, but such a process typically demands substantial time and computational resources.

Retrieval-augmented generation (RAG) offers an alternative solution and has drawn substantive effectiveness to mitigate hallucination by introducing external knowledge (Gao et al., 2023b; Asai et al., 2023b). After document retrieval, RAG can be treated as a multi-document question answering (MDQA) task. Recent studies (Shi et al., 2023a; Yoran et al., 2024) indicate that the variability in document quality may cause distractions and impair the generation quality. Besides, knowledge conflicts, such as discrepancies within retrieved documents and between parametric and external non-parametric knowledge, may hinder the performance of LLMs (Chen et al., 2022; Jin et al., 2024b; Ni et al., 2024). Thus, addressing the integration of diverse knowledge during generation remains a significant challenge for LLMs.

The primary method for infusing new knowledge into LLMs involves supervised fine-tuning or continued training, which is resource-intensive. Prior research in RAG has introduced various improvements (Vu et al., 2023), such as improving retrieval quality (Shi et al., 2023d; Xu et al., 2023), refining responses through multiple iterations (Peng et al., 2023; Li et al., 2024), using optimized prompts (Ni et al., 2024), and developing new decoding strategies (Shi et al., 2023b; Zhao et al., 2024). However, these methods typically require retraining or multiple iterations. Contrastive decoding (Li et al., 2023) offers a training-free solution for hallucination mitigation and inspires many subsequent works (Shi et al., 2023b; Zhao et al., 2024), but they often concentrate on single-document scenarios and the resolution of conflicts between internal and external knowledge, overlooking the challenge of integrating multiple documents.

In this work, we propose a novel decoding strategy, termed **D**ynamic Contrasti**v**e **D**ecoding (**DvD**), to enhance the integration of various knowledge during the generation. The goal of DvD is to dynamically amplify knowledge from se-

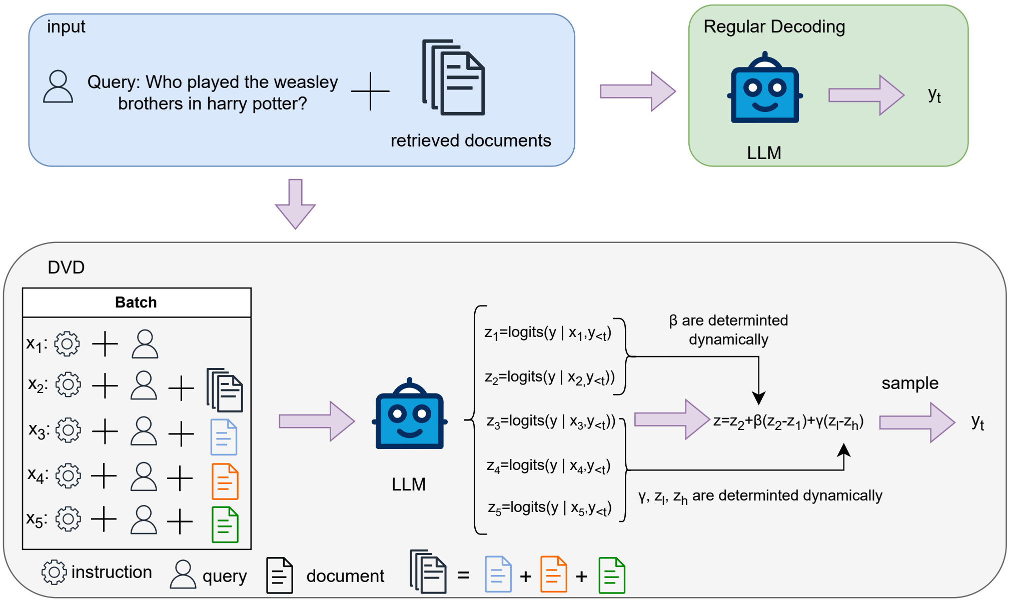

Figure 1: The framework of DVD. We propose a new decoding strategy with selection criteria and dynamic weight to incorporate knowledge from all documents and amplify knowledge from selected documents.

lected documents during integration to improve model-generated responses. The process starts with QA pairs associated with multiple retrieved documents. We create prompts for each question in *no-document*, *single-document*, and *multi-document* formats, and feed them into LLM in a single batch. During each inference step, the model produces logits for each prompt. Our method introduces a novel strategy for assessing logits from different prompts. These logits are then adjusted using contrastive decoding to refine the logits that guide the token generation. Furthermore, it investigates dynamically adjusting weights during the generation process, rather than relying on static values. See Figure 1 for better illustration.

To evaluate the effectiveness of our proposed method, we conducted zero-shot experiments across diverse datasets, including the ALCE-ASQA (Gao et al., 2023a) and Natural Questions (Kwiatkowski et al., 2019). Our experiments, utilizing the LLaMA2 (Touvron et al., 2023) and Vicuna (Chiang et al., 2023), demonstrate that our method consistently achieves superior response quality. This enhancement is attributed to our novel approach of dynamically amplifying knowledge from selected documents during the integration of different knowledge. A thorough analysis of our selection criteria, weight computation, and document count reveals consistent performance gains across all datasets. Importantly, our method is entirely plug-and-play, requiring no additional training. Furthermore, it seamlessly synergizes with other techniques, further augmenting the efficacy of the RAG system.

