Coarse-to-Fine Highlighting: Reducing Knowledge Hallucination in Large Language Models

Qitan Lv [1] **Jie Wang** [1✉] **Hanzhu Chen** [1] **Bin Li** [1] **Yongdong Zhang** [1] **Feng Wu** [1]

Abstract

Generation of plausible but incorrect factual information, often termed hallucination, has attracted significant research interest. Retrieval-augmented language model (RALM)—which enhances models with up-to-date knowledge—emerges as a promising method to reduce hallucination. However, existing RALMs may instead exacerbate hallucination when retrieving lengthy contexts. To address this challenge, we propose COFT, a novel **CO**arse-to-**F**ine highligh**T**ing method to focus on different granularity-level key texts, thereby avoiding getting lost in lengthy contexts. Specifically, COFT consists of three components: *recaller*, *scorer*, and *selector*. First, *recaller* applies a knowledge graph to extract potential key entities in a given context. Second, *scorer* measures the importance of each entity by calculating its contextual weight. Finally, *selector* selects high contextual weight entities with a dynamic threshold algorithm and highlights the corresponding paragraphs, sentences, or words in a coarse-to-fine manner. Extensive experiments on the knowledge hallucination benchmark demonstrate the effectiveness of COFT, leading to a superior performance over 30% in the F1 score metric. Moreover, COFT also exhibits remarkable versatility across various long-form tasks, such as reading comprehension and question answering.

1. Introduction

Large language models (LLMs) have exhibited remarkable power and impressive generalization capabilities across a wide range of domains (Brown et al., 2020; El-Kassas et al., 2021). However, even the currently most capable LLM still exhibits knowledge hallucination issues, i.e., GPT4[1] (OpenAI, 2023) may also generate plausible yet incorrect factual information (Zhang et al., 2023b). Moreover, in long-form tasks consisting of multiple sentences or paragraphs, hallucination can be exacerbated (Wang & Sennrich, 2020).

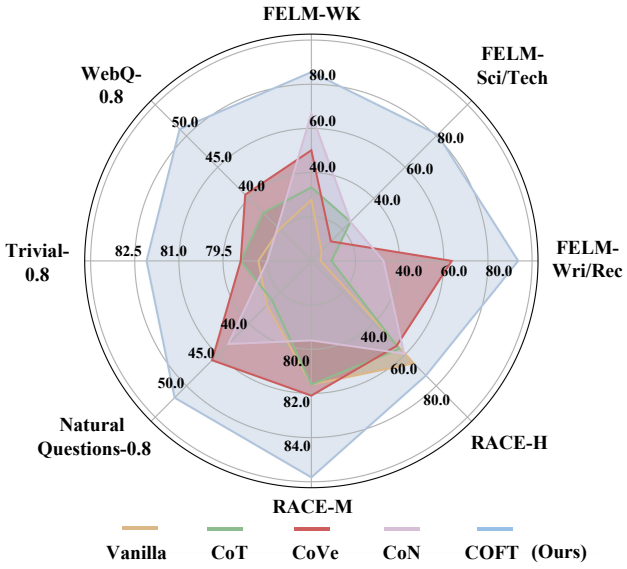

Figure 1. COFT achieves state-of-the-art performance on a broad range of long-form tasks compared with existing methods, using ChatGPT as the backbone.

To address this challenge, extensive research efforts have been devoted to reducing knowledge hallucination in LLMs (Kojima et al., 2022; Dhuliawala et al., 2023). Canonical methods, such as chain-of-thought (Wei et al., 2022), encourage LLMs to first generate internal thoughts or reasoning steps before responding (Adolphs et al., 2021; Wei et al., 2022). These methods enhance the logic of the reasoning process in LLMs, thereby implicitly reducing knowledge hallucination. Recently, retrieval-augmented language model (RALM) has emerged as a new trend to address hallucination, which enhances up-to-date knowledge in a plug-and-play manner (Vu et al., 2023; Yu et al., 2023).

[1]CAS Key Laboratory of Technology in GIPAS & MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China. Correspondence to: Jie Wang <jiewangx@ustc.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard

Extensive works demonstrate the effectiveness of RALMs (Gao et al., 2023b; Yu et al., 2023). RALMs retrieve the most relevant contexts from external knowledge sources for LLMs to make judgments. These contexts can contain thousands of tokens, such as relevant documents from search engines or database query results (Liu et al., 2023). The potential benefit of RALM is its ability to integrate relevant external knowledge, thereby enriching the LLMs' understanding of input text and generating answers based on this information. This is particularly beneficial when LLMs lack direct knowledge of a question (Yu et al., 2022).

Albeit with multiple benefits of RALMs, they confront significant challenges that severely hinder their performance and deployment. On the one hand, **the lack of complete contextual semantics.** When only retrieving several relevant sentences, the lack of complete contextual semantics may lead to misunderstandings. On the other hand, **the lost in the long context.** When retrieving the entire document for comprehensive information, irrelevant texts also distract their reasoning (Shi et al., 2023). Despite LLMs' ability to process long contexts, performances significantly decrease as the input grows longer, even for models explicitly designed for long contexts (Liu et al., 2023).

Therefore, in this paper, we seek to answer the question: *Can we propose a novel approach that preserves complete contextual semantics and exhibits robustness to long context?* With this consideration, we delve explicitly into the two significant challenges and propose a novel approach, namely **CO**arse-to-**F**ine highligh**T**ing (COFT), which preserves complete contextual semantics and avoids getting lost in long context. The key idea of COFT is to focus on the key texts when retrieving the entire document. COFT is a novel framework and effectively addresses the challenges within canonical RALM methods. Specifically, COFT consists of three components:

(i) *Recaller* integrates an external open-source knowledge graph (KG), wikidata, to extract potential key entities as candidates within the query and reference context. To enrich the candidates, *recaller* also retrieves their one-hop neighbors from the KG. The objective of *recaller* is to identify potential key entities.

(ii) *Scorer* applies a small language model, Llama 7B (Touvron et al., 2023), to calculate *contextual weight* of each candidate entities. Entities with higher contextual weights indicate a stronger correlation with the query, and vice versa. *Scorer* assigns different weights to measure the importance of each entity.

(iii) *Selector* proposes a dynamic threshold algorithm that considers both the length and informativeness of reference contexts to select high contextual weight entities. *Selector* then highlights each context based on these entities in a coarse-to-fine manner. *Selector* selects the final key entities and highlights the reference context.

COFT is a novel framework to reduce knowledge hallucination in LLMs. As shown in Figure 1, experiments on the knowledge hallucination benchmark demonstrate the effectiveness of COFT with an average improvement of 32.1% in the F1 score metric. COFT also serves as a plug-and-play framework for many long-form tasks, which achieves an average improvement of 4.6% in the precision metric for reading comprehension and a maximum improvement of 10.5% in the F1 score metric for question answering.

2. Related Work

2.1. Retrieval-Augmented Language Models

Retrieval-Augmented Language Models (RALMs) that enhance models with up-to-date knowledge by external knowledge sources, extend the knowledge boundaries of LLMs (Guu et al., 2020; Lewis et al., 2020; Izacard et al., 2022). These models first retrieve an external evidence corpus, such as Wikipedia, to pinpoint documents relevant to the query as reference texts (Karpukhin et al., 2020; Sachan et al., 2023). Then, a reader component analyzes these documents and provides a response (Izacard & Grave, 2020; Yu et al., 2021). This approach effectively retrieves reference texts related to the query, thereby enhancing the credibility of generated questions (Gao et al., 2023b; Jiang et al., 2023b). The evolution also leads to the emergence and popularity of retrieval-augmented products, such as ChatGPT plugins, Langchain, and New Bing.

2.2. Chain-of-X Approaches in LLMs

LLMs are capable of decomposing complex problems into a series of intermediate steps and generate internal thoughts or reasoning steps before responding, known as Chain-of-Thought (CoT) prompting (Wei et al., 2022). The CoT approach mirrors human problem-solving by breaking complex issues into smaller components, helping LLMs focus on each segment, reducing errors, and enhancing logic in reasoning (Wang et al., 2022). Following-up works effectively extend CoT to other chain-of-X methods, including chain-of-explanation (Huang et al., 2023) and chain-of-knowledge (Wang et al., 2023b). More recently, chain-of-verification (Dhuliawala et al., 2023) prompts LLMs to draft initial responses, plan verification questions, answer questions, and generate the final verified response, which reduces the likelihood of LLMs to misunderstand a specific concept. Chain-of-note (Yu et al., 2023) enables LLMs to annotate retrieved documents and incorporates them in formulating the response to enhance the robustness of LLMs.

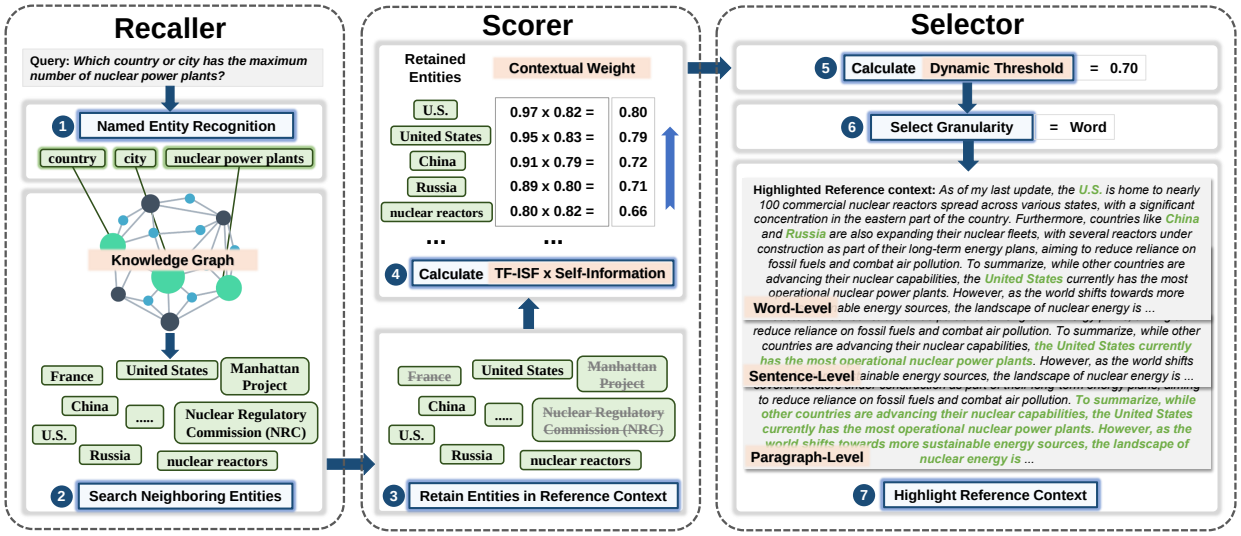

Figure 2. An overview of COFT. COFT integrates *recaller*, *scorer*, and *selector* into a unified framework to reduce knowledge hallucination. The workflow is as follows. (1) Perform Named Entity Recognition on the query to extract potential candidate entities. (2) Search the neighboring entities for each potential entity in the knowledge graph to enrich the candidates. (3) Retain candidates that are also present in the reference context as the final key entities. (4) Calculate the contextual weight for each key entity. (5) Calculate the threshold to filter a dynamic proportion of entities. (6) Choose the granularity for highlighting, such as word, sentence, or paragraph. (7) Highlight the reference context based on filtered entities and selected granularity.

2.3. Knowledge Hallucination

Hallucination is a general problem in LLMs, affecting various natural language processing tasks, such as reading comprehension (Maynez et al., 2020), open-domain question answering (Roller et al., 2020), and remains unresolved by simply enlarging training data or model size (Zhang et al., 2023a). We mainly discuss generation-time and retrieval-augmented methods to reduce knowledge hallucination, which are most relevant to our COFT.

For generation-time correction, efforts typically improve the token generation policy to enhance the reliability of generated contents (Mielke et al., 2022; Wu et al., 2023). Some methods enable models to generate contents along with corresponding confidence scores and correct low confidence output to reduce hallucinations (Gao et al., 2023b). Chain-of-**X** approaches also improve reasoning for logical tasks, which implicitly reduces hallucination. Several approaches get improved results with extended reasoning steps, such as deductive verification (Ling et al., 2023; Peng et al., 2023) and self-verification (Miao et al., 2023a; Galitsky, 2023).

For retrieval-augmented language models (RALMs), they mitigate hallucinations by applying external retrievers to provide query-relevant references and inject up-to-date knowledge, rather than relying solely on LLMs. RALMs can decrease hallucinations by using factual documents for grounding (Jiang et al., 2023c; Shuster et al., 2021). Several methods use automatic fact-checking and regeneration (Peng et al., 2023) or agreement voting and attribution analysis to conduct multi-round assessments (Chen et al., 2023a; Gao et al., 2023a). While RALMs help reduce hallucinations, they require high-quality texts. Irrelevant texts may exacerbate hallucination and performance declines as texts grow longer (Liu et al., 2023; Shi et al., 2023).

3. Preliminaries

3.1. Notations

We denote an input prompt for LLM as $\mathbf{x} = (\mathbf{x}^{\text{ins}}, \mathbf{x}^{\text{que}}, \mathbf{x}^{\text{refs}})$, where $\mathbf{x}^{\text{ins}}$ denotes the instructions for downstream tasks, $\mathbf{x}^{\text{que}}$ denotes the queries, and $\mathbf{x}^{\text{refs}}$ denotes reference contexts. Let $\mathcal{S} = [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \ldots]$ denote the sentence list of $\mathbf{x}^{\text{refs}}$, where $\mathbf{s}_i$ denote the i-th sentence and $\mathcal{E} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \ldots]$ denote the candidate key entity list, where $\mathbf{e}_k$ denote the k-th candidate. For a given entity $\mathbf{e}_k$, we denote $f_{\mathbf{e}_k,\mathbf{s}_i}$ and $f_{\mathbf{e}_k,\mathcal{S}}$ as the number of times $\mathbf{e}_k$ appears in $\mathbf{s}_i$ and $\mathcal{S}$. Let $|\mathbf{s}_i|$ and $|\mathcal{S}|$ denote the number of words within sentence $\mathbf{s}_i$ and the reference context $\mathcal{S}$. Let $\mathbf{t}_i$ denote the i-th token in $\mathbf{x}^{\text{refs}}$, $P(\mathbf{t}_i)$ denote its output probability by a small language model $\mathcal{M}_s$, and $I(\mathbf{t}_i)$ denote the self-information of token $\mathbf{t}_i$. Let $\oplus$ denote the concatenation of two texts.

3.2. Self-Information

Self-information is a fundamental concept in information theory, which quantifies the amount of information contained in a random event (Shannon, 1948). In natural language processing, an event can be regarded as a generation step (i.e., a token), and the distribution corresponds to its

output distribution. We can obtain self-information of a token $\mathbf{t}_i$ by the follow equation:

$$I(\mathbf{t}_i) = -\log_2 P(\mathbf{t}_i \mid \mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{i-1})$$

where $I(\mathbf{t}_i)$ denotes the self-information of token $\mathbf{t}_i$ and $P(\mathbf{t}_i)$ denotes its output probability.

In information theory, self-information represents the amount of information contained in a random event. The higher the probability of a random event occurring, the lower its self-information. Rare events convey more information, thus having higher self-information, while common events convey less information, resulting in lower self-information.

In natural language processing, self-information can be utilized to evaluate the informativeness in **lexical units such as words, sentences, or paragraphs.** Lexical units with higher self-information carry important information, acting as key units that determine the semantics of the context. Conversely, lexical units with lower self-information contain less information and exert a smaller impact on the semantic interpretation of the context. Some works apply self-information in creative language (Bunescu & Uduehi, 2022) and information compression (Li et al., 2023; Jiang et al., 2023a). The self-information between two independent events exhibits an additive property as follows:

$$\begin{aligned} I(\mathbf{t}_0, \mathbf{t}_1) &= -\log_2 P(\mathbf{t}_0, \mathbf{t}_1) \\ &= -\log_2 P(\mathbf{t}_0) P(\mathbf{t}_1 \mid \mathbf{t}_0) \\ &= -\log_2 P(\mathbf{t}_0) - \log_2 P(\mathbf{t}_1 \mid \mathbf{t}_0) \\ &= I(\mathbf{t}_0) + I(\mathbf{t}_1) \end{aligned} \tag{1}$$

This means we can measure the self-information of a lexical unit by summing the self-information of its tokens.

4. Method

We propose a **CO**arse-to-**F**ine highligh**T**ing method (COFT) that promotes LLMs to focus on key lexical units, which preserves complete contextual semantics and avoids getting lost in long contexts. COFT can highlight different granularity-level lexical units in a coarse-to-fine manner, such as paragraphs, sentences, and words. COFT organically integrates three modules in a unified framework. An overview of COFT is shown in Figure 2.

4.1. Recaller

Recaller first generates candidate key entities extracted from the query and then retains the candidates occurred in the reference contexts. Specifically, for a given query and reference context, the workflow of *recaller* is as follows:

(i) *Recaller* first conducts named entity recognition on the query to extract named entities that represent keywords within the query. These entities include some specific terms and important nouns such as people, places, organizations, etc.

(ii) After obtaining named entities, *recaller* leverages them to search one-hop neighbor entities in wikidata to enrich candidate entities. The named entities and corresponding one-hop neighbors are combined to form candidate entities for the query.

(iii) *Recaller* finally retains candidate entities that are also present in the reference context, forming the final candidate key entities list.

As shown in the left part of Figure 2, given a query such as "Which country or city has the maximum number of nuclear power plants?", *recaller* **first** performs named entity recognition to identify entities like "country", "city", and "nuclear power plants". **Then**, *recaller* extracts one-hop neighboring entities from wikidata for each named entity, such as "United States" and "France". **Finally**, based on these named entities and neighboring entities, *recaller* retains entities that are present in the reference context as the final candidate key entities list. For example, "France" will not be retained because it is not in the reference context.

4.2. Scorer

After obtaining candidate key entities, *scorer* proceeds to assess their importance. With this desiderata, *scorer* proposes an entity-level iterative algorithm based on a small language model, Llama 7B (Touvron et al., 2023) to calculate the contextual weight of each entity in the context. Algorithm 1 outlines the overall procedure.

Algorithm 1 Pseudo code for entity-level iterative algorithm

Input: A query $\mathbf{x}^{\text{que}}$, a reference context $\mathbf{x}^{\text{refs}}$, a key candidate entity list $\mathcal{E}$, and a small language model $\mathcal{M}_s$.
Segment the reference context $\mathbf{x}^{\text{refs}}$ into sentences list $\mathcal{S} = [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \ldots]$.
Initialize the TF-ISF dictionary $\mathcal{D}_{TF\text{-}ISF}$, the self-information dictionary $\mathcal{D}_{SI}$, and the contextual weight dictionary $\mathcal{D}_{CW}$.
for $\mathbf{e}_k$ **in** $\mathcal{E}$ **do**
    Retain $\mathbf{e}_k$ occurred in each reference sentence $\mathbf{s}_i \in \mathcal{S}$.
    Calculate the TF-ISF score of each entity via Equation 2 and append entities and corresponding TF-ISF scores into $\mathcal{D}_{TF\text{-}ISF}$.
    Calculate the self-information score of each entity by the language model $\mathcal{M}_s$ via Equation 3 and append entities and self-information scores into $\mathcal{D}_{SI}$.
    Calculate the contextual weights of each entity using $\mathcal{D}_{TF\text{-}ISF}$ and $\mathcal{D}_{SI}$ via Equation 4 and append all entities and their contextual weights into $\mathcal{D}_{CW}$.
end for
Output: Contextual weights dictionary $\mathcal{D}_{CW}$.

Specifically, we first segment reference contexts $\mathbf{x}^{\text{refs}}$ into sentence list $\mathcal{S} = [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \ldots]$. Drawing upon the TF-IDF (**T**erm **F**requency–**I**nverse **D**ocument **F**requency) algorithm

(Sparck Jones, 1972), a well-suited text relevance assessment and text mining approach that enables the exclusion of the majority of common entities while preserving important entities. We introduce the TF-ISF algorithm, which involves considering the TF-IDF algorithm at the Sentence level. For a given entity $\mathbf{e}_k$ in sentence $\mathbf{s}_i$, the corresponding TF-ISF calculation function is as follows:

$$TF\text{-}ISF(\mathbf{e}_k) = \frac{f_{\mathbf{e}_k,\mathbf{s}_i}}{|\mathbf{s}_i|} \times \log_2\left(\frac{|\mathcal{S}|}{f_{\mathbf{e}_k,\mathcal{S}}+1}\right) \quad (2)$$

where $f_{\mathbf{e}_k,\mathbf{s}_i}$ and $f_{\mathbf{e}_k,\mathcal{S}}$ denote the number of times $\mathbf{e}_k$ appears in $\mathbf{s}_i$ and $\mathcal{S}$. $|\mathbf{s}_i|$ and $|\mathcal{S}|$ denote the number of words within sentence $\mathbf{s}_i$ and reference contexts $\mathcal{S}$.

TF-ISF evaluates the importance of entities in reference context based on word frequency and effectively distinguishes common but unimportant entities. Higher TF-ISF suggests that the entity plays a more important role in understanding the sentence semantics, and vice versa.

We further concatenate the query and reference context to measure the importance of each token in the reference context based on self-information. Given the input $\mathbf{x}^{\text{que}} \oplus \mathbf{x}^{\text{refs}}$, the self-information calculation function is as follows:

$$I(\mathbf{t}_i) = -\log_2 P\left(\mathbf{t}_i \mid \mathbf{x}^{\text{que}}, \mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{i-1}\right) \quad (3)$$

where $\mathbf{t}_i$ denotes the i-th token within the reference context $\mathbf{x}^{\text{refs}}$, $P(\mathbf{t}_i|\mathbf{x}^{\text{que}}, \mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{i-1})$ denotes its output probability by the small language model $\mathcal{M}_s$, and $I(\mathbf{t}_i)$ denotes the self-information of token $\mathbf{t}_i$. We can further leverage the additivity property of self-information in Equation 1 to merge tokens into entity $\mathbf{e}$, thereby obtaining the self-information of each individual key candidate entity $I(\mathbf{e})$.

To comprehensively consider both the TF-ISF and self-information, we propose contextual weights to indicate the importance of each key candidate entity in the reference context. A higher contextual weight suggests greater importance of the entity to answer the query. The contextual weight calculation function is as follows:

$$w(\mathbf{e}_k) = TF\text{-}ISF(\mathbf{e}_k) \times I(\mathbf{e}_k) \quad (4)$$

where $TF\text{-}ISF(\mathbf{e}_k)$ and $I(\mathbf{e}_k)$ denote the TF-ISF and self-information of a key candidate entity $\mathbf{e}_k$, respectively. Other combination methods for TF-ISF and self-information scores are also feasible, and we leave it as a future work.

4.3. Selector

After obtaining candidate key entities and their contextual weights, *selector* highlights the final lexical units for the query. Specifically, *selector* first sorts entities based on contextual weights, and proposes a dynamic threshold algorithm to filter a dynamic proportion of key entities. The dynamic thresholds can be defined as $\tau = 0.5 \times (\tau_{len} + \tau_{info})$, where τ_{len} and τ_{info} denote the min-max normalized value of the length and informativeness for each reference context. τ varies with the length and informativeness of the reference context, as longer and more informative reference context requires more highlights. Then, *selector* highlights the reference context according to the granularity of selected lexical units. This highlighting process is as follows:

(i) Split the reference context according to the granularity of selected lexical units.
(ii) Calculate the contextual weight of the split lexical units by summing the contextual weight of candidate key entities occurred in the split.
(iii) Sort these lexical units in descending order by their contextual weight, and select the lexical units with contextual weights in the top $\tau \times 100\%$ for highlighting.

After selecting the highlighted lexical units, *selector* inserts special symbols around these lexical units. Considering the rich diversity of formatting found in publicly accessible web data, which forms a part of the pre-training corpus for LLMs, we adopt markdown syntax, particularly the **bold syntax (**)** as an example, to highlight important lexical units. This approach aligns with the natural occurrence of formatted text in online sources, thereby enabling the LLMs to more accurately interpret and process textual emphasis as it appears in real-world scenarios. Take word-level granularity highlighting as an example. If the selected highlighted entities are "nuclear power plants" and "United States", then the sentence "The nuclear power plants in the United States play a crucial role in providing ..." will be highlighted as "The **nuclear power plants** in the **United States** play a crucial role in providing ..." as input for LLM inference. Other highlighting methods, such as HTML bold symbols or different markdown syntax are also viable options and we leave the exploration as a future work.

5. Experiments

We design experiments to evaluate the effectiveness of COFT for reducing knowledge hallucination and demonstrate the versatility of COFT on a variety of tasks. With this desiderata, we divide the experiments into four parts:

(i) To evaluate the effectiveness of COFT, we compare COFT with existing state-of-the-art methods for reducing knowledge hallucination.
(ii) To demonstrate the versatility of COFT, we conduct experiments on reading comprehension and question-answering benchmarks.
(iii) To investigate the contribution of each component within COFT, we conduct the ablation study.
(iv) To provide more insight into COFT, we conduct the visualization study.

Table 1. Results of knowledge hallucination benchmark on WK (world knowledge), Sci/Tech (science and technology), and Wri/Rec (writing and recommendation) domains. We denote COFT at the paragraph, sentence, and word levels as COFT_p, COFT_s, and COFT_w. The results of vanilla, CoT, and RALM methods are taken from FELM (Chen et al., 2023c). We **bold** the best results for each LLM backbone.

		WK			Sci/Tech			Wri/Rec		
Backbone	Methods	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
Vicuna-33B	Vanilla	34.5	27.8	45.5	25.8	17.4	50.2	27.1	16.4	78.7
	CoT	32.3	27.4	39.5	20.4	12.7	52.9	26.5	17.0	60.3
	RALM	48.7	45.7	52.1	34.2	24.7	55.8	27.1	16.2	82.8
	CoVe	47.3	47.6	47.1	47.2	39.8	58.2	64.0	66.7	61.5
	CoN	55.9	55.7	56.1	59.3	58.1	60.6	62.4	55.3	71.5
	COFT_p	**69.3**	**71.9**	66.9	67.9	62.9	73.8	70.4	66.8	74.4
	COFT_s	62.0	63.1	60.9	68.7	**67.1**	70.4	66.2	64.7	67.7
	COFT_w	64.4	61.7	**67.4**	**70.9**	65.7	**77.2**	**77.3**	**67.9**	**89.8**

		WK			Sci/Tech			Wri/Rec		
Backbone	Methods	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
ChatGPT	Vanilla	9.1	27.6	5.4	4.1	6.5	2.9	0.7	4.2	0.4
	CoT	2.6	33.3	1.4	4.2	25.1	2.3	2.7	9.1	1.6
	RALM	25.2	34.9	19.7	17.4	16.7	18.2	20.1	54.1	12.4
	CoVe	20.0	50.1	12.5	18.2	12.5	33.3	23.1	63.6	14.1
	CoN	18.2	66.7	10.6	20.0	25.0	16.7	31.4	32.7	30.3
	COFT_p	78.6	83.8	74.0	83.9	**81.2**	86.8	77.5	85.9	70.5
	COFT_s	76.8	75.7	77.9	74.6	79.1	70.5	76.8	84.4	70.5
	COFT_w	**81.6**	**85.5**	**77.9**	**84.4**	80.9	**88.4**	**81.1**	**93.7**	**71.5**

		WK			Sci/Tech			Wri/Rec		
Backbone	Methods	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
GPT4	Vanilla	40.2	76.9	27.2	19.7	60.0	11.8	22.3	89.5	12.7
	CoT	50.2	79.4	36.7	25.2	64.0	15.7	26.2	89.1	15.4
	RALM	53.6	80.8	40.1	34.7	59.5	24.5	52.2	63.8	44.2
	CoVe	49.7	55.4	45.1	66.7	83.3	55.6	48.2	56.9	41.8
	CoN	52.8	45.2	63.6	66.7	75.0	60.0	68.8	78.6	61.1
	COFT_p	83.1	79.7	**86.8**	**89.9**	84.4	**96.1**	**91.8**	85.5	**99.1**
	COFT_s	80.0	92.3	70.6	76.6	84.9	69.8	85.5	89.2	82.1
	COFT_w	**87.3**	**94.8**	80.9	77.9	**86.0**	71.3	84.7	**92.9**	77.9

5.1. Experiment Setups

Experiment Setups. We apply LLMs including Vicuna[2] (`vicuna-33B-v1.3`) (Zheng et al., 2023), ChatGPT[3] (`gpt-3.5-turbo`) and GPT4 (`gpt-4`) (OpenAI, 2023) as backbone models. To guarantee stable and reproducible results, we utilize greedy decoding and set the temperature parameter as 0 in all experiments. For **knowledge hallucination**, we use FELM (Chen et al., 2023c) as the benchmark with precision, recall, and F1 score as evaluation metrics (Chen et al., 2023c). For **reading comprehension**, we use RACE-H (high school level reading comprehension) and RACE-M (middle school level reading comprehension) (Lai et al., 2017) as benchmarks with precision as the metric (Bi et al., 2024; Rae et al., 2021). For **question answering**, we use Natural Question (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and WebQ (Berant et al., 2013) as benchmarks with EM and F1 score as metrics (Chen et al., 2017; Zhu et al., 2021). Details of experiment setups and datasets are in Appendix B. For knowledge hallucination, we use word, sentence, and paragraph granularity levels of COFT (denoted as COFT_w, COFT_s, and COFT_p). For reading comprehension and question answering, we focus specifically on the word-level COFT_w (denoted as COFT).

Baseline Methods. We examine five variants for each of LLMs: **(i) vanilla**: standalone LLMs without any additional preprocessing modules or external retrievers. Vanilla LLMs represent the original capabilities of LLMs. **(ii) Chain-of-thought (CoT)** (Wei et al., 2022): LLMs are asked to

[2]https://huggingface.co/lmsys/vicuna-33b-v1.3
[3]https://platform.openai.com/

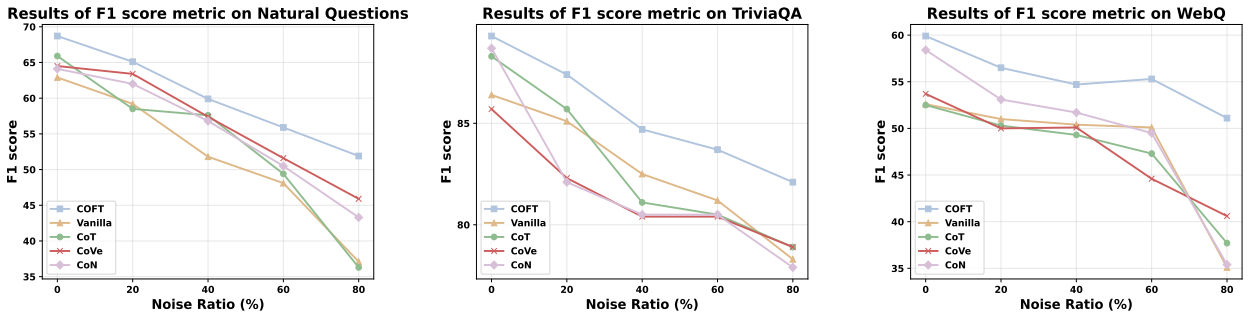

Figure 3. Evaluation on F1 score metric of noise robustness in question answering task, utilizing ChatGPT as the backbone model. COFT demonstrates superior performance on all three open-domain QA benchmarks, especially at higher noise ratios.

first generate internal thoughts or reasoning steps before responding. **(iii) RALM**: following (Chen et al., 2023c), we use LLMs with BM25 algorithm (Robertson et al., 2009) to retrieve the most relevant texts as input to vanilla LLMs. **(iv) Chain-of-verification (CoVe)** (Dhuliawala et al., 2023): CoVe prompts LLMs to draft the initial response, plan verification questions, answer the question, and generate the final verified response. **(v) Chain-of-note (CoN)** (Yu et al., 2023): enables LLMs to sequentially annotate the retrieved documents and incorporates them to formulate the response.

5.2. Knowledge Hallucination Results

In this section, we conduct experiments on the knowledge hallucination benchmark. As shown in Table 1, we observe that COFT significantly and consistently outperforms existing methods on the hallucination benchmark. Specifically, for all three backbone models, COFT achieves average improvements of 34.5%; 33.1%; 28.7% in the F1 score metric, 16.3%; 22.6%; 11.6% in precision metric, and 30.9%; 35.9%; 28.7% in recall metric for WK (world knowledge, a wide domain including movies, countries, places, and so on), Sci/Tech (Science and Technology spanning various academic disciplines such as physics, chemistry, and biology), and Wri/Rec (Writing and Recommendation, including details of some books and movies) domains.

While methods such as CoT and CoN do not consistently enhance the performance of Vicuna-33B and ChatGPT across various datasets, COFT consistently demonstrates a superior performance over vanilla models. Notably, in the science and technology domain, COFT achieves a maximum performance enhancement of over 60% in the F1 score metric, which effectively underscores the importance of capturing key information in the entire context. The universality of three backbone models also suggests that COFT possesses the potential across various LLMs.

5.3. Reading Comprehension Results

Reading comprehension task necessitates that LLMs answer certain questions based on the entire content, requiring the model to retain a comprehensive understanding of the complete contextual semantics. Through the reading comprehension task, we investigate COFT's ability for full-context awareness in long contexts. We conduct experiments on RACE-H and RACE-M (Lai et al., 2017) and only use the provided reading passages and do not use other information from retrieval systems. Consequently, we do not include RALM as a baseline. We present the results of COFT using ChatGPT as the backbone in Table 2. More Results using Vicuna-33B and GPT4 as backbones are in Appendix C.

Table 2. Results of the reading comprehension task in the precision metric, utilising ChatGPT as the backbone model.

Backbone	Methods	RACE-H	RACE-M
ChatGPT	Vanilla	65.6	81.6
	CoT	56.3	81.6
	CoVe	54.5	82.1
	CoN	59.4	79.6
	COFT	**73.4**	**85.8**

As shown in Table 2, COFT exhibits great performance on both the RACE-H and RACE-M, which outperforms the suboptimal results by 7.8% and 3.7% in the precision metric. We observe that COFT achieves more performance enhancement on the more challenging and complex dataset, RACE-H. This suggests that COFT possesses potential for application in more complex real-world scenarios. Moreover, COFT consistently yields improved results over vanilla models, which demonstrates the effectiveness of focusing on key lexical units and maintaining full context semantics.

5.4. Question Answering Results

Question answering task requires the LLM to effectively focus on keywords and phrases within a question. Following CoN (Yu et al., 2023), we conduct experiments on question-answering tasks to evaluate the robustness of COFT under scenarios where reference texts contain both relevant and noisy documents. These noise documents are retrieved based on their semantic similarity to the input questions, which often contain similar but misleading information. We

Table 3. The results of ablation study on the knowledge hallucination benchmark, FELM, using ChatGPT as the backbone model.

Backbone	Methods	WK			Sci/Tech			Wri/Rec		
		F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
ChatGPT	COFT$_{w/o\ recaller}$	74.6	81.0	69.1	73.9	80.7	68.2	63.6	86.1	60.1
	COFT$_{w/o\ TF\text{-}ISF}$	78.3	82.4	74.7	78.5	81.8	75.5	67.1	84.5	55.7
	COFT$_{w/o\ SI}$	76.9	80.9	73.3	76.1	80.5	72.1	64.5	85.8	51.7
	COFT$_{w/o\ scorer}$	76.2	80.3	72.6	74.6	85.9	65.8	60.1	87.2	45.8
	COFT$_{w/o\ selector}$	77.3	79.7	75.1	75.7	82.4	70.1	70.7	86.1	60.1
	COFT	**81.6**	**85.5**	**77.9**	**85.5**	**86.5**	**84.5**	**75.2**	**88.3**	**65.4**

employ the *noise ratio* to represent the extent of noisy interference under varying degrees of noise. For instance, if the top-k documents are retrieved for LLMs, then $k \times r$ represents the number of noisy documents, while $k \times (1-r)$ indicates the number of relevant documents. For example, with a 20% noise ratio and a requirement for the top-5 documents, 4 would be relevant documents, and 1 would be a noisy document. We concatenate relevant and noisy documents randomly, to mitigate position bias (Zheng et al., 2023). This requires LLMs to identify the most relevant information under lengthy and noisy conditions.

As illustrated in Figure 3, we observe that compared to other methods, COFT demonstrates relative robustness to reference texts containing noisy text, maintaining focus on highlighted key text within reference contexts. These results demonstrate that COFT is robust against noisy texts, exhibiting a higher tolerance for noisy information, which more closely aligns with user inputs in real-world scenarios.

5.5. Ablation Study

To further investigate the contribution of each component within COFT, we conduct a series of ablation experiments on the entire framework. We select a word-level version, COFT$_w$ to conduct the ablation study. Other granularity versions of COFT including sentences or paragraphs follow a similar way. For simplicity, we denote COFT$_w$ as COFT in this section. Specifically, we denote COFT without *recaller* extracting candidate key entities as COFT$_{w/o\ recaller}$, COFT without the TF-ISF score as COFT$_{w/o\ TF\text{-}ISF}$, COFT without the self-information score as COFT$_{w/o\ SI}$, COFT without *scorer* calculating the contextual weight as COFT$_{w/o\ scorer}$, and COFT without dynamic threshold selecting key candidate entities as COFT$_{w/o\ selector}$, respectively. We set the threshold τ to 0.5 for COFT$_{w/o\ selector}$ as an example. More detailed results are in Appendix E.1.

We present ablation results of COFT using ChatGPT as the backbone model in Table 3. More Results using backbone models including Vicuna-33B and GPT4 are in Appendix E.2. As shown in Table 3, the absence of any component within COFT results in a performance degradation of the entire framework. Notably, *recaller* and *scorer* have more significant impacts on the performance of COFT, which demonstrates the importance of extracting candidate key lexical units from the reference text and ranking them based on contextual weight to reduce knowledge hallucination.

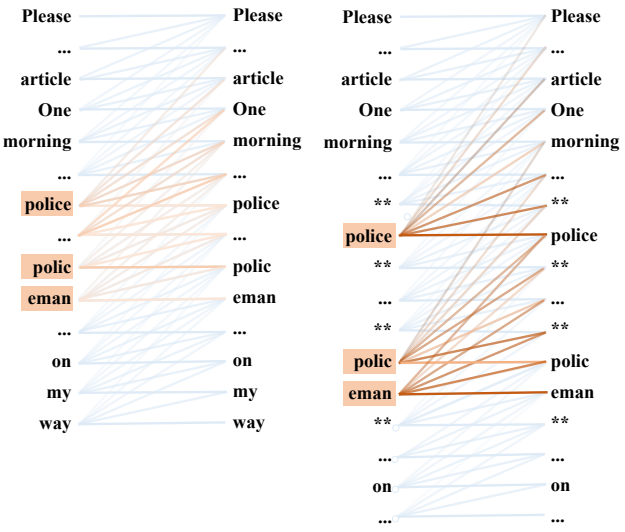

Figure 4. Visualization of the information flow in Vicuna-33B before (left) and after (right) highlighting key lexical units (between two ** symbols). The line color depth reflects the significance of the information flow from the right word to the left.

5.6. Visualization Study

To provide more insight into COFT, we conduct a visualization study. As mentioned above, COFT promotes LLMs to focus on key texts in the entire context. We employ attention scores to trace the information flow in the reference context based on Vicuna-33B, both before and after highlighting (Wang et al., 2023c). As shown in Figure 4, the highlighted key lexical units possess higher attention scores and exhibit stronger interactions with other words. This suggests that LLMs better focus on these highlighted key lexical units during inference.

6. Conclusions

In this paper, we propose a novel **CO**arse-to-**F**ine high-ligh**T**ing method to effectively reduce knowledge hallucination. Specifically, we propose *recaller*, *scorer*, and *selector*

to form a general framework for LLMs to focus on key texts and avoid getting lost in long contexts. Extensive experiments on the knowledge hallucination task demonstrate the effectiveness of COFT with an average improvement of 32.1% in the F1 score metric. This superior performance over existing state-of-the-art methods demonstrates the effectiveness of COFT in reducing knowledge hallucination in LLMs. COFT also serves as a plug-and-play framework for many long-form tasks that achieves an average improvement of 4.6% in the precision metric for reading comprehension tasks and a maximum improvement of 10.5% in the F1 score metric for question-answering tasks.

Acknowledgements

This work was supported in part by National Key R&D Program of China under contract 2022ZD0119801, National Nature Science Foundations of China grants U23A20388, 62021001, U19B2026, and U19B2044. We would like to thank all the anonymous reviewers for their insightful comments.

Impact Statement

This paper follows existing research in the field of hallucination based on existing LLMs and open-source datasets. Moreover, our work does not involve human or animal experiments and will not provide new LLMs or datasets, so the ethical impacts and expected societal implications are those that are well established when advancing the field of hallucination.

This paper presents work whose goal is to advance the field of hallucination. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

As for the limitations of COFT, while COFT can significantly reduce knowledge hallucination in LLMs by focusing on key texts, it cannot update the knowledge within LLMs. Exploring low-cost methods to update knowledge within LLMs for to furture reduce knowledge hallucination in LLMs will be the focus of our future work. We will also focus on employing this approach as a means to explicitly interpret the LLMs.

References

Adolphs, L., Shuster, K., Urbanek, J., Szlam, A., and Weston, J. Reason first, then respond: Modular generation for knowledge-infused dialogue. *arXiv preprint arXiv:2111.05204*, 2021.

Anagnostidis, S., Pavllo, D., Biggio, L., Noci, L., Lucchi, A., and Hoffmann, T. Dynamic context pruning for efficient and interpretable autoregressive transformers. *arXiv preprint arXiv:2305.15805*, 2023.

Berant, J., Chou, A., Frostig, R., and Liang, P. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1533–1544, 2013.

Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., Ding, H., Dong, K., Du, Q., Fu, Z., et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Bunescu, R. and Uduehi, O. O. Distribution-based measures of surprise for creative language: Experiments with humor and metaphor. In *Proceedings of the 3rd Workshop on Figurative Language Processing (FLP)*, pp. 68–78, 2022.

Chen, A., Pasupat, P., Singh, S., Lee, H., and Guu, K. Purr: Efficiently editing language model hallucinations by denoising language model corruptions. *arXiv preprint arXiv:2305.14908*, 2023a.

Chen, C., Borgeaud, S., Irving, G., Lespiau, J.-B., Sifre, L., and Jumper, J. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023b.

Chen, D., Fisch, A., Weston, J., and Bordes, A. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

Chen, S., Zhao, Y., Zhang, J., Chern, I., Gao, S., Liu, P., He, J., et al. Felm: Benchmarking factuality evaluation of large language models. *arXiv preprint arXiv:2310.00741*, 2023c.

Chern, I.-C., Wang, Z., Das, S., Sharma, B., Liu, P., Neubig, G., et al. Improving factuality of abstractive summarization via contrastive reward learning. *arXiv preprint arXiv:2307.04507*, 2023.

Chevalier, A., Wettig, A., Ajith, A., and Chen, D. Adapting language models to compress contexts. *arXiv preprint arXiv:2305.14788*, 2023.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B.,

Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. Palm: Scaling language modeling with pathways, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A., and Weston, J. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*, 2023.

El-Kassas, W. S., Salama, C. R., Rafea, A. A., and Mohamed, H. K. Automatic text summarization: A comprehensive survey. *Expert systems with applications*, 165: 113679, 2021.

Fei, W., Niu, X., Zhou, P., Hou, L., Bai, B., Deng, L., and Han, W. Extending context window of large language models via semantic compression. *arXiv preprint arXiv:2312.09571*, 2023.

Frantar, E. and Alistarh, D. Qmoe: Practical sub-1-bit compression of trillion-parameter models. *arXiv preprint arXiv:2310.16795*, 2023.

Galitsky, B. A. Truth-o-meter: Collaborating with llm in fighting its hallucinations. 2023.

Gao, L., Dai, Z., Pasupat, P., Chen, A., Chaganty, A. T., Fan, Y., Zhao, V., Lao, N., Lee, H., Juan, D.-C., et al. Rarr: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16477–16508, 2023a.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., and Wang, H. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023b.

Ge, T., Hu, J., Wang, X., Chen, S.-Q., and Wei, F. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*, 2023.

Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Hsieh, C.-Y., Li, C.-L., Yeh, C.-K., Nakhost, H., Fujii, Y., Ratner, A., Krishna, R., Lee, C.-Y., and Pfister, T. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, 2023.

Huang, F., Kwak, H., and An, J. Chain of explanation: New prompting method to generate quality natural language explanation for implicit hate speech. In *Companion Proceedings of the ACM Web Conference 2023*, pp. 90–93, 2023.

Izacard, G. and Grave, E. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*, 2020.

Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., and Grave, E. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 2022.

Jiang, H., Wu, Q., Lin, C.-Y., Yang, Y., and Qiu, L. Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*, 2023a.

Jiang, H., Wu, Q., Luo, X., Li, D., Lin, C.-Y., Yang, Y., and Qiu, L. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *arXiv preprint arXiv:2310.06839*, 2023b.

Jiang, Z., Xu, F. F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J., Yang, Y., Callan, J., and Neubig, G. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*, 2023c.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

Leviathan, Y., Kalman, M., and Matias, Y. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2023.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

Li, Y., Dong, B., Lin, C., and Guerin, F. Compressing context to enhance inference efficiency of large language models. *arXiv preprint arXiv:2310.06201*, 2023.

Ling, Z., Fang, Y., Li, X., Huang, Z., Lee, M., Memisevic, R., and Su, H. Deductive verification of chain-of-thought reasoning. *arXiv preprint arXiv:2306.03872*, 2023.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*, 2023.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach, 2019.

Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*, 2020.

Miao, N., Teh, Y. W., and Rainforth, T. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. *arXiv preprint arXiv:2308.00436*, 2023a.

Miao, X., Oliaro, G., Zhang, Z., Cheng, X., Jin, H., Chen, T., and Jia, Z. Towards efficient generative large language model serving: A survey from algorithms to systems. *arXiv preprint arXiv:2312.15234*, 2023b.

Mielke, S. J., Szlam, A., Dinan, E., and Boureau, Y.-L. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872, 2022.

OpenAI. Gpt-4 technical report, 2023.

Peng, B., Galley, M., He, P., Cheng, H., Xie, Y., Hu, Y., Huang, Q., Liden, L., Yu, Z., Chen, W., et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*, 2023.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

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

Roit, P., Ferret, J., Shani, L., Aharoni, R., Cideron, G., Dadashi, R., Geist, M., Girgin, S., Hussenot, L., Keller, O., et al. Factually consistent summarization via reinforcement learning with textual entailment feedback. *arXiv preprint arXiv:2306.00186*, 2023.

Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Shuster, K., Smith, E. M., et al. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*, 2020.

Sachan, D. S., Lewis, M., Yogatama, D., Zettlemoyer, L., Pineau, J., and Zaheer, M. Questions are all you need to train a dense passage retriever. *Transactions of the Association for Computational Linguistics*, 11:600–616, 2023.

Shannon, C. E. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Shi, F., Chen, X., Misra, K., Scales, N., Dohan, D., Chi, E. H., Schärli, N., and Zhou, D. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pp. 31210–31227. PMLR, 2023.

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. Megatron-lm: Training multi-billion parameter language models using model parallelism. *Cornell University - arXiv,Cornell University - arXiv*, Sep 2019.

Shuster, K., Poff, S., Chen, M., Kiela, D., and Weston, J. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*, 2021.

Sparck Jones, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

Sun, W., Shi, Z., Gao, S., Ren, P., de Rijke, M., and Ren, Z. Contrastive learning reduces hallucination in conversations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 13618–13626, 2023.

Team, G. Gemini: A family of highly capable multimodal models, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023.

Vu, T., Iyyer, M., Wang, X., Constant, N., Wei, J., Wei, J., Tar, C., Sung, Y.-H., Zhou, D., Le, Q., et al. Freshllms: Refreshing large language models with search engine augmentation. *arXiv preprint arXiv:2310.03214*, 2023.

Wang, C. and Sennrich, R. On exposure bias, hallucination and domain shift in neural machine translation. *arXiv preprint arXiv:2005.03642*, 2020.

Wang, H., Ma, S., Dong, L., Huang, S., Wang, H., Ma, L., Yang, F., Wang, R., Wu, Y., and Wei, F. Bitnet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453*, 2023a.

Wang, J., Sun, Q., Chen, N., Li, X., and Gao, M. Boosting language models reasoning with chain-of-knowledge prompting. *arXiv preprint arXiv:2306.06427*, 2023b.

Wang, L., Li, L., Dai, D., Chen, D., Zhou, H., Meng, F., Zhou, J., and Sun, X. Label words are anchors: An information flow perspective for understanding in-context learning. *arXiv preprint arXiv:2305.14160*, 2023c.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022.

Wu, Z., Hu, Y., Shi, W., Dziri, N., Suhr, A., Ammanabrolu, P., Smith, N. A., Ostendorf, M., and Hajishirzi, H. Fine-grained human feedback gives better rewards for language model training. *arXiv preprint arXiv:2306.01693*, 2023.

Yu, D., Zhu, C., Fang, Y., Yu, W., Wang, S., Xu, Y., Ren, X., Yang, Y., and Zeng, M. Kg-fid: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. *arXiv preprint arXiv:2110.04330*, 2021.

Yu, W., Zhu, C., Li, Z., Hu, Z., Wang, Q., Ji, H., and Jiang, M. A survey of knowledge-enhanced text generation. *ACM Computing Surveys*, 54(11s):1–38, 2022.

Yu, W., Zhang, H., Pan, X., Ma, K., Wang, H., and Yu, D. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*, 2023.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

Zhang, M., Press, O., Merrill, W., Liu, A., and Smith, N. A. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*, 2023a.

Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023b.

Zhao, Y., Lin, C.-Y., Zhu, K., Ye, Z., Chen, L., Zheng, S., Ceze, L., Krishnamurthy, A., Chen, T., and Kasikci, B. Atom: Low-bit quantization for efficient and accurate llm serving. *arXiv preprint arXiv:2310.19102*, 2023.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S., and Chua, T.-S. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*, 2021.

Zhu, X., Li, J., Liu, Y., Ma, C., and Wang, W. A survey on model compression for large language models. *arXiv preprint arXiv:2308.07633*, 2023.

A. More Related Works

A.1. Language Models

Language models such as GPT (Radford et al., 2018), BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and Megatron-LM (Shoeybi et al., 2019) have led to a learning paradigm shift in natural language processing (NLP). Models are first pre-trained on extensive volumes of unlabeled text corpora with language modeling objectives, and then fine-tuned on downstream tasks. Recently, large language models (LLMs) including ChatGPT, PaLM (Chowdhery et al., 2022), and Gemini (Team, 2023) have shown great performance in both few-shot and even zero-shot scenarios (Brown et al., 2020).

A.2. Knowledge Hallucination

Besides the methods mentioned in Section 2.3 to address knowledge hallucinations during the generation time or through the RALM framework, these are some methods that address hallucinations during training time. These interventions during the training stage of LLMs to tackle the issue of model hallucinations are termed training-time correction. For training-time correction, efforts are made to enhance the raw left-to-right outputs of either an encoder-decoder or a decoder-only language model. This enhancement involves training or suitably adjusting the model's weights to reduce the likelihood of hallucinated content. This includes using reinforcement learning (Roit et al., 2023; Wu et al., 2023) as well as contrastive learning methods (Chern et al., 2023; Sun et al., 2023). For training-time correction methods, models designed to resolve knowledge hallucinations during the training phase typically require the use of open-source LLMs and substantial computational resources. Our COFT effectively reduces the hallucination issue in LLMs without the finetuning process. Moreover, LLMs after the training-time generation method can also be integrated as a part of our COFT pipeline.

A.3. Context Compression

One significant challenge in the computation of self-attention mechanisms is the computational complexity $\mathcal{O}(\mathcal{L}^2)$, which exhibits a quadratic scaling in relation to the length of the input sequence $\mathcal{L}$. Numerous variations of the Transformer architecture have been introduced, aiming to modify the conventional attention mechanism into more efficient alternatives specifically designed for tasks involving very long context (Zaheer et al., 2020; Katharopoulos et al., 2020). Extensive endeavors also focus on context compression by compressing the context into fewer soft tokens. This includes substitutes with summary tokens (Chevalier et al., 2023), leveraging additional auto-encoder schemes (Ge et al., 2023), and semantic compression (Fei et al., 2023). Sparse attention (Anagnostidis et al., 2023) adopts a methodology predicated on learning to dynamically excise uninformative context tokens for each individual token. Several efforts also select contexts to compress the input prompt (Li et al., 2023; Jiang et al., 2023a;b). However, due to the incomplete context, these methods may confront inevitable losses of information in real-world scenarios characterized by more complex distributions of attention.

B. More Details of Datasets and Experiment Setups

We present more details of datasets and experiment setups in this section.

Table 4. Statistics of the knowledge hallucination benchmark, FELM. #Segments denotes the total number of segments. Segment Length and reference Length denote the average length of the segment and reference texts, respectively. Size denotes the number of samples for each domain.

Dataset-Domain	#Segments	Segment Length	Reference Length	Size
FELM-WK	567	17.5	486.1	184
FELM-Sci/Tech	717	19.2	193.6	125
FELM-Wri/Rec	1637	18.4	141.7	136

For **more details of experiment setup**, in this paper, we use ChatGPT and GPT4 as the representatives of the current closed-source LLMs, both of which can be get access via OpenAI[4]. We apply Vicuna-33B (Zheng et al., 2023) as a representative of open-source LLMs. All experiments were performed on four Nvidia A100 GPUs (80GB). We implement our approach based on PyTorch 1.13.0[5] and Huggingface's Transformers[6]. For experiments with original prompts exceeding 4k tokens, we utilize extened length models, i.e., GPT-3.5-Turbo-16k

[4]https://platform.openai.com/

[5]https://pytorch.org/

[6]https://github.com/huggingface/transformers

and GPT-4-32k as our backbones. To guarantee stable and reproducible results, we utilize greedy decoding and set the temperature parameter as 0 in all experiments. For the small language models used for calculating self-information, we apply LLaMA-7B[7], and other open-source models can also be replaced based on specific requirements. More detailed configurations for the best performance of each task and dataset can be seen within our code.

Table 5. Statistics of the reading comprehension benchmarks, RACE-H and RACE-M. The values below the Training/Valid/Testing Set are the number of passages and questions in each dataset, respectively. Passage/Question/Option Len denotes the average length of the passages, questions, and options, respectively. Vocab size denotes the number of words in the vocabulary.

Dataset	Training Set	Valid Set	Testing Set	Passage Len	Question Len	Option Len	Vocab Size
RACE-M	6,409/25,421	368/1,436	362/1,436	231.1	9.0	3.9	32,811
RACE-H	18,728/62,445	1,021/3,451	1,045/3,498	353.1	10.4	5.8	125,120

For **more details of datasets**, we list below all the datasets and corresponding evaluation metrics used in knowledge hallucination, reading comprehension, and question-answering tasks, respectively by COFT as follows.

For the knowledge hallucination task, we employ FELM (Chen et al., 2023c) as our benchmark. Specifically, FELM requires to conduct a factual evaluation of several segments based on reference texts. This requires LLMs to categorize each segment as either true or false according to the given reference context. We utilize WK (world knowledge), Sci/Tech (science/technology), and Wri/Rec (writing/recommendation) domains as our knowledge hallucination benchmark. These datasets are derived from instances where individuals prompt ChatGPT and annotators subsequently annotate the responses for factuality evaluations. We summarize the details of this knowledge hallucination benchmark in Table 4. Following FELM (Chen et al., 2023c), we use precision, recall, and F1 score as our evaluation metrics.

For the reading comprehension task, we employ RACE-M (middle school level reading comprehension task) and RACE-H (high school level reading comprehension task) (Lai et al., 2017) as our benchmarks. RACE is collected from the English exams for middle and high school Chinese students in the age range between 12 to 18. RACE consists of nearly $28,000$ passages and nearly $100,000$ questions generated by human experts (English instructors), and covers a variety of topics that are carefully designed to evaluate the students' ability to understand and reasoning. The reasoning types of RACE include word matching, paraphrasing, single-sentence reasoning, multi-sentence reasoning, and insufficient/ambiguous. We summarize the details of this reading comprehension benchmark in Table 5. To better satisfy long-text reading comprehension tasks, we retain only those samples in RACE where the length of provided reading passages of top 70%. We also observe that the length of passages and the vocabulary size in RACE-H are significantly larger compared to RACE-M, indicating the greater difficulty level of high school examinations. For Vicuna-33B, we use one-shot setting. Following (Bi et al., 2024; Rae et al., 2021), we use precision as our evaluation metric.

For the question-answering task, we employ Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and WebQ (Berant et al., 2013) as our benchmarks. We present details of these datasets as follows:

Natural Questions (Kwiatkowski et al., 2019): natural questions corpus comprises real anonymized, aggregated queries directed to the Google search engine. An annotator is provided with a question and a corresponding Wikipedia page from the top 5 search results. They annotate a long answer (usually a paragraph) and a short answer (one or more entities) if they are found on the page, or they mark it as null if no long or short answer is identified. Natural Questions corpus offers a substantial dataset for end-to-end training in the field of question answering, facilitating research in natural language comprehension. It enables the study of human performance in annotating QA annotations for naturally generated questions, contributing to a better understanding of the challenges in this domain.

Table 6. Statistics of the question answering benchmarks. Full size denotes the original size of these benchmarks. The IR recall evaluation is based on the retrieval of the full test set. The subset refers to the remaining dataset obtained after removing the instances that could not be retrieved.

Dataset	Full Size	IR Recall	Subset Size
Natural Questions	3,610	73.82	1,477
TriviaQA	7,993	89.95	5,148
WebQ	2,032	64.22	1,073

TriviaQA (Joshi et al., 2017): TriviaQA is a very challenging reading comprehension dataset that consists of more than

[7]https://ai.meta.com/llama/

650K question-answer-evidence triples. It contains 95K question-answer pairs authored by trivia enthusiasts, accompanied by independently gathered evidence documents. On average, there are six evidence documents per question, which serve as high-quality supervision for answering the questions. TriviaQA possesses several notable characteristics: (1) It features relatively intricate and compositional questions. (2) There is substantial syntactic and lexical variability observed between questions and the corresponding answer-evidence sentences. (3) The dataset necessitates more extensive cross-sentence reasoning in order to locate answers.

WebQ (Berant et al., 2013): WebQ uses the Google Search API[8] to obtain questions that start with a specific word and contain precisely one entity. The Google Search API was employed to supply the edges of the graph. Specifically, WebQ queries the question by excluding the entity, the phrase before the entity, or the phrase after it. Each query generates five candidate questions, which are then added to the queue. This process continues until one million questions have been visited. Out of those, a random subset of $100,000$ questions is submitted to Amazon Mechanical Turk[9] (AMT). Workers on AMT are tasked with answering the questions using only the Freebase[10] page associated with the entity in the question. If the question is unanswerable based on Freebase, workers are instructed to mark it as such.

We follow the CoN (Yu et al., 2023) for text retrieval on these three datasets. During the process of listing retrieved documents, we set a rule to stop searching based on the number of relevant and irrelevant texts; we stop searching when both types reach our criteria. Situations wherein the DPR (Karpukhin et al., 2020) fails to retrieve pertinent documents for certain queries will not be included in our robustness evaluation. Furthermore, to better simulate the scenarios involving long context reasoning and robustness against noisy text, we establish a criterion: for each question-answering pair, we only retain those pairs where the retrieved text exceeds 1500 words in length. Pairs with retrieved text falling below this threshold are discarded. Consequently, the subset is more compact than the original full-size dataset set, as shown in Table 6.

C. More Results of Reading Comprehension

As we mentioned above, COFT can be effectively implemented across various NLP tasks for LLM long-form inference. In this section, we present more results of COFT with Vicuna-33B and GPT-4 as backbone models on the reading comprehension task to serve as a supplement to Section 5.3, where ChatGPT is employed as the backbone model. We observe from Table 7 that COFT consistently enhances performance across various LLM backbones in both RACE-H and RACE-M benchmarks. Specifically, COFT obtains superior performances of 11.6% and 3.1% in RACE-H and RACE-M for the Vicuna-33B model and 1.2% and 1.3% in RACE-H and RACE-M for GPT4 model, respectively. These results effectively demonstrate that COFT shows versatility under multiple LLMs as backbone models in the reading comprehension task, which also suggests that COFT effectively promotes LLMs to retain a comprehensive understanding of the long contextual semantics and to focus on keywords and phrases relevant to the question. Furthermore, we observe that COFT achieves more performance enhancement on the more challenging and complex dataset, RACE-H. This also suggests that COFT possesses potential for application in more complex real-world scenarios. Notably, when utilizing Vicuna-33B as the backbone model, COFT achieves 11.6% superior performance in precision metric on RACE-H over the suboptimal approaches. This also indicates the potential of COFT to better assist relatively "small" models in more effectively maintaining complete context semantics, focusing on key lexical units, and avoiding getting lost in the lengthy context. These findings also demonstrate the efficacy of COFT applicable in reading comprehension tasks, where complete contextual semantics are necessary.

Table 7. Results of the reading comprehension task in the precision metric, utilizing Vicuna-33B and GPT4 as the backbone models. We **bold** the best results for each backbone, respectively.

Backbone	Methods	RACE-H	RACE-M
Vicuna-33B	Vanilla	44.8	74.2
	CoT	43.7	76.6
	CoVe	56.8	78.2
	CoN	51.7	75.7
	COFT	**68.4**	**81.3**
GPT4	Vanilla	78.9	88.4
	CoT	87.9	88.6
	CoVe	78.8	89.7
	CoN	79.5	86.2
	COFT	**89.1**	**89.9**

[8]https://developers.google.com/custom-search

[9]https://www.mturk.com/

[10]https://developers.google.com/freebase

Table 8. The results of ablation study on the knowledge hallucination benchmark, FELM, using Vicuna-33B as the backbone model. We denote COFT without *recaller* as COFT$_{w/o\ recaller}$, COFT without TF-ISF score as COFT$_{w/o\ TF\text{-}ISF}$, COFT without self-information score as COFT$_{w/o\ SI}$, COFT without *scorer* as COFT$_{w/o\ scorer}$, and COFT without *selector* as COFT$_{w/o\ selector}$, respectively.

		WK			**Sci/Tech**			**Wri/Rec**		
Backbone	Methods	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
Vicuna-33B	COFT$_{w/o\ recaller}$	57.8	55.7	60.1	55.9	60.4	52.1	56.1	54.8	57.4
	COFT$_{w/o\ TF\text{-}ISF}$	60.7	56.5	65.5	57.9	55.4	60.7	64.6	63.5	65.7
	COFT$_{w/o\ SI}$	59.9	55.7	64.8	59.4	57.8	61.1	63.9	62.5	65.3
	COFT$_{w/o\ scorer}$	57.3	53.7	61.5	52.6	50.8	54.5	60.1	59.3	61.1
	COFT$_{w/o\ selector}$	60.3	57.9	62.8	64.7	59.1	71.4	67.7	62.8	73.5
	COFT	**64.4**	**61.7**	**67.4**	**70.9**	**65.7**	**77.2**	**77.3**	**67.9**	**89.8**

Table 9. The results of ablation study on the knowledge hallucination benchmark, FELM using GPT4 as the backbone model. We denote COFT without *recaller* as COFT$_{w/o\ recaller}$, COFT without TF-ISF score as COFT$_{w/o\ TF\text{-}ISF}$, COFT without self-information score as COFT$_{w/o\ SI}$, COFT without *scorer* as COFT$_{w/o\ scorer}$, and COFT without *selector* as COFT$_{w/o\ selector}$, respectively.

		WK			**Sci/Tech**			**Wri/Rec**		
Backbone	Methods	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
GPT4	COFT$_{w/o\ recaller}$	81.3	85.6	77.5	71.9	78.6	66.3	77.1	83.3	71.7
	COFT$_{w/o\ TF\text{-}ISF}$	82.7	89.5	76.9	74.5	79.2	70.4	81.5	86.5	77.1
	COFT$_{w/o\ SI}$	79.2	85.1	74.1	74.8	78.8	71.1	80.8	85.8	76.3
	COFT$_{w/o\ scorer}$	77.9	83.4	73.1	73.4	76.5	70.6	79.4	83.5	75.6
	COFT$_{w/o\ selector}$	80.9	84.2	77.9	74.0	81.8	67.5	78.6	88.7	70.5
	COFT	**87.3**	**94.8**	**80.9**	**77.9**	**86.0**	**71.3**	**84.5**	**92.9**	**77.9**

D. More Results of Question Answering

COFT also exhibits robustness against noise texts present in the reference contexts. We provide more question answering results illustrated in Figures 5, 6, 7, 8, and 9. We also provide detailed data tables as a numerical complement to the visual results in Section 5.4. As illustrated in Tables 11, 12, and 13, we observe that COFT is capable of maintaining relative robustness compared to other methods under conditions of severe noisy scenarios. This also demonstrates the effectiveness of mining key lexical and phrases relevant to the query. We further observe that as the noise ratio increases, that is, a greater proportion of irrelevant text in the reference context, COFT demonstrates enhanced robustness compared to other methods, thereby yielding relatively superior results. COFT achieves improvements or comparable results to baseline methods across nearly all conditions of *noise ratio*. Notably, under conditions where the *noise ratio* is 80%, COFT achieves a maximum improvement of 6.5% in EM metric and 10.5% in the F1 score metric when utilizing ChatGPT as the backbone model, which also demonstrates the noise robustness under similar nosy documents and the capability to focus on the highlighted key lexical units to the given query of our COFT.

Specifically, when ChatGPT serves as the backbone model and the *noise ratio* goes from 0% to 80%, on the Natural Questions dataset, COFT achieves average improvements of 4.3% in the EM metric and 3.4% in the F1 score metric. On the TriviaQA dataset, our COFT achieves average improvements of 2.7% in the EM metric, alongside 2.0% in the F1 score metric. Furthermore, on the WebQ dataset, COFT achieves average improvements of 2.3% in the EM metric and 4.7% in the F1 score metric. These results underscore the efficacy of the COFT approach in enhancing the performance of ChatGPT.

When Vicuna-33B serves as the backbone model, COFT has demonstrated notable improvements across different evaluation metrics. On the Natural Questions dataset, COFT achieves average improvements of 1.6% in the EM metric and 3.0% in the F1 score metric. On the TriviaQA dataset, COFT achieves average improvements of 2.0% in the EM metric, along with 2.5% in the F1 score metric. Furthermore, on the WebQ dataset, COFT achieves average improvements of 1.4% in the EM metric and 1.8% in the F1 score metric. These results underscore the efficacy of the COFT approach in enhancing the performance of Vicuna-33B.

When GPT4 serves as the backbone model, our COFT exhibits enhancements across various evaluation metrics as well.

Table 10. The inference time (per sample on average) on the FELM benchmark for the vanilla, CoT, RALM, CoVe, CoN, and COFT methods. We report the results using Vicuna-33B, ChatGPT, and GPT4 as backbone models, respectively. For Vicuna-33B, we deploy it locally and record the inference time. For ChatGPT and GPT4, we utilize the API interfaces provided by OpenAI to conduct inference and record the corresponding inference time. (Unit: seconds)

Backbone Models	Vanilla	CoT	RALM	CoVe	CoN	COFT
Vicuna-33B (Local Deployment)	29.01	29.87	31.15	37.41	35.01	31.72
ChatGPT (API)	3.86	4.15	4.45	7.24	5.13	4.50
GPT4 (API)	5.14	5.33	5.74	10.22	6.68	5.81

Specifically, on the Natural Questions dataset, COFT achieves average improvements of 1.5% in the EM metric and 2.0% in the F1 score metric. On the TriviaQA dataset, COFT achieves average improvements of 3.4% in the EM metric, along with 0.6% in the F1 score metric. Moreover, on the WebQ dataset, COFT achieves average improvements of 4.6% in the EM metric and 2.5% in the F1 score metric. These results highlight the effectiveness of COFT in enhancing the performance of GPT4 in these question-answering tasks.

These results further highlight COFT's efficacy in comparison to canonical methods. Such robustness to lengthy and noisy texts closely aligns with the real-world scenarios of user prompt inputs, effectively aiding LLMs in delivering more accurate responses. This effectively facilitates the practical deployment of LLMs in scenarios where high-precision and reliable answers are critically essential. Moreover, the effectiveness across various LLM backbones also demonstrates the potential of COFT to serve as a versatile plug-and-play framework over a wide range of long-form downstream NLP tasks.

E. More Results of Ablation Study

E.1. Detailed Ablation Results of *Selector*

In Section 5.5, we conduct the ablation study on *selector* by setting the threshold to a fixed value of 0.5, utilizing ChatGPT as the backbone model. In this section, we conduct a more detailed ablation study of the *selector*. We experiment with the threshold τ for *selector*, ranging from 0.1 to 1.0, and report the ablated results of Vicuna-33B, ChatGPT, and GPT4, respectively to provide more insight into our dynamic threshold algorithm.

As shown in Tables 14, 15, and 16. We still observe that our dynamic threshold algorithm achieves consistently superior and robust results against all other fixed thresholds. This effectively demonstrates the necessity of considering both the length and the amount of information of a given input reference context when setting the filtering thresholds to key lexical units. Moreover, the proposed dynamic threshold algorithm may potentially be beneficial to consider additional factors or optimize the combination method of context length and the amount of information to get improved results and we leave the exploration as a future work.

E.2. Ablation Results for Vicuna-33B and GPT4

In Section 5.5, we report the results of the ablation study using ChatGPT as the backbone model. In this section, we will further present the results using Vicuna-33B and GPT4 as backbone models to obtain more insights into the individual components constituting COFT across various backbone models. As illustrated in Tables 8 and 9, we still observe that the absence of each component within COFT invariably leads to a decline in performance across diverse domains for Vicuna-33B and GPT4 in the FELM benchmark, which demonstrates that COFT organically integrates these components into a unified framework as well.

Remarkably, we observe that in the absence of a *scorer*, i.e., *selector* randomly retains the top $\tau \times 100\%$ of key candidates obtained by the *recaller* using a dynamic threshold algorithm, rather than preserving them in descending order based on contextual weight, leads to a more significant decline in performance. This underscores the critical importance of effectively measuring the candidates' significance and highlights these candidates in reducing the issue of knowledge hallucination within LLMs as well.

These results underscore the organic integration of the three core components of COFT, *recaller*, *scorer*, and *selector*. This

Table 11. Results of question answering tasks in Natural Questions, TriviaQA, and WebQ benchmarks, utilizing ChatGPT as the backbone model. We evaluate the performance of each method in terms of EM and F1 score across various noise ratios (Yu et al., 2023). We **bold** the best results for each noise ratio, respectively.

			NQ		**TriviaQA**		**WebQ**	
Backbone	Methods	Noise Ratio	EM	F1 Score	EM	F1 Score	EM	F1 Score
ChatGPT	Vanilla	80%	25.9	37.1	67.4	78.3	13.9	35.1
	CoT		24.5	36.3	69.6	78.9	13.8	37.7
	CoVe		27.1	45.9	68.8	78.9	24.4	40.6
	CoN		23.9	43.3	68.5	77.9	17.2	35.4
	COFT		33.6	51.9	74.3	82.1	27.6	51.1
	Vanilla	60%	37.7	48.1	70.4	81.2	33.7	50.1
	CoT		36.1	49.4	72.2	80.5	34.5	47.3
	CoVe		32.8	51.6	71.3	80.4	35.7	44.6
	CoN		37.7	50.5	68.7	80.5	34.5	49.5
	COFT		43.2	55.9	75.1	83.7	37.3	55.3
	Vanilla	40%	37.0	51.8	73.3	82.5	31.1	50.4
	CoT		40.7	57.6	73.7	81.1	31.9	49.3
	CoVe		39.3	57.4	72.5	80.4	34.7	50.1
	CoN		42.7	56.8	70.4	80.5	35.1	51.7
	COFT		46.4	59.9	76.6	84.7	35.7	54.7
	Vanilla	20%	44.4	59.2	73.5	85.1	35.5	51.0
	CoT		46.2	58.5	75.4	85.7	35.7	50.3
	CoVe		42.1	63.4	76.7	82.3	34.4	50.0
	CoN		46.7	62.0	71.9	82.1	35.7	53.1
	COFT		50.2	65.1	79.2	87.4	38.7	56.5
	Vanilla	0%	49.8	62.9	75.5	86.4	35.4	52.6
	CoT		51.6	65.9	75.3	88.3	35.5	52.5
	CoVe		44.8	64.5	75.4	85.7	34.5	53.7
	CoN		50.6	64.1	78.7	88.7	35.5	58.4
	COFT		53.9	68.7	79.2	89.3	38.7	59.9

Table 12. Results of question-answering tasks in Natural Questions, TriviaQA, and WebQ benchmarks, utilizing Vicuna-33B as the backbone model. We evaluate the performance of each method in terms of EM and F1 score across various noise ratios (Yu et al., 2023). We **bold** the best results for each noise ratio, respectively.

			NQ		**TriviaQA**		**WebQ**	
Backbone	Methods	Noise Ratio	EM	F1 Score	EM	F1 Score	EM	F1 Score
Vicuna-33B	Vanilla	80%	15.7	22.1	42.4	50.1	9.2	13.8
	CoT		17.2	25.8	43.1	51.5	9.6	15.5
	CoVe		16.4	24.9	45.2	53.7	11.7	16.4
	CoN		16.2	24.3	46.8	54.9	10.5	15.5
	COFT		19.7	30.6	49.2	58.6	13.8	21.0
	Vanilla	60%	17.9	27.6	46.8	57.7	15.7	20.5
	CoT		17.8	28.1	48.5	59.2	13.1	21.1
	CoVe		19.9	27.9	49.1	61.4	14.8	24.1
	CoN		17.3	28.5	49.5	60.6	15.7	26.4
	COFT		21.3	32.8	52.8	63.1	16.2	28.2
	Vanilla	40%	18.4	28.5	52.1	63.0	14.9	24.7
	CoT		18.9	29.4	53.5	62.5	14.4	25.7
	CoVe		22.5	32.1	53.4	62.8	15.8	28.4
	CoN		20.1	31.6	52.8	63.4	15.5	27.3
	COFT		23.7	35.0	55.9	65.8	19.7	30.4
	Vanilla	20%	19.4	32.6	54.3	63.3	20.4	29.8
	CoT		20.7	32.9	54.9	63.9	20.9	30.5
	CoVe		23.5	33.7	55.3	64.3	22.5	31.7
	CoN		21.9	34.3	55.7	63.4	21.8	30.2
	COFT		25.5	36.2	57.3	67.5	22.4	33.4
	Vanilla	0%	21.5	34.9	56.9	66.5	22.8	32.7
	CoT		24.5	35.5	57.6	64.2	20.4	31.5
	CoVe		25.7	38.1	59.4	66.7	25.8	35.7
	CoN		24.2	37.6	58.9	65.8	24.6	36.9
	COFT		26.7	39.5	59.8	68.4	26.5	35.9

Table 13. Results of question answering tasks in Natural Questions, TriviaQA, and WebQ benchmarks, utilizing GPT4 as the backbone model. We evaluate the performance of each method in terms of EM and F1 score across various noise ratios (Yu et al., 2023). We **bold** the best results for each noise ratio, respectively.

			NQ		**TriviaQA**		**WebQ**	
Backbone	Methods	Noise Ratio	EM	F1 Score	EM	F1 Score	EM	F1 Score
GPT4	Vanilla	80%	53.3	60.1	45.3	58.5	19.7	38.7
	CoT		54.1	62.4	49.6	60.1	20.7	40.6
	CoVe		53.4	60.9	44.5	59.1	16.7	39.3
	CoN		54.3	61.8	50.8	60.4	6.9	31.9
	COFT		57.3	65.4	54.4	62.7	24.6	43.0
	Vanilla	60%	54.9	63.2	49.8	65.5	20.1	40.1
	CoT		57.3	66.1	55.2	61.5	20.7	40.4
	CoVe		54.0	64.4	45.4	61.4	20.3	41.9
	CoN		56.8	66.1	50.0	66.8	10.3	33.3
	COFT		59.6	67.8	57.8	68.4	24.1	44.1
	Vanilla	40%	57.1	66.3	54.8	65.4	19.4	41.5
	CoT		57.7	67.8	56.7	67.6	22.6	42.5
	CoVe		58.4	66.1	55.4	64.1	21.4	41.1
	CoN		58.2	67.1	55.6	66.1	16.4	36.0
	COFT		59.1	70.3	59.3	72.4	25.8	44.8
	Vanilla	20%	60.6	68.8	58.8	75.3	16.1	39.5
	CoT		61.1	69.4	58.3	72.5	22.8	41.8
	CoVe		58.8	70.0	55.7	70.9	22.7	39.3
	CoN		61.4	69.8	57.8	70.8	16.5	36.6
	COFT		62.4	72.1	62.3	74.1	28.8	44.6
	Vanilla	0%	63.5	74.8	62.5	81.4	19.4	42.8
	CoT		63.8	74.4	62.5	82.1	22.6	41.7
	CoVe		59.6	71.4	62.2	73.3	23.3	43.4
	CoN		63.2	74.1	59.8	76.7	16.8	36.2
	COFT		64.3	75.7	66.7	82.3	29.7	46.2

integration is not merely additive but forms a cohesive framework that significantly reduces the problem of knowledge hallucination in LLMs.

Table 14. A more detailed ablation study for *selector* on the knowledge hallucination benchmark, FELM, using ChatGPT as the backbone model. To demonstrate the superiority of our dynamic threshold algorithm, we set the threshold τ ranging from 0.1 to 1.0.

		WK			**Sci/Tech**			**Wri/Rec**		
Backbone	Threshold	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
ChatGPT	$\tau = 0.1$	76.8	77.9	75.7	78.0	82.6	74.0	71.8	87.5	60.9
	$\tau = 0.2$	69.8	70.6	69.1	72.6	78.2	67.7	68.1	85.1	56.7
	$\tau = 0.3$	77.5	82.0	73.5	74.5	80.9	69.0	73.9	87.3	64.1
	$\tau = 0.4$	75.2	82.5	69.1	74.5	84.3	66.7	71.5	87.3	60.5
	$\tau = 0.5$	77.3	79.7	75.1	75.7	82.4	70.1	70.7	86.1	60.1
	$\tau = 0.6$	78.7	84.7	63.2	81.4	84.8	78.3	72.3	85.1	62.8
	$\tau = 0.7$	75.4	75.1	75.7	81.6	81.3	81.9	56.6	81.8	43.3
	$\tau = 0.8$	66.2	69.4	63.2	72.4	80.4	65.9	70.8	84.4	60.9
	$\tau = 0.9$	74.5	73.9	75.0	61.7	86.1	48.1	69.3	88.0	57.1
	$\tau = 1.0$	77.9	81.5	74.6	82.8	83.4	82.2	61.2	82.2	48.7
	COFT	**81.6**	**85.5**	**77.9**	**85.5**	**86.5**	**84.5**	**75.2**	**88.3**	**65.4**

Table 15. A more detailed ablation study for *selector* on the knowledge hallucination benchmark, FELM, using Vicuna-33B as the backbone model. To demonstrate the superiority of our dynamic threshold algorithm, we set the threshold τ ranging from 0.1 to 1.0.

		WK			**Sci/Tech**			**Wri/Rec**		
Backbone	Threshold	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
Vicuna-33B	$\tau = 0.1$	56.6	55.8	57.5	63.2	64.0	62.4	60.8	52.3	72.5
	$\tau = 0.2$	54.2	53.9	55.1	65.9	53.4	70.2	72.3	62.1	86.4
	$\tau = 0.3$	54.0	56.8	51.5	67.3	63.4	71.7	69.7	62.8	78.2
	$\tau = 0.4$	52.9	54.5	51.3	69.7	62.4	75.2	67.5	56.8	83.3
	$\tau = 0.5$	57.3	53.7	61.5	52.6	50.8	54.5	60.1	59.3	61.1
	$\tau = 0.6$	57.5	57.5	57.6	67.0	62.7	72.0	73.2	64.7	84.2
	$\tau = 0.7$	60.9	59.6	62.2	67.6	60.6	76.5	67.8	60.3	77.4
	$\tau = 0.8$	53.0	52.5	53.6	66.1	58.7	75.6	62.6	51.9	78.8
	$\tau = 0.9$	50.2	64.1	41.3	63.8	62.9	64.7	65.4	60.3	71.5
	$\tau = 1.0$	52.2	48.9	55.9	62.4	59.1	66.1	66.5	55.5	82.8
	COFT	**64.4**	**61.7**	**67.4**	**70.9**	**65.7**	**77.2**	**77.3**	**67.9**	**89.8**

F. Inference Time Comparisons

We note that COFT requires an additional process of highlighting the input text before feeding it into the LLM for reasoning. Compared to the vanilla model, this process could potentially introduce extra inference time. Hence, in this section, we record and compare the average inference time per sample of different methods including vanilla, CoT, RALM, CoN, CoVe, and COFT on the knowledge hallucination benchmark, FELM, to explore the influence of additional inference time and provide more insight of our COFT. We report the word-level granularity COFT as an example, as the inference times for COFT at three different granularity levels (paragraph level, sentence level, and word level) are nearly identical.

We report the average inference time per sample as a metric, as shown in Table 10. We observe from the table that although the incorporation of COFT as a preprocessing module for LLM introduces additional inference time costs, this impact is marginal. On average, the increase in inference time cost per sample due to the introduction of COFT, compared to the vanilla model, is 12%. Notably, when utilizing accelerated APIs such as GPT, this additional inference time is less than one second, yet it offers an average improvement of 33.2% and a maximum of 60.5% in the F1 score for existing LLMs to reduce the issue of knowledge hallucination. Furthermore, COFT exhibits higher inference efficiency compared to methods

Table 16. A more detailed ablation study for *selector* on the knowledge hallucination benchmark, FELM, using GPT4 as the backbone model. To demonstrate the superiority of our dynamic threshold algorithm, we set the threshold τ ranging from 0.1 to 1.0.

		WK			**Sci/Tech**			**Wri/Rec**		
Backbone	Threshold	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
GPT4	$\tau = 0.1$	74.6	89.3	64.1	65.4	80.7	55.0	61.0	78.3	50.0
	$\tau = 0.2$	73.7	92.0	61.5	69.1	80.7	55.0	67.8	62.1	74.7
	$\tau = 0.3$	67.7	91.3	53.8	72.7	82.4	65.1	44.2	63.0	34.0
	$\tau = 0.4$	75.8	92.6	64.1	65.1	81.4	54.3	69.5	63.0	77.5
	$\tau = 0.5$	77.9	83.4	73.1	73.4	76.5	70.6	79.4	83.5	75.6
	$\tau = 0.6$	83.3	90.9	76.9	48.1	75.4	35.3	78.5	84.3	73.5
	$\tau = 0.7$	74.6	89.3	64.1	68.7	82.5	58.8	80.6	84.7	76.9
	$\tau = 0.8$	76.5	89.7	66.7	67.8	77.9	60.0	77.9	79.2	76.7
	$\tau = 0.9$	80.0	90.3	71.8	61.5	79.9	50.0	67.3	78.3	59.0
	$\tau = 1.0$	80.1	90.3	71.8	36.3	84.5	23.1	57.1	70.6	48.0
	COFT	**87.3**	**94.8**	**80.9**	**77.9**	**86.0**	**71.3**	**84.5**	**92.9**	**77.9**

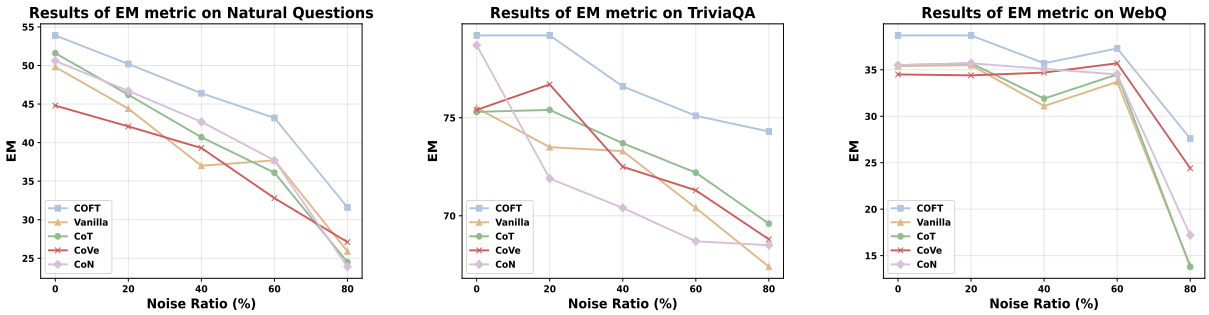

Figure 5. Evaluation on EM metric of noise robustness in question answering task, utilizing ChatGPT as the backbone model: COFT demonstrates superior performance on all three open-domain QA benchmarks, especially at higher noise ratios.

such as CoN, and CoVe, indicating that the additional computational overhead introduced by COFT is limited. We may focus on exploring ways to further reduce the time cost of the COFT, including lightweighting small language models to get a faster calculation of contextual weight(Zhu et al., 2023; Hsieh et al., 2023) or adopting more efficient (Wang et al., 2023a; Frantar & Alistarh, 2023; Zhao et al., 2023) and rational large model inference strategies such as speculative decoding (Chen et al., 2023b; Leviathan et al., 2023) as for future works.

G. More Results of Smaller Self-information Calculator

In Section 5.2, we utilize Llama 7B as the self-information calculator due to its great performance across a wide range of downstream tasks (Touvron et al., 2023). To further demonstrate the generalization and versatility of