Grounding Language Model with Chunking-Free In-Context Retrieval

Anonymous ACL submission

Abstract

This paper presents a novel Chunking-Free In-Context (CFIC) retrieval approach, specifically tailored for Retrieval-Augmented Generation (RAG) systems. Traditional RAG systems often struggle with grounding responses using precise evidence text due to the challenges of processing lengthy documents and filtering out irrelevant content. Commonly employed solutions, such as document chunking and adapting language models to handle longer contexts, have their limitations. These methods either disrupt the semantic coherence of the text or fail to effectively address the issues of noise and inaccuracy in evidence retrieval.

CFIC addresses these challenges by circumventing the conventional chunking process. It utilizes the encoded hidden states of documents for in-context retrieval, employing auto-aggressive decoding to accurately identify the specific evidence text required for user queries, eliminating the need for chunking. CFIC is further enhanced by incorporating two decoding strategies, namely Constrained Sentence Prefix Decoding and Skip Decoding. These strategies not only improve the efficiency of the retrieval process but also ensure that the fidelity of the generated grounding text evidence is maintained. Our evaluations of CFIC on a range of open QA datasets demonstrate its superiority in retrieving relevant and accurate evidence, offering a significant improvement over traditional methods. By doing away with the need for document chunking, CFIC presents a more streamlined, effective, and efficient retrieval solution, making it a valuable advancement in the field of RAG systems.

1 Introduction

Recently, retrieval-augmented generation (RAG) has marked a significant advancement in the field of natural language processing (NLP). This technique has demonstrated remarkable effectiveness in reducing hallucination in text generation (Ji et al., 2023), particularly in knowledge-intensive tasks like open-domain question answering (Wang et al., 2019; Lewis et al., 2020; Shuster et al., 2021a; Komeili et al., 2022). An RAG system typically consists of two components: the retriever and the generator. Given an input query, the retriever first identifies relevant evidence text, upon which the generator then generates the answer.

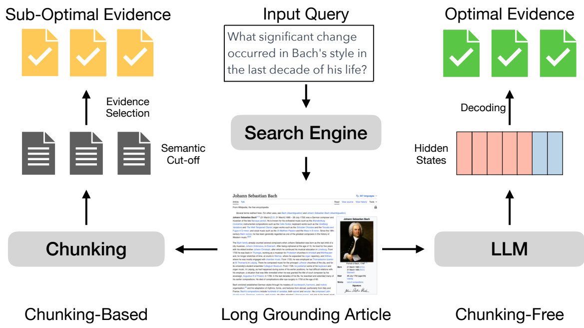

Figure 1: Comparison of Chunking-Based and Chunking-Free Methods. The left panel illustrates the chunking-based method, involving chunking a lengthy document into smaller passages followed by refinement through passage ranking. The right panel depicts the chunking-free method proposed in this paper, where grounding text is directly decoded by LLMs without the need for document chunking.

The generator's output should be grounded by precise evidence text obtained by the retriever. However, this poses challenges for most retrieval systems, as they often retrieve lengthy documents such as web pages. In practice, we only need specific grounding text from these documents to help answer user queries. Using lengthy documents directly in the RAG system presents two difficulties. First, generation models may struggle to handle the extensive length of these documents. Second, irrelevant or distracting content within the documents can lead the model astray from the main query, resulting in inaccurate response generation (Gao et al., 2024).

To address this issue, common approaches involve chunking documents into smaller passages and employing strategies like reranking for relevance (Nogueira and Cho, 2020; Mao et al., 2021; Gao et al., 2024), or selecting passages based on other measurements (Asai et al., 2022; Jiang et al., 2023). However, the chunking process is often suboptimal, as determining the granularity of the passage chunking is challenging. Improper chunking can disrupt the semantics and result in incomplete and incoherent retrieved information (Dong et al., 2023). Another method involves adapting large language models (LLMs) to process longer contexts by training them on long contexts or implementing a sliding context window (Ratner et al., 2022; Xu et al., 2023a; Chen et al., 2023). While these methods enable LLMs to handle longer texts, they do not fully address the issue of noise in the lengthy documents and cannot output the grounding text for the generated response (Kaddour et al., 2023).

In this paper, we propose a Chunking-Free In-Context (CFIC) retrieval approach aimed at helping the RAG system mitigate information bias introduced by document chunking and irrelevant noisy text. Specifically, given an input query and a long grounding document, instead of refining the long documents with a chunking-based method, we leverage the document's encoded hidden states to perform Chunking-free In-Context Retrieval. It circumvents the traditional chunking process, allowing the retrieval system to auto-aggressively decode and pinpoint the precise evidence text to ground the response generation to a query. Figure 1 shows the comparison between the chunking-based method and the chunking-free method for grounding text retrieval. The chunking-free method demonstrates a superior ability to identify optimal evidence text, as it considers the entire document for a comprehensive perspective.

Concretely, CFIC involves encoding a document into transformer hidden states. When a user query is input, CFIC continues to encode the query alongside task instructions following the hidden states, subsequently generating grounding text. In practice, we can cache the documents' hidden states to further reduce computation[1]. Given the expectation for CFIC to process lengthy documents, it becomes imperative to adapt CFIC for handling long contexts. Considering the trade-off between efficiency and effectiveness, in this paper, we adapt CFIC to accommodate a 32k context, utilizing LLAMA2-7B-chat as the foundational model. To achieve this, we construct a dataset containing long document, user query and precise text evidence to training the foundation model via Supervised Fine-Tuning (SFT).

Despite its promise, CFIC encounters two major challenges: (1) **Efficiency issue**: the auto-aggressive generation process involves executing attention interactions for generating each new token, a procedure that becomes particularly time-consuming with longer contexts due to the management of exponentially larger attention matrices. This process requires substantial computational resources (Kaplan et al., 2020), and (2) **Faithfulness issue**: it is challenging to ensure the generation model's output remains faithful to the original input context, given its open-ended decision boundary (Li et al., 2022b). To address these, we propose two decoding strategies that accelerate inference and ensure that generated text evidence originates from the corpus. These include: (1) utilizing sentence prefixes as decoding candidates to shift the model's decision boundary from open-ended to document-dependent generation and (2) upon locating the appropriate sentence prefix, bypassing the decoding of intermediate tokens and directly selecting sentence ends with the highest likelihood of the [eos] token, thereby terminating the generation. Furthermore, to retrieve multiple text spans as evidence, we sample several sentence prefixes with the best likelihood as candidates and rank them by sequence likelihood. By this means, CFIC not only enhances the relevance and accuracy of retrieved evidence text but also preserves the semantic integrity of the information, effectively addressing major drawbacks of current retrieval systems.

We tested CFIF on the LongBench tasks (Bai et al., 2023) including: (1) single-document question answering with datasets like NarrativeQA, Qasper, MulitfieldQA, and (2) multi-document QA with datasets like Musqus and HotpotQA. The experiment results verify the effectiveness of our method. In summary, our contributions are as follows: (1) we propose a chunking-free in-context retrieval method dedicated to the RAG system, aiding in locating precise text evidence to answer user queries; (2) we propose the CFIC model of which the ability to find text evidence from long context is enhanced via Supervised Fine-Tuning with self-constructed dataset; (3) we design two decoding

[1]In a single-sided transformer model, the forward side is auto-regressive; once an output token's hidden state is computed, it remains unchanged for subsequent forward steps, allowing us to use these encoded states as a cache.

strategies that significantly improve the efficiency and accuracy of the CFIC's decoding process.

2 Related Work

The RAG framework, initially introduced in the works of Lewis et al. (2020), aimed to enhance language models' capacity for generating knowledge-based responses(Izacard and Grave, 2020; Chung et al., 2022; Yang et al., 2023). Subsequent research primarily focus on refining the RAG's two core components. On the retrieval front, significant strides have been made towards more efficient and precise retrieval methods (Khandelwal et al., 2019; Nishikawa et al., 2022; Li et al., 2022a; Kang et al., 2023). For example, the arise of Dense Passage Retrieval significantly surpasses traditional sparse dense (Karpukhin et al., 2020). Parallel efforts on the generation side have concentrated on fine-tuning generative models to better harmonize with retrieved information, a notable example being the work of Izacard and Grave (2021b) in optimizing external knowledge utilization (Izacard and Grave, 2021a; Chung et al., 2022; Kamalloo et al., 2023).

Nevertheless, RAG encounters specific challenges, especially in managing lengthy and complex retrieved documents. Researchers, including Mao et al. (2021), have developed chunking and reranking techniques to enhance passage relevance. Furthermore, Guu et al. (2020) introduced methods for jointly learning retriever and generator models, thereby improving the coherence and relevance of outputs. Addressing the issue of lengthy contexts in RAG has involved either refining contexts (Li et al., 2022a; Jiang et al., 2023) or adapting generation models to handle extended contexts (Xu et al., 2023a; Ratner et al., 2022; Chen et al., 2023).

Recent advancements in RAG predominantly incorporate large-scale language models (LLMs), such as GPT-3 and GPT-4, to augment language processing capabilities (Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023a; Google, 2023). The integration of LLMs has paved the way for more contextually rich and nuanced generation, especially in aligning generated responses with human preferences (Ram et al., 2023; Liu et al., 2023b). In RAG systems employing LLMs, the accuracy of retrieved textual evidence is crucial for reducing hallucinations and incorporating external knowledge (Shuster et al., 2021b; Zhang et al., 2023b; Yao et al., 2023; Bang et al., 2023). However, the challenge of processing long and noisy contexts persists (Liu et al., 2023a; Li et al., 2022a; Xu et al., 2023b). This paper introduces a chunking-free in-context retrieval approach that leverages transformer hidden states to generate grounding text evidence, treating evidence retrieval as a generative process. This method represents a more streamlined and efficient retrieval solution for RAG systems, marking a significant advancement over previous retrieval methodologies.

3 Method

3.1 Preliminary

In a RAG system, the system takes a user query q as input, retrieves text evidence $\mathcal{A}$ from a text corpus $\mathcal{C}$ using a retriever $\theta(\cdot)$ as external knowledge, and utilizes a generation model $\phi(\cdot)$ to produce the final response $\mathcal{T}$. This pipeline can be formalized as:

$$\mathcal{A} = \theta(q, \mathcal{C}), \quad \mathcal{T} = \phi(q, \mathcal{A}). \tag{1}$$

The retriever $\theta(\cdot)$ can be either a standalone retriever (*e.g.*, DPR (Karpukhin et al., 2020)) or a commercial search engine (*e.g.*, Google), and the generation model $\phi(\cdot)$ is usually a trained LM. Based on Eq. (1), the quality of the generated text $\mathcal{T}$ is bounded by the accuracy of the evidence $\mathcal{A}$, emphasizing the importance of accurately finding the accurate text evidence.

In practice, most RAG systems' retrievers cannot accurately find exact text evidences, but only retrieve lengthy documents (*e.g.*, web pages or pre-indexed articles) that contain the evidences. As mentioned in Section 1, such lengthy documents might bias the generated content. Thus, given the retrieved evidence $\mathcal{A}$, we usually select a few useful text spans, called supporting text evidence $\mathcal{P} = \{p_1, \cdots, p_k\} \in \mathcal{A}$, to support the answer generation for the input query q in a RAG system.

We define the process of finding supporting passages as a mapping function $f(\cdot)$:

$$\mathcal{P} = \{p_1, \cdots, p_k\} = f(\mathcal{A}). \tag{2}$$

The mapping function $f(\cdot)$ can take various forms, such as chunking the text evidence $\mathcal{A}$ and prioritizing relevant chunks through re-ranking. In this paper, we define the mapping function $f(\cdot)$ as a generation process in which we directly generate the supporting text evidence $\mathcal{P}$ conditioned on the transformer hidden-states $\mathbf{h} = \texttt{Trans}(\mathcal{A})$ of the lengthy document:

$$\mathcal{P} = f(\mathcal{A}) \sim \texttt{Generator}(\mathcal{P}|\mathbf{h}, q). \tag{3}$$

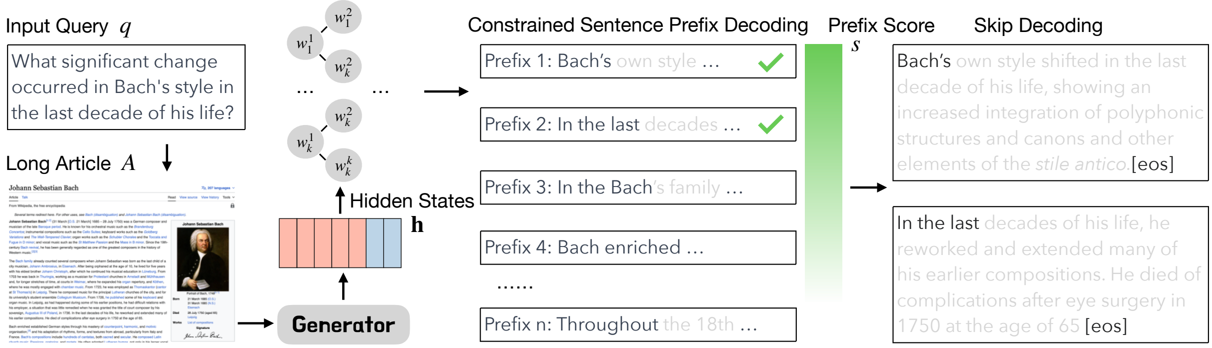

Figure 2: Overview of the proposed method: CFIC. The middle part shows the Constrained Sentence Prefix Decoding strategy which ensures the generated text prefixes originate from the input article. The right part shows the Skip Decoding strategy which bypasses decoding the intermediate tokens while terminating generation at the position with the best likelihood of [eos] token. Gray tokens in the figure are bypassed during generation.

Compared to regular auto-regressive decoding, the above process is characterized by the fact that the generation target $\mathcal{P}$ contains text sourced from $\mathcal{A}$. This means that once we determine the decoding prefix, we can skip the intermediate tokens and directly find the terminating position by computing the probability of inserting [eos] token. This greatly improves inference efficiency while ensuring that the generated text accurately represents the source text. Additionally, a single supporting passage may not always be sufficient for question answering. Therefore, we can obtain multiple sentence prefixes as top-k candidates using sampling decoding. In this paper, our proposed model CFIC applies these ideas to generate the top-k supporting text evidence $\mathcal{P}$, which are further discussed in the following sections.

3.2 The Proposed Model: CFIC

Figure 2 presents an overview of our proposed model, CFIC. The process begins with CFIC receiving a user query. It then retrieves a long article as grounding evidence through a search engine (e.g., Google). Subsequently, CFIC combines the long document and the query into an input prompt, following the format outlined in Table 2. This input prompt is encoded into hidden states. Based on these hidden states, CFIC first identifies the top-k sentence prefix candidates using the Constrained Sentence Prefix Decoding strategy. This strategy ranks the sentence prefixes considering the generation score (accumulated token log probabilities normalized by token length) of each sentence prefix. CFIC then skips the decoding of intermediate tokens and terminates the generation process by locating the [eos] token position with the highest likelihood (Skip Decoding). Consequently, we obtain k grounding evidence texts that can aid in supporting downstream tasks. It is important to note that this paper primarily focuses on pinpointing precise grounding text evidence within the long document, rather than on the retrieval of the long document. Therefore, we assess our CFIC and all baseline models using the LongBech benchmark, which provides pre-prepared long documents. In the subsequent sections, we will introduce the two proposed decoding strategies and then discuss the training and inference processes of CFIC.

Constrained Sentence Prefix Decoding Normally, the generation process of an auto-aggressive decoding model is as:

$$w_n \sim \prod_{n=1}^{|w|} p(w_n \in \mathcal{V} | w_{<n}, \mathbf{h}), \qquad (4)$$

where $\mathbf{h}$ represents the hidden states of previous tokens. The current token, denoted by w_n, is selected from the entire vocabulary $\mathcal{V}$ of the generation model. In the case of CFIC, the generation target $\mathcal{P}$ consists of text spans that originate directly from the source context. Consequently, it is possible to define a more constrained generation space to ensure the faithfulness of the text produced. Specifically, we suggest employing the prefix of each sentence within the source context as generation constraints. This approach guarantees that the text generated by CFIC can be traced back to the input context. Thus, Eq. (4) can be modified as:

$$w_n \sim \prod_{n=1}^{|w|} p(w_n \in \bar{\mathcal{V}} | w_{<n}, \mathbf{h}), \qquad (5)$$

where $\bar{\mathcal{V}}$ denotes a token set contains each sentence's prefix.

The sentence prefix serves as an position identifier to facilitate the identification of the starting point of a supporting passage within the source context. To select the top-k candidate passages, it is essential to differentiate k distinct sentence prefixes. This is achieved through the constrained top-k sampling decoding, a process that entails selecting the next token w_n from the top-k most likely tokens $\bar{\mathcal{V}}_k \in \bar{\mathcal{V}}$ based on the token's probability, $p(w_n|w_{<n})$. The sampling process terminate once the generated sentence prefixes are capable of uniquely identifying positions in the source context. The number of decoding steps required until termination is denoted by β, resulting in up to k^β prefix candidates after β steps. We denote the generated sentence prefix by b. Subsequently, these prefix candidates are ranked according to the prefix sequence score s, which calculates the normalized accumulated log probability of tokens as follows:

$$s = \frac{1}{|w|}\sum_{n=1}^{|w|} \log p(w_n|w_{<n}). \tag{6}$$

Finally, the k sentence prefixes with the highest scores are selected.

Referring to Figure 2 for illustration, the decoding process initiates by sampling k tokens, such as [*Bach, In, ..., Throughout*], to represent the first set of candidate tokens. Given that multiple sentences in the long article begin with the tokens [*Bach, In*], the decoding of subsequent tokens is necessary. For sentences that start with "*Bach*", the decoding terminates at step $\beta = 2$. And for sentences beginning with "*In*", the decoding ends at step $\beta = 3$. Following this, we retain $k = 2$ sentence prefixes to identify the supporting passages.

Skip Decoding Similarly, since the generation target originates exactly from the source text, once the generation prefix is determined, we can use the generated prefix as a position identifier to locate the original text in the source text. Subsequently, we can bypass decoding the intermediate tokens and directly compute the token probability $p(\text{[eos]})$ for the [eos] token after each sentence following the generated prefix. We select the position with the highest probability as the termination point. In practice, we calculate $p_{\text{[eos]}}$ after each sentence within a predefined token distance d. Formally, given a generated prefix b, we determine the termination position as follows:

$$w^*_{\text{[eos]}} = \arg\max_{l \in \mathcal{L}} p_{\text{[eos]}}(b \oplus l), \quad |l| \le d, \tag{7}$$

where l represents the token sequence following the prefix b with a maximum length of d.

Training and Inference As previously discussed, we define the task of identifying supporting passages from a long source text for grounding downstream tasks as evidence generation. To this end, it is crucial to enhance the generation model with the capability to pinpoint precise textual evidence within extensive texts. In this study, CFIC achieves this through Supervised Fine-Tuning (SFT). We employ a prompt, formed using the pair $(q, \mathcal{A})$ as outlined in Table 2, as the input, and use the text evidence $\mathcal{P}$ as the target for generation. The model is trained using the negative log-likelihood (NLL) loss function:

$$\mathcal{L}(q, \mathcal{A}, \mathcal{P}^*) = -\sum_{n=1}^{|\mathcal{P}^*|} \log p(\mathcal{P}^*_n|\mathcal{P}^*_{<n}, q, \mathcal{A}). \tag{8}$$

The training dataset is introduced in Section 4.1.

During the inference stage, given the input $(q, \mathcal{A})$, we apply Constrained Sentence Prefix Decoding and Skip Decoding strategies to extract k supporting passages. Should these passages exhibit overlapping sections, we amalgamate such intersecting passages into a single cohesive passage. Subsequently, these collated supporting passages are utilized to ground downstream tasks.

4 Experiment

4.1 Datasets and Evaluation Metric

As mentioned above, we train the CFIC model using data that contains $(q, \mathcal{A}, \mathcal{P})$ triplets via SFT. Most current datasets cannot provide such data format. Thus, we use self-constructed SFT data to train the CFIC model, and evaluate all baselines on the LongBench benchmark (Bai et al., 2023). Specifically, to construct the SFT training data, we first collect a corpus of lengthy articles, including Wikipedia articles, novels, and news articles. Subsequently, we randomly select text spans from these articles and ask ChatGPT to generate a query that can be answered by each text span. As for evaluation, we choose five datasets from LongBench including NarrativeQA (Kočiský et al., 2017), Qasper (Dasigi et al., 2021), MultiFieldQA (Bai et al., 2023)), HotpotQA (Yang et al., 2018) and MuSiQue (Trivedi et al., 2022). Following the LongBench benchmark, we use F1-score as the evaluation metric. For further details of Long-

Dataset	SFT	NarrativeQA	Qasper	MultiFieldQA	HotpotQA	MuSiQue
Num of Samples	25,652	200	200	150	200	200
Ave. Length	12,248	18,409	3,619	4,559	9,151	11,214

Table 1: Statistical information of the datasets utilized in this paper, where the average length indicates the word count, typically smaller than the BPE-tokenized token length.

Below is an article, read the article and answer my question after the article. Now the article begins: `{Article}` Now the article ends. *Select several sentences from the article to answer my question.* Question: `{Question}`

Table 2: Prompt template in training and evaluation.

Bench, please refer to Bai et al. (2023). We show the statistical information of all datasets in Table 1.

4.2 Baseline Settings

In this study, we focus on in-context retrieval within the Retrieval-Augmented Generation (RAG) system. As such, we employ stand-alone LLMs as generators. Specifically, we utilize Llama2-7B-chat-4k (Touvron et al., 2023b) and Vicuna-v1.5-7B-16k (Zheng et al., 2023) as our generators. To assess our chunking-free approach against the traditional chunking-based methods, the baseline model settings are as follow:

Chunking-Base Method Chunking-based methods generally commence by segmenting a lengthy document into smaller passages using heuristic strategies, followed by reranking these passages with a ranking model. In our research, we investigate two prevalent chunking strategies: (1). Sliding Window Chunking (SW): This strategy involves dividing the document into sentences and then grouping these sentences into passages. Each passage is designed not to exceed a predefined maximum length of 256 words, with a stride of one sentence. (2). Paragraph-based Chunking (Para): Here, the document is split by paragraph markers (e.g., \n). We employ “bge-large-en-v1.5” (Xiao et al., 2023) and “llm-embedder” (Zhang et al., 2023a) as the ranking models. We utilize the SW and Para strategies to divide the document into passages, which are then reranked by the ranking models. The highest-ranking passages are chosen as the input context for the generators to support the QA tasks.

Chunking-Free Method For the chunking-free models, we present the outcomes using Vicuna-v1.5-7B-16k (Zheng et al., 2023), LongChat-7B-32k (Li et al., 2023), and LongAlpaca-7B-32k (Chen et al., 2023) as baseline models. These models refine lengthy documents into concise text evidence, which then serves as context for generator to support QA tasks. To ensure a fair comparison, all baseline models provide a comparable volume of textual evidence for downstream tasks, maintaining consistency in the number of passages or token length. We also explore the effectiveness of feeding full articles into generators. We introduce our inplementation details in Appendix A.

4.3 Main Results

Table 3 shows the main experiment results which are the performance across different QA tasks using various refined text evidence as context. From the results we have the following findings: **First**, CFIC significantly outperforms other LLMs in chunking-free in-context retrieval tasks as CFIC is specifically optimized to select precise text evidence crucial for grounding QA tasks. This underscores the necessity and effectiveness of supervised fine-tuning (SFT) in adpting the foundation model into the in-context retrieval task. **Second**, Chunking-based methods serve as strong baselines due to their ability to extract passages directly from the source context, whereas LLMs lacking SFT tend to generate content that may not always align faithfully with the source material. CFIC, however, consistently surpass all chunking-based baselines across all datasets, indicating the potentiality of the chunking-free in-context retrieval paradigm. **Last**, Compared to using the entire article as context, our CFIC model significantly improves the performance of QA tasks across most datasets, except for the NarrativeQA dataset. This improvement evidences the critical role of identifying and utilizing the right and precise context in optimizing QA task performance, demonstrating the CFIC model’s efficiency in context filtering and utilization. As for the NarrativeQA dataset, we find that NarrativeQA’s precise text evidence frequently appears at the start of lengthy articles, a location that LLMs tend to prioritize their attention (Liu et al., 2023a).

		Llama2-7B-chat-4k					Vicuna-v1.5-7B-16k				
Model	chunk	nar	qas	mul	hot	mus	nar	qas	mul	hot	mus
BGE	SW	13.9	22.0	34.0	34.0	14.0	12.1	27.3	37.5	33.6	13.5
BGE	Para	12.1	21.7	31.4	31.2	12.3	10.2	23.2	34.7	31.7	12.5
LLM-Embedder	SW	14.1	23.2	34.3	33.8	14.6	13.2	27.4	39.1	31.6	12.6
LLM-Embedder	Para	13.2	21.7	34.1	32.9	12.6	12.3	25.1	36.3	31.1	12.1
Vicuna-7B	-	13.7	19.0	23.3	22.0	9.7	12.3	23.5	24.0	23.8	11.0
LongChat-7B	-	12.2	19.7	29.5	27.9	9.6	11.1	21.9	32.4	30.2	9.7
LongAlpaca-7B	-	12.8	19.3	26.8	28.8	10.3	11.2	21.2	25.2	27.2	10.2
CFIC-7B(Ours)	-	18.3	**27.7**	**41.2**	**34.0**	**14.7**	17.5	**31.0**	**39.8**	**33.8**	**16.2**
Full Article	-	**18.7**	19.2	36.8	32.8	9.4	**19.4**	26.1	38.5	25.3	9.8

Table 3: Main experiment results, which are the QA performance across various datasets, using different refined text evidence as context. Following Bai et al. (2023), we use F1-score as the evaluation metric. The best results are in bold and the secondary results are marked with underline.

This might explain why CFIC does not perform as well on this dataset, given that its approach to identifying precise evidence could inadvertently introduce errors, thereby diminishing its accuracy. In practice, however, that precise text evidence can be located throughout the entire length of an article, not just at the beginning.

	Llama2-7B-chat-4k				
Model	nar	qas	mul	hot	mus
CFIC-7B	18.3	**27.7**	**41.2**	**34.0**	**14.7**
w/o prefix	16.4	26.0	39.3	33.0	12.5
w/o skip	15.8	27.0	37.6	30.1	11.6
w/o both	13.2	20.2	37.4	30.1	9.2
LongAlpaca-7B	12.8	19.3	26.8	28.8	10.3
Full Article	**18.7**	19.2	36.8	32.8	9.4

Table 4: Results of the ablation Study.

4.4 Discussion

Ablation Study To assess the effectiveness of the design of CFIC, we conduct an ablation study by removing key components of the model, including: (1). Removal of Sentence Prefix Decoding Strategy (*w/o* prefix): we remove the constraint of limiting the decoding space to sentence prefixes. Instead, a beam search algorithm was employed to sample short sequences (each comprising 8 tokens) based on the input article. Subsequently, the top-k short sequences were matched back to the input article to identify starting prefixes. (2). Removal of Skip Decoding (*w/o* skip): we dispensed with the practice of bypassing intermediate tokens following the sentence prefix decoding. The model continued to decode the remaining tokens up to a maximum length of 256 tokens. (3). Removal of Both Decoding Strategies (*w/o* both): the CFIC model was tasked to decode outputs using a greedy search algorithm, devoid of both the sentence prefix and skip decoding strategies. (4). Absence of SFT (LongAlpaca-7B): LongAlpaca-7B is a context-extended version of LLAMA2-7B-chat. We utilized LongAlpaca-7B as the base model, representing the variant of CFIC without task-specific SFT.

The results of the ablation experiments are presented in Table 4. Our findings can be summarized as follows: (1). The removal of any of the CFIC model components resulted in a notable degradation in performance, underscoring the collective contribution of these elements to the model's effectiveness. (2). The most substantial decrease in performance was observed when SFT was omitted. This suggests that the vanilla LLM struggles to accurately locate precise grounding text from lengthy documents, despite its enhanced capability for processing extended contexts. (3). Removing either the sentence prefix decoding or the skip decoding strategies led to an obvious reduction in performance. This finding verifies our hypothesis that these decoding strategies not only curtail decoding computational demands but also improve the fedelity of the generated grounding text.

Choice of Decoding Length In our CFIC model, as defined in Eq. (7), the generation process is terminated upon locating the position of the [eos] token within a predetermined distance d. This distance is analogous to the maximum generation length typically set in standard text generation tasks, which governs the length of the decoded text. The selection of d involves a careful balance: too small a value may lead to excessively brief output grounding text, offering scant information for substantiating downstream tasks. Conversely, a larger d may result in longer output texts, potentially introducing additional textual noise and necessitating

Query: What hedge fund's collapse in 1998 highlighted the need for regulation of derivatives? **Answer**: Long Term Capital Management (LTCM)
CFIC-7B: In 1998, a trillion-dollar hedge fund called Long Term Capital Management (LTCM) was near collapse. Using mathematical models to calculate debt risk, LTCM used derivatives to leverage $5 billion into more than $1 trillion. The derivative transactions were not regulated, nor were investors able to evaluate LTCM's exposures.
LongAlpaca-7B: The catastrophic financial events of recent months have proved them (Born and Sheila Bair) right. In 2010, a documentary film Inside Job further alleged that derivatives regulation was ineffective from the Clinton administration on.
GPT-3.5-Turbo: The hedge fund whose collapse in 1998 highlighted the need for regulation of derivatives was Long Term Capital Management (LTCM).
GPT-4: In 1998, a trillion-dollar hedge fund called Long Term Capital Management (LTCM) was near collapse. Using mathematical models to calculate debt risk, LTCM used derivatives to leverage $5 billion into more than $1 trillion.

Table 5: Results of Case Study. The text colored with teal refers to the grounding evidence for the user query.

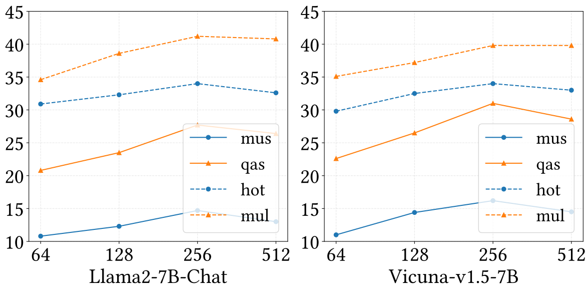

Figure 3: The choice of Maximum Decoding Length.

increased computational resources to process the extended sequences.

To investigate the optimal choice of decoding length d in CFIC, we conducted experiments with various settings of this parameter. The results of these experiments are depicted in Figure 3. Our findings substantiate the initial hypotheses: the performance across all tasks progressively improves and reaches its zenith at a d value of 256. Beyond this point, performance begins to wane, suggesting that a setting of $d = 256$ strikes an effective balance for these tasks. This observation aligns with the intuition that a span of 256 tokens typically suffices to encapsulate a semantically complete and coherent unit of information.

Case Study: CFIC v.s. GPTs OpenAI's model APIs, including GPT-3.5 and GPT-4, serve as robust baselines in the domain of LLM. However, they were excluded from the primary model comparisons in our experiments for two primary reasons: (1). these APIs lack of control over decoding process resulting in the inability to manipulate their decoding mechanisms to align with our methodological requirements. (2). The foundational models of GPT-3.5 and GPT-4 are characterized by their vast parameter sizes (*e.g.*, 175 billion parameters), endowing them with exceptional language modeling capabilities, especially in handling extended contexts. However, our focus with CFIC is on applying LLMs with comparatively smaller parameter sizes. This approach ensures more manageable computational resource requirements and enhances model scalability.

Despite these exclusions, we conducted a comparative case study, the results of which are presented in Table 5. This study reveals that our CFIC-7B model consistently provided complete and relevant grounding text evidence in response to queries. In contrast, the other models exhibited limitations: (1). LongAlpaca-7B failed to accurately locate appropriate grounding text, resulting in the generation of information irrelevant to the downstream tasks. (2). GPT-3.5 is able to directly respond to queries, it did not successfully identify precise grounding text from the original source material. (3). Although GPT-4 managed to retrieve grounding text pertinent to the query, the information provided was incomplete, lacking the necessary comprehensiveness to fully support the response logically.

5 Conclusion

This study introduces a Chunking-Free In-Context (CFIC) retrieval method for the RAG system, addressing the challenges of processing lengthy documents and refining evidence retrieval. Unlike traditional chunking-based methods that either compromise textual coherence or struggle with noise and inaccuracies, CFIC leverages auto-aggressive decoding to pinpoint precise evidence directly, eliminating the reliance on chunking. CFIC incorporates Constrained Sentence Prefix Decoding and Skip Decoding strategies to further enhances retrieval efficiency and accuracy. Through comprehensive evaluations on various open QA datasets, CFIC has demonstrated remarkable improvements in sourcing relevant and precise evidence to ground language models.

Limitations

This paper introduces a novel approach for Retrieval-Augmented Generation systems through the Chunking-Free In-Context (CFIC) retrieval method. Despite its advancements and effectiveness, there are certain limitations that warrant discussion.

One of the primary limitations stems from the training data used to develop our models. The dataset, self-constructed and annotated using ChatGPT, may harbor annotation biases. Such biases could affect the model's performance, particularly in its ability to generalize across different types of data or domains. While our approach excels in tasks requiring precise text evidence, it may offer limited assistance in scenarios demanding a high-level understanding of context, such as summarization tasks. This limitation is due to the model's focused capability on specific evidence retrieval rather than broader context comprehension.

Additionally, in this study, we have set the maximum length that CFIC can handle to 32k tokens. While this threshold accommodates a wide range of documents, it may not suffice for longer texts, such as novels, which exceed this limit. This constraint is primarily dictated by the available computational resources, highlighting a need for more efficient processing methods or greater computational power to extend CFIC's applicability to longer documents. With the increase in computational resources and advancements in model acceleration algorithms, we envision the future possibility of enabling CFIC to handle even longer contexts. This could potentially extend to encoding the entire corpus, facilitating corpus-level in-context retrieval for each query.

Ethical Impact

The development of CFIC builds upon existing Large Language Models (LLMs), which are trained on vast, diverse text corpora. This foundation introduces potential risks associated with biases inherent in the original training data. These biases can manifest in the model's outputs, influencing the quality and impartiality of the retrieved evidence.

Furthermore, the long documents processed by CFIC are sourced from the web, a domain rife with its biases. The web's text content reflects a wide array of perspectives, some of which may be skewed or unrepresentative of broader viewpoints. Given that CFIC is designed to process and retrieve information from these documents, there is a risk that the model might inadvertently perpetuate or amplify these biases without the capacity to discern or mitigate them.

References

Akari Asai, Matt Gardner, and Hannaneh Hajishirzi. 2022. Evidentiality-guided generation for knowledge-intensive NLP tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2226–2243, Seattle, United States. Association for Computational Linguistics.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

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

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023. Longlora: Efficient fine-tuning of long-context large language models. *arXiv:2309.12307*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers.

Zican Dong, Tianyi Tang, Lunyi Li, and Wayne Xin Zhao. 2023. A survey on long text modeling with transformers. *arXiv preprint arXiv:2302.14502*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Google. 2023. Gemini: A family of highly capable multimodal models. `https://goo.gle/GeminiPaper`.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. In *International Conference on Machine Learning*. JMLR.org.

Gautier Izacard and Edouard Grave. 2020. Distilling knowledge from reader to retriever for question answering. *arXiv preprint arXiv:2012.04584*.

Gautier Izacard and Edouard Grave. 2021a. Distilling knowledge from reader to retriever for question answering. In *International Conference on Learning Representations*.

Gautier Izacard and Edouard Grave. 2021b. Leveraging passage retrieval with generative models for open domain question answering.

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.

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

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*.

Ehsan Kamalloo, Nouha Dziri, Charles LA Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. *arXiv preprint arXiv:2305.06984*.

Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. 2023. Knowledge graph-augmented language models for knowledge-grounded dialogue generation. *arXiv preprint arXiv:2305.18846*.

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.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

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

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8460–8478, Dublin, Ireland. Association for Computational Linguistics.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2017. The narrativeqa reading comprehension challenge.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. 2023. How long can open-source llms truly promise on context length?

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2022a. Large language models with controllable working memory.

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

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts.

Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023b. Webglm: Towards an efficient web-enhanced question answering system with human preferences. *arXiv preprint arXiv:2306.07906*.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Reader-guided passage reranking for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 344–350, Online. Association for Computational Linguistics.

Sosuke Nishikawa, Ryokan Ri, Ikuya Yamada, Yoshimasa Tsuruoka, and Isao Echizen. 2022. Ease: Entity-aware contrastive learning of sentence embedding. *arXiv preprint arXiv:2205.04260*.

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

OpenAI. 2023. Gpt-4 technical report. `https://cdn.openai.com/papers/gpt-4.pdf`.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2022. Parallel Context Windows Improve In-Context Learning of Large Language Models. *arXiv*. Window.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021a. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021b. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

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

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multihop questions via single-hop question composition.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A globally normalized BERT model for open-domain question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5878–5882, Hong Kong, China. Association for Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2023a. Retrieval meets Long Context Large Language Models. *arXiv*. Experimental.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2023b. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025*.

Hui Yang, Sifu Yue, and Yunzhong He. 2023. Auto-gpt for online decision making: Benchmarks and additional opinions. *arXiv preprint arXiv:2306.02224*.

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

Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Mu-Nan Ning, and Li Yuan. 2023. Llm lies: Hallucinations are not bugs, but features as adversarial examples. *arXiv preprint arXiv:2310.01469*.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023a. Retrieve anything to augment large language models.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023b. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

A Implementation Detail

To train CFIC, we employed the "LLAMA2-7B-chat" as the foundation model for our CFIC. During the training, we set the batch size to 1 per GPU and the learning rate to 1e-5. We set the gradient accumulation step as 8 and utilized the AdamW optimizer with an epsilon value of 1e-8. The model's maximum length parameter was set to 32768. We train the model for 600 steps on 8 * Nvidia A800 80GB GPUs. For CFIC, We set the number of sampled sentence prefixes as $k = 3$ and the maximum decoding length as $d = 256$ (refers to Eq. (7)). Besides, we use a warm-up strategy to adjust the learning rate. To save GPU memory, we employed DeepSpeed's Stage 2 zero optimization to save GPU memory.