2 Related Work

2.1 Retrieval Augmented Generation

Retrieval-augmented generation (RAG) is a prominent research area in the development of LLMs, significantly improving answer accuracy and reducing hallucinations, especially in knowledge-intensive tasks (Gao et al., 2023b; Asai et al., 2023a). RAG operates by retrieving data from external sources and integrating it into response generation across two main phases: retrieval and generation. The training of the retrieval and generation components can be conducted independently, sequentially, or jointly (Asai et al., 2023a). This paper focuses solely on the generation phase, where the generator processes both traditional contextual information and retrieved text segments. Numerous studies aim to enhance the quality of generation through methods such as information compression (Yang et al., 2023; Xu et al., 2023), document reranking (Ma et al., 2023b; Zhuang et al., 2023; Sachan et al., 2022; Shi et al., 2023a), query rewriting (Ma et al., 2023a), structural and optimization modifications (Cheng et al., 2023; Shi et al., 2023c). Other methods include multi-round feedback (Peng et al., 2023; Asai et al., 2023b; Li et al., 2024), and improved prompts (Zheng et al., 2023; Ni et al., 2024). While many strategies necessitate training-specific modules, this paper emphasizes a plug-and-play decoding strategy that requires no training and is readily adaptable to various datasets.

2.2 Knowledge Conflicts

The generation phase for LLMs involves integrating both internal parametric and external non-parametric knowledge, which is challenging when knowledge conflict happens (Xu et al., 2024). Many studies have explored the behavior of LLMs in the presence of knowledge conflicts (Chen et al., 2022; Jin et al., 2024a; Ni et al., 2024; Xie et al., 2024; Tan et al., 2024; Jin et al., 2024b). These studies have identified factors that impact the preference of LLM during generation, such as confirmation bias, text similarity, semantic completeness (Tan et al., 2024; Xie et al., 2024; Jin et al., 2024a). These works typically create conflict datasets and develop strategies for better boundary understanding and response generation in LLMs, yet often limited to just a few external documents. Our work expands on this by incorporating multiple documents, aligning with RAG and practical scenarios, aiming to enhance the integration of diverse internal and external knowledge during generation.

2.3 Contrastive Decoding

Contrastive decoding, introduced by Li et al. (2023), identifies text by maximizing log probability discrepancies between expert and amateur models. This training-free method is effective and widely applicable, inspiring many studies (Zhang et al., 2023; Chuang et al., 2024; Jin et al., 2024a; Kim et al., 2023; Shi et al., 2023b; Zhao et al., 2024). Shi et al. (2023b) introduced context-aware decoding (CAD) to amplify output disparities with and without context, improving performance across datasets. Zhao et al. (2024) used contrastive decoding to merge knowledge from internal and external documents, incorporating a dynamic weight to adjust logits during generation. However, these approaches typically consider only one or two retrieved documents. In contrast, our work addresses the incorporation of knowledge from multiple documents, introducing new selection criteria and fusion methods to integrate all knowledge from both internal parametric and external multiple documents.

3 Methodology

We explain the details of our method in this section. We propose a new decoding strategy that can amplify knowledge from the selected documents during the generation phase to adjust the final logits used to sample answer tokens.

3.1 Notations

For each sample, we use q to present the question. The documents are retrieved based on their relevance with q. We neglect the retrieval phase and assume the retrieved documents as $D = \{d_1, d_2, ..., d_N\}$, where d_i is a single document and N is the overall number of documents. [1] Given q and D, our task is to generate answers for q based on retrieved documents D. The quality of documents varies, while the language model is supposed to incorporate its internal parametric knowledge and external knowledge from D to generate accurate and comprehensive answers.

We use x to present the input of large language models, which is constructed based on q, D, and certain prompt template T, and the output is indicated as y. The large language model is presented as θ and generates each token in answer y with auto-regressive style. At each time step t, LLM θ first generate logits z_t for answer token y_t, and compute the probability distribution as follows:

$$z_t = \theta(x, y_{<t}) \tag{1}$$

$$p_\theta(y_t|x, y_{<t}) = \texttt{softmax}(z_t) \tag{2}$$

The actual token y_t in answers y is generated based on the probability distribution through certain sampling strategies.

$$y_t \sim p_\theta(y_t|x, y_{<t}) \tag{3}$$

3.2 Dynamic Contrastive Decoding

Contrastive Decoding (Li et al., 2023) is an effective method to enhance the difference between logits with different input x and make the logits used to generate answer y more reasonable. Previous researches (Zhao et al., 2024; Shi et al., 2023b) only compare the input with single document (i.e., $x = T(q, d_1)$) or without documents(i.e., $x = T(q)$). However, we want to incorporate knowledge from all documents and amplify knowledge from certain important documents.

[1] The overall number of retrieved documents N is not less than 5, making it a multiple document setting.

We construct the input x in a special style. We consider multiple inputs simultaneously and apply different prompt templates to construct them. There are three types of inputs, corresponding to three templates. First, we consider the input without the documents, i.e., $x_1 = T_1(q)$. Second, we consider the input with all documents concatenating together, $x_2 = T_2(q, D)$. Last, we consider the input with a single document for each document in D, i.e., $x_3 = T_3(q, d_1), x_4 = T_3(q, d_2), ..., x_{N+2} = T_3(q, d_N)$. In conclusion, we construct $N + 2$ inputs for each sample, where N is the number of documents. Inspired by Su (2023), we construct these inputs into a batch and feed them into the LLM. $B = \{x_1, x_2, ..., x_{N+2}\}$. The LLM generates corresponding $N + 2$ logits simultaneously for each sample, which is denoted as Z. $Z = \{z_1, z_2, ..., z_{N+2}\}$.

$$Z = \theta(B) \tag{4}$$

We want to incorporate internal and external knowledge and amplify or neglect knowledge from certain documents, which need criteria to assess the quality of logits and make selections. Previous work often computes the entropy for each logit. However, LLMs tend to assign probabilities to numerous tokens in the vocabulary after pretraining, leading to the overall entropy being influenced by the meaningless probabilities of many tokens. Therefore, we emphasize the importance of head tokens, and only compute the entropy for tokens with top K[2] probability. We use the scoring function f to compute the following score s_i for each logits z_i in the batch B and get scores $S = \{s_1, s_2, ..., s_{N+2}\}$:

$$s_i = f(z_i) \tag{5}$$

$$f(z_i) = -\sum_{j=1}^{K} p(t_j)\log p(t_j), t_j \in V_{topK} \tag{6}$$

where V_{topK} is the set of tokens with top K highest probability. According to the characteristics of entropy, the lower the score, the better the distribution tends to be. The score s_1 and s_2 corresponding to inputs without and with documents, respectively, are first used to determine the importance of internal parametric knowledge. We assume that the model should prioritize the provided documents but cannot entirely disregard the influence of internal knowledge. Only if s_1 is more than one order of magnitude lower than the value of s_2 (i.e., s_1 <= s_2 / 10), should the LLM retain its reliance on internal knowledge. Otherwise, LLM should depend on the knowledge from documents to answer the question and eliminate self-interference. This weight threshold is related to the characteristics of datasets and is settled in the preliminary experiments. The scores s_3 to s_{N+2} are used to determine the importance of each document. The documents with the lowest score and highest score are selected to adjust the logits and amplify knowledge from the specific document, denoted as z_l and z_h respectively. The official formula is as follows:

$$\hat{z} = z_2 + \beta * (z_2 - z_1) + \gamma * (z_l - z_h) \tag{7}$$

where β and γ are hyperparameters, and β is set to 0 if s_0 is more than one order of magnitude lower than s_1.

Overall, the answer token is sampled based on the probability distribution generated on $\hat{z}$:

$$\begin{aligned} y_t \sim p_\theta(y_t|x, y_{<t}) &= \texttt{softmax}(\hat{z}) \\ = \texttt{softmax}(z_2 &+ \beta(z_2 - z_1) + \gamma * (z_l - z_h)) \end{aligned} \tag{8}$$

Equally,

$$y_t \sim p_\theta(y_t|x_2, y_{<t}) \frac{p_\theta(y_t|x_2, y_{<t})}{p_\theta(y_t|x_1, y_{<t})}^{\beta} \frac{p_\theta(y_t|x_l, y_{<t})}{p_\theta(y_t|x_h, y_{<t})}^{\gamma} \tag{9}$$

where x_1 and x_2 are inputs without and with documents correspondingly, x_l and x_h correspond to the two inputs with the lowest and highest scores.

Our method can be seen as an extension to CAD proposed by Shi et al. (2023b). We consider the influence of a single document, amplify knowledge from specific documents, and design special metrics to select the target document during the generation process.

In the preliminary experiments, we find that the setting of hyperparameters β and γ are crucial to downstream performance. It is inconvenient to run lots of experiments to explore the perfect weight for every dataset and language model. Therefore, we want to dynamically set these weights at each time step during the generation. Previous work (Zhao et al., 2024; Jiang et al., 2021) used the highest from the normalized predicted token probabilities probability for LLM confidence, which is not very effective in our experiments (See section 5.2 for

[2] K is a hyperparameter and we set K to 10 in main experiments. The influence of K is demonstrated in section 5.1.

further details). Inspired by Wang and Zhou (2024), we apply the difference in probability between the top 2 tokens as the confidence. Specifically,

$$C_i = p(y_t^1|z_i) - p(y_t^2|z_i) \tag{10}$$

$$\beta = \mathtt{max}(C_2 - C_1, 0) * \mathbb{1}(s_2/10 < s_1) \tag{11}$$

$$\gamma = \mathtt{max}(C_l - C_h, 0) \tag{12}$$

where $p(y_t^1|z_i)$ refers to the highest probability from distribution z_i and $p(y_t^2|z_i)$ refers to the second highest probability value. Therefore the dynamic version of γ is determined by the confidence difference between logits with the lowest score and highest score, while β is determined by logits with and without documents jointly.

In conclusion, we propose a new decoding strategy with selection criteria and dynamic weight to incorporate knowledge from all documents and amplify knowledge from selected documents.

4 Experiments

4.1 Experimental Settings

Datasets We conduct the experiments on a zero-shot open-domain QA setting, where documents are retrieved through retrievers. Since the retrieval phase is not our focus and to ensure fair comparisons with other work, we utilized pre-processed public datasets. Specifically, we apply the ALCE-ASQA benchmark provided by Gao et al. (2023a), and Natural Questions datasets pre-processed by Izacard and Grave (2020). It is worth noting that the retrieval quality is not perfect, with a Recall@5 (R@5) of less than 1. The details of datasets can be found in the original paper or Appendix A.

Models Due to cost considerations, we only use LLaMA2-7B, LLaMA2-13B (Touvron et al., 2023) and Vicuna-13B (Chiang et al., 2023) for experiments, from which not only can we see the impact of different scales of the same model, but we can also see the impact of whether the model has been supervised finetuned.

Metrics Our primary evaluation metric is the quality of answers, which is assessed by checking whether the gold answers (provided by the dataset) are exact substrings of the generation (Gao et al., 2023a). We do not use exact match scores between generated answers and gold answers as metrics because our experiments are zero-shot settings and our language models possess certain expansion abilities (especially Vicuna-13B). They tend to generate sentences rather than single words to answer the question. Therefore, metrics that check substrings are more applicable and indicative, denoted as "str-em" for further clarification.

Model	Decoding	ASQA	NQ
LLaMA2-13B	RD-closed	10.53	20.99
	RD-full	13.29	25.37
	RD-single	13.48	24.09
	CAD	14.39	25.00
	Z-dynamic	14.93	24.90
	Our-fixed	**16.51**	27.06
	Our-dynamic	16.18	**27.86**
Vicuna-13B	RD-closed	26.68	34.85
	RD-full	36.94	56.34
	RD-single	27.51	46.84
	CAD	37.96	56.92
	Z-dynamic	28.91	48.78
	Our-fixed	38.24	**57.67**
	Our-dynamic	**38.68**	56.98
LLaMA2-7B	RD-closed	9.28	17.26
	RD-full	12.41	21.05
	RD-single	12.30	18.06
	CAD	14.73	19.29
	Z-dynamic	14.61	17.25
	Our-fixed	15.42	21.18
	Our-dynamic	**15.85**	**21.96**

Table 1: Str-em results on ALCE-ASQA and NQ benchmark under zero-shot setting. RD-closed, RD-full, RD-single corresponds to Regular Decoding without documents, with all documents concatenated, with single retrieval document. Z-dynamic refers to work of Zhao et al. (2024). Our-fixed means fixed β and γ while Our-dynamic refers to dynamic β and γ.

Baselines We propose a new decoding strategy, so we mainly compare our methods with other decoding methods, such as regular decoding, CAD (Shi et al., 2023b) and work of Zhao et al. (2024). There are various variants for regular decoding, corresponding to decoding based on input without a document, with all documents, and with a single document, which we denote as "RD-closed, RD-full, RD-single". The single document is retrieved from the retriever and ranked first. There are also two variants for work of Zhao et al. (2024), corresponding to decoding with fixed weight and the dynamic weight, and we only consider dynamic weight and denote it as "Z-dynamic".

The number of documents N is set to 5 and K

is set to 10 in our main experiments. The influence of these important hyperparameters is explored in section 5.1 and 5.3. To ensure a fair comparison, all decoding methods differ only in their inputs or adjustments to logits. Subsequent token sampling methods based on the logits remain the same, with the temperature set to 1 as Gao et al. (2023a). Additional experimental details, such as the prompt template and the setting of rest hyperparameters, can be found in Appendix B.

4.2 Main Results

The results are presented in table 1. From the results, we can see that: (1) Our proposed decoding strategy, DVD, consistently outperforms other decoding methods with both fixed and dynamic weights across all models. (2) Our method with fixed and dynamic weights shows comparable performance, consistent with findings from Zhao et al. (2024)'s work. While the fixed weight approach exhibits better performance in certain instances, the dynamic weight approach outperforms it in others. The impact of these weights is further explored in section 5.2. (3) For the ASQA dataset, we use retrieval results coming from DPR without reranking, causing irrelevant information to potentially interfere with the model. Vicuna-13B, which has undergone further fine-tuning, can better utilize contextual information and mitigate irrelevant influences. Therefore, the RD-full result is superior to RD-single for Vicuna-13B, while RD-single outperforms RD-full for LLaMA2-13B and LLaMA2-7B. Additionally, Zhao et al. (2024)'s work (Z-dynamic) only considers a pair of documents and uses their difference to adjust final logits, making it a slight improvement compared to RD-single. In contrast, CAD applies the difference between logits with and without documents, making it more similar to RD-full. Our method demonstrates universality and can achieve better results after the incorporation of all knowledge and dynamical enhancement of knowledge from selected documents. (4) For the NQ dataset, the zero-shot setting and irrelevant retrieved passages pose challenges as well. RD-full achieves better performance compared to RD-closed and RD single, indicating reduced influence from irrelevant documents. The fine-tuned Vicuna-13B still achieves great performance under a zero-shot setting. As said before, Z-dynamic and CAD are slight improvements to RD-single and RD-full correspondingly. We retain most hyperparameters used in sampling same with the ASQA dataset for simplicity, such as the number of new tokens, temperature, and sample method, which indicates that there may still be room for performance improvement. However, given that all decoding strategies employ the same sampling method, our method consistently outperforms other decoding methods with both fixed and dynamic weights.

Selection Criteria	Weight	ASQA
Our DVD	fixed	16.51
	dynamic	16.18
Random	fixed	14.22
	dynamic	13.42
Retrieval	fixed	16.13
	dynamic	15.83

Table 2: Str-em results on ALCE-ASQA with LLaMA2-13b on zero-shot setting of different selection criteria. Selection Criteria refer to different methods to choose z_l and z_h. Fixed weight and dynamic weight refer to fixed or dynamic β and γ. See details in section 3.2.

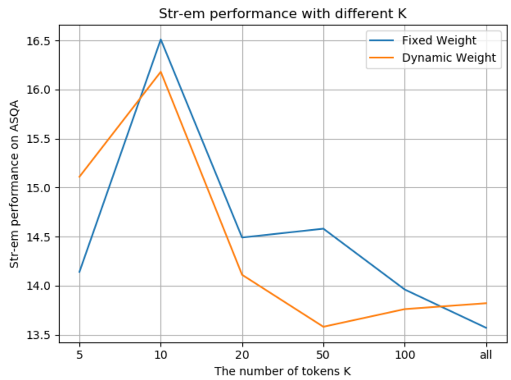

Figure 2: Str-em performance with different K. K is the number of tokens.

5 Analysis

In this section, we conduct experiments from various perspectives to explore the factors that affect out method and demonstrate its efficiency. We mainly present the results of LLaMA2-13B on the ALCE-ASQA benchmark for better illustration.

5.1 Selection Criteria

In section 3.2, we propose to use the entropy of head tokens with top K probability to assess the logits and choose the logits that are worth amplifying (i.e., z_l and z_h). To demonstrate the efficiency of this selection criteria, we compare it with other selection criteria for choosing z_l and z_h, such as choosing randomly and choosing based on the

Weight	Confidence	Calculation	ASQA
Fixed	No need for confidence	$\gamma = 0.1$	14.60
		$\gamma = 0.2$	16.19
		$\gamma = 0.4$	**16.51**
		$\gamma = 0.6$	14.56
		$\gamma = 0.8$	14.76
		$\gamma = 1.0$	16.24
Dynamic	$C_i = p(y_t^1\|z_i)$	$\gamma = C_l$	15.69
		$\gamma = (C_l + C_h)/2$	15.01
		$\gamma = \mathtt{max}(C_l - C_h, 0)$	13.74
	$C_i = p(y_t^1\|z_i) - p(y_t^2\|z_i)$	$\gamma = C_l$	14.46
		$\gamma = (C_l + C_h)/2$	15.74
		$\gamma = \mathtt{max}(C_l - C_h, 0)$	16.18

Table 3: Str-em results on ALCE-ASQA with LLaMA2-13b on the zero-shot setting of different calculation of weight γ. Fixed weight approach doesn't require confidence. Dynamic weight approaches have many variants based on the calculation of confidence and weight.

ranking of the retrieval system. The results are presented in table 2.

The results show that: (1) Our method outperforms static selection criteria, such as random selection or selection based on retrieval ranking. (2) Using the ranking of the retrieval system directly to select the logits and amplify knowledge also yields great improvement compared to results in table 1, while choosing randomly leads to inferior results. This indicates the effect of our motivation, amplifying knowledge from specific documents dynamically selectively during the incorporation of all documents can help the model generate better answers. While the retrieval system can offer insights into selecting certain documents compared to random selection, choosing the document with the highest retrieval ranking is not always the optimal choice.

In addition to comparison with static selection criteria, we also explore the influence of the number of tokens K. K determines the calculation range of entropy, ranging from a few head tokens to all tokens. We conduct experiments with different K and present the outcomes in the figure 2.

"All" refers to using all tokens to calculate the entropy, which is equivalent to regular entropy. The results align with our motivations that the overall entropy, impacted by the meaningless probability of numerous tokens, may not adequately represent the quality of distribution in autoregressive-style LLMs. Head tokens with high probability deserve more attention and can serve as good indicators for documents worth amplifying. The number of tokens considered impacts the performance of both fixed and dynamic weights, as it affects the selection criteria across different logits. In our experiments on ALCE-ASQA and NQ benchmark with a series of LLaMA2 and Vicuna models, the best performance is achieved when the number of tokens K is set to 10. However, the optimal K may vary depending on the characteristics of the dataset and language models, necessitating additional experiments to determine the ideal value.

5.2 The Design of Weight

In addition to selection criteria, the value of weight also impacts the final adjustment of logits that are used to sample tokens. β is related to the influence of internal parametric knowledge, while γ is related to the influence of knowledge from selected knowledge. Since the former is well studied in previous work (Li et al., 2023; Shi et al., 2023b), we mainly discuss the influence of different implementations of γ in this section.

The value of γ can either be a static hyperparameter or determined dynamically during the generation phase, as discussed in section 3.2. For static approaches, we conduct experiments with different fixed values of γ and present the results in table 3. For dynamic approaches, the calculation process involves model confidence. We apply the difference in probability between the top 2 tokens as the confidence, as demonstrated in equation 10 in section 3.2. Previous researches often use the highest probability directly as the confidence, which can be presented in an official formula as $C_i = p(y_t^1|z_i)$. We also conduct experiments to compare these two implementations.

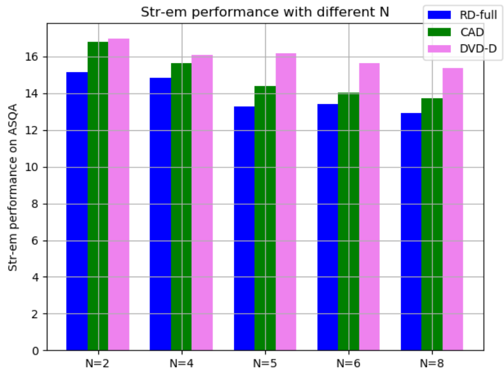

Figure 3: Str-em performance with different N. N is the number of documents.

After the calculation of model confidence, how to use confidence to determine the weight is also an important issue, leading to various calculation variants. We apply the difference of confidence as weights as shown in equation 11 and 12. There are also variants like using the average confidence ($\gamma = (C_l + C_h)/2$) or using only the confidence that needs to be emphasized (Zhao et al., 2024) ($\gamma = C_l$). We conduct experiments on all variants and present the results in table 3.

The results show that: (1) The value of γ significantly impacts performance. The optimal value of γ depends not only on language models but also on the retrieval system. If the overall quality of retrieval is high, the model should prioritize the concatenation of all documents. Conversely, if the overall retrieval quality is low and irrelevant documents are present, the model should amplify specific knowledge and focus on particular documents. In our experiments on the ALCE-ASQA benchmark, γ is set to 0.4 for LLaMA2-13B to get better performance. (2) For dynamic approaches, while many variants lead to great performance compared to results in table 1, our design of $C_i = p(y_t^1|z_i) - p(y_t^2|z_i)$ and $\gamma = \mathtt{max}(C_l - C_h, 0)$ outperforms other variants. This finding aligns with previous research about using the difference in probability between the top 2 tokens as confidence (Wang and Zhou, 2024; Xiang et al., 2024), and is consistent with rationality that C_l and C_h should jointly determine the weight. The design of $C_i = p(y_t^1|z_i)$ and $\gamma = C_l$ also performs well compared to results in table 1 and those of fixed approaches, making it applicable when speed and computational efficiency are prioritized. While there are more designs and combinations for confidence and weight calculation, they are beyond the focus of this paper.

5.3 The Number of Documents

Our work concentrates on multi-document scenarios and construct the input for every document as said in section 3.2. We investigate our method with different values of N to demonstrate its effectiveness in a broader range of situations. For simplicity, we utilize the dynamic weight approach to represent our method. We primarily compare our method with RD-full and CAD, as they apply to various document scenarios and serve as strong baselines. The results are presented in figure 3.

The results show that our proposed method can outperform regular decoding and CAD across different number of documents. As the number of documents N increases, the interference of irrelevant information for LLM is also increasing, while our method that amplifying knowledge from specific documents can consistently be helpful.

6 Conclusion

In this paper, we propose a decoding strategy that can amplify knowledge from the selected documents during the generation phase to adjust the final logits used to sample answer tokens. We construct the inputs batch-wise with different templates and instructions, and get corresponding logits from LLM. We design a new selection criteria that computes the entropy of head tokens with high probability to assess the logits and choose the ones that worth amplifying. The contrastive decoding is used to adjust the logits, where the weights are calculated based on logits dynamically during the generation phase.

We explore several selection criteria and calculation of weights to demonstrate the efficiency of our design. Extensive experiments show that DVD makes consistent improvement on downstream performance and is superior to other decoding strategies, such as regular decoding and CAD. DVD explores the usage of contrastive decoding under the setting of multi-documents, making the incorporation process of knowledge more diverse.

In conclusion, our method propose a new decoding strategy to incorporate knowledge in a more discriminative way under the multi-document setting. Our method is plug-and-play and doesn't require any training, and it can be combined with other orthogonal methods to improve the overall performance of the RAG system.

Limitations

Our work has the following limitations:

(1) Our method is applied on the logit level, necessitating access to each logit in the batch, and subsequently adjusts the final logits used for sampling answer tokens. Consequently, its applicability may be limited to white-box models that are open-source and offer access to such information. Closed-source models, such as ChatGPT, GPT4, and others, may not be compatible with our method due to the lack of access to the underlying logits.

(2) We propose to construct the input in a batch with different templates and instructions, which can help LLM consider multiple inputs simultaneously and incorporate all kinds of knowledge including internal parametric knowledge and external non-parametric knowledge from documents. However, this methodology may result in increased resource utilization during inference, particularly in terms of hardware consumption. Actual hardware consumption is directly proportional to the size of batch, i.e. the number of documents. Therefore, our method may require lots of resources when applied in situation where the number of documents exceeds 10. To mitigate this limitation, alternative batch construction methods can be explored. For instance, concatenating two or more documents into a single input within the batch may reduce memory consumption. However, it's important to note that this approach may compromise the accuracy of document selection.

(3) In this paper, we only consider limited situations such as zero-shot muli-document QA setting and models up to 13B due to the cost consideration. We will also conduct experiments to test our method on a wider range of application scenarios in the future, such as few-shot settings, bigger models, and more kinds of datasets. While our approach has demonstrated the effectiveness of amplifying knowledge from specific documents during the generation phase, it's important to acknowledge the existence of various other selection criteria and fusion methods. Further investigation into these alternatives may yield additional performance improvements.

References

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Benjamin Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Sim'on Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Raphael Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Lukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Ryan Kiros, Matthew Knight, Daniel Kokotajlo, Lukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Adeola Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel P. Mossing, Tong Mu, Mira Murati, Oleg Murk, David M'ely, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Ouyang Long, Cullen O'Keefe, Jakub W. Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alexandre Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Pondé de Oliveira Pinto, Michael Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario D. Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin D. Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas A. Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cer'on Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll L. Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023a. Acl 2023 tutorial: Retrieval-based language models and applications. *ACL 2023*.

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

Hung-Ting Chen, Michael J.Q. Zhang, and Eunsol Choi. 2022. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In *Conference on Empirical Methods in Natural Language Processing*.

Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. 2023. Lift yourself up: Retrieval-augmented text generation with self memory. *ArXiv*, abs/2305.02437.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023a. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey. *ArXiv*, abs/2312.10997.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ArXiv*, abs/2311.05232.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint*.

Xuhui Jiang, Yuxing Tian, Fengrui Hua, Chengjin Xu, Yuanzhuo Wang, and Jian Guo. 2024. A survey on large language model hallucination via a creativity perspective. *ArXiv*, abs/2402.06647.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, Xiaojian Jiang, Jiexin Xu, Li Qiuxia, and Jun Zhao. 2024a. Tug-of-war between knowledge: Exploring and resolving knowledge conflicts in retrieval-augmented language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16867–16878, Torino, Italia. ELRA and ICCL.

Zhuoran Jin, Pengfei Cao, Hongbang Yuan, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. 2024b. Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models. *ArXiv*, abs/2402.18154.

Taehyeon Kim, Joonkee Kim, Gihun Lee, and Se-Young Yun. 2023. Instructive decoding: Instruction-tuned large language models are self-refiner from noisy instructions. *arXiv preprint arXiv:2311.00233*.

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.

Moxin Li, Wenjie Wang, Fuli Feng, Fengbin Zhu, Qifan Wang, and Tat-Seng Chua. 2024. Think twice before assure: Confidence estimation for large language models through reflection on multiple answers. *ArXiv*, abs/2403.09972.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023a. Query rewriting for retrieval-augmented large language models. *ArXiv*, abs/2305.14283.

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

Shiyu Ni, Keping Bi, J. Guo, and Xueqi Cheng. 2024. When do llms need retrieval augmentation? mitigating llms' overconfidence helps retrieval augmentation. *ArXiv*, abs/2402.11457.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Lidén, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *ArXiv*, abs/2302.12813.

Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3781–3797, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Huai hsin Chi, Nathanael Scharli, and Denny Zhou. 2023a. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen-tau Yih. 2023b. Trusting your evidence: Hallucinate less with context-aware decoding. *arXiv preprint arXiv:2305.14739*.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Victoria Lin, Noah A. Smith, Luke Zettlemoyer, Scott Yih, and Mike Lewis. 2023c. In-context pretraining: Language modeling beyond document boundaries. *ArXiv*, abs/2310.10638.

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

Jianlin Su. 2023. Naive bayes-based context extension. `https://github.com/bojone/NBCE`.

Hexiang Tan, Fei Sun, Wanli Yang, Yuanzhuo Wang, Qi Cao, and Xueqi Cheng. 2024. Blinded by generated contexts: How language models merge generated and retrieved contexts for open-domain qa? *ArXiv*, abs/2401.11911.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. 2023. Freshllms: Refreshing large language models with search engine augmentation. *ArXiv*, abs/2310.03214.

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. *ArXiv*, abs/2402.10200.

Chong Xiang, Tong Wu, Zexuan Zhong, David Wagner, Danqi Chen, and Prateek Mittal. 2024. Certifiably robust rag against retrieval corruption.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*.

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

Rongwu Xu, Zehan Qi, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge conflicts for llms: A survey. *CoRR*, abs/2403.08319.

Haoyan Yang, Zhitao Li, Yong Zhang, Jianzong Wang, Ning Cheng, Ming Li, and Jing Xiao. 2023. Prca: Fitting black-box large language models for retrieval question answering via pluggable reward-driven contextual adapter. In *Conference on Empirical Methods in Natural Language Processing*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

Yue Zhang, Leyang Cui, Wei Bi, and Shuming Shi. 2023. Alleviating hallucinations of large language models through induced hallucinations. *ArXiv*, abs/2312.15710.

Zheng Zhao, Emilio Monti, Jens Lehmann, and Haytham Assem. 2024. Enhancing contextual understanding in large language models through contrastive decoding. *arXiv preprint arXiv:2405.02750*.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed Chi, Quoc Le, and Denny Zhou. 2023. Take a step back: Evoking reasoning via abstraction in large language models. *ArXiv*, abs/2310.06117.

Shengyao Zhuang, Bing Liu, Bevan Koopman, and G. Zuccon. 2023. Open-source large language models are strong zero-shot query likelihood models for document ranking. *ArXiv*, abs/2310.13243.

A Details about Datasets

In this paper, we use ALCE-ASQA and NQ benchmark to evaluate our method.

ALCE-ASQA is proposed by Gao et al. (2023a). There are many variants about this dataset. We choose the one retrieved by DPR without reranked oracle retrieval results (asqa_eval_dpr_top100.json in their repository[3]). There are 948 evaluation samples. And we use their official eval code in the repository to evaluate our generated answers. See their repository for more details.

Natural Questions (NQ) is a popular QA dataset proposed by Kwiatkowski et al. (2019) and is widely used in many open-domain researches. The retrieval system affect downstream performance. Therefore, we use the retrieval results and preprocessed NQ dataset from Izacard and Grave (2020) directly for simplicity. Since our work focus on zero-shot multi-document setting, we only use the test set with 3610 samples. According to their repository[4], the R@5 value is 73.8, making it suitable for our experiments that aim at improving performance under irrelevant interfere.

B Experimental Details

We provide more details about our experiments in this section.

First, the prompt templates we use in the experiments are diverse. As for ALCE-ASQA benchmark, we apply asqa_closedbook.json as T_1 for input without document and apply asqa_default.json as T_2 and T_3 for input with all documents and single document. Both files are provided by original work (Gao et al., 2023a), and we apply their prompts directly to avoid the influence of different templates. One of the example of our constructed input based on these template is presented in table 4. As for NQ benchmark, we apply simple prompt, `"Question: {question} \n Answer:"` for closed-book setting, and `"Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant). \n\n {documents} \n\n Question: {question} Answer:"` for multi-documents setting, where documents are also formatted as `"Document [{document.index}](Title: {document.title}) {document.text}"`.

Then, we will list the settings of hyperparameters we used in the experiments. The seed is set to 42. The generation configuration includes, temperate is set to 1, the value of top_p is set to 0.95 and the number of max_new_tokens is 300. The value of β is set to 0.25 for all models in the setting of fixed weight. The perfect value of γ is different for every model, which is 0.1, 0.4 and 0.2 for vicuna 13b, llama2 13b and llama2 7b correspondingly.

[3] https://github.com/princeton-nlp/ALCE

[4] https://github.com/facebookresearch/FiD

An input instance on the ALCE-ASQA dataset	
x_1	Instruction: Write an accurate, engaging, and concise answer for the given question. Use an unbiased and journalistic tone.\n\n Question: Who has the highest goals in world football? \n\n Answer:
x_2	Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided search results (some of which might be irrelevant) and cite them properly. Use an unbiased and journalistic tone. Always cite for any factual claim. When citing several search results, use [1][2][3]. Cite at least one document and at most three documents in each sentence. If multiple documents support the sentence, only cite a minimum sufficient subset of the documents. \n\n Question: Who has the highest goals in world football? \n\n Document [1](Title: FIFA World Rankings) FIFA World Rankings The FIFA World Ranking is a ranking system for men's national teams in association football, ... \n Document [2](Title: FIFA World Rankings) based on the importance of the match and the strength of the opponent. ... \n Document [3](Title: FIFA World Rankings) The 19 July 2018 release was cancelled following the new calculation method implementation. ... \n Document [4](Title: World Football Elo Ratings) Ukraine 26 years, and for Montenegro 11 years. For Croatia and Slovakia th ... \n Document [5](Title: FIFA World Ranking system (2006–2018)) match status multipliers are as follows: A win against a very highly ranked opponent is a considerably great... \n Answer:
x_3	Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided search results (some of which might be irrelevant) and cite them properly. Use an unbiased and journalistic tone. Always cite for any factual claim. When citing several search results, use [1][2][3]. Cite at least one document and at most three documents in each sentence. If multiple documents support the sentence, only cite a minimum sufficient subset of the documents. \n\n Question: Who has the highest goals in world football? \n\n Document [1](Title: FIFA World Rankings) FIFA World Rankings The FIFA World Ranking is a ranking system for men's national teams in association football, currently led by Belgium.... \n Answer:
x_4	Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided search results (some of which might be irrelevant) and cite them properly. Use an unbiased and journalistic tone. Always cite for any factual claim. When citing several search results, use [1][2][3]. Cite at least one document and at most three documents in each sentence. If multiple documents support the sentence, only cite a minimum sufficient subset of the documents. \n\n Question: Who has the highest goals in world football? \n\n Document [2](Title: FIFA World Rankings) based on the importance of the match and the strength of the opponent. ... \n Answer:
x_5	...
x_6	...
x_7	Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided search results (some of which might be irrelevant) and cite them properly. Use an unbiased and journalistic tone. Always cite for any factual claim. When citing several search results, use [1][2][3]. Cite at least one document and at most three documents in each sentence. If multiple documents support the sentence, only cite a minimum sufficient subset of the documents. \n\n Question: Who has the highest goals in world football? \n\n Document [5](Title: FIFA World Ranking system (2006–2018)) match status multipliers are as follows: A win against a very highly ranked opponent is a considerably great... \n Answer:

Table 4: One instance of our constructed input based on templates of ALCE-ASQA. The batch consists of x_1, ... ,x_7 together and fed into LLM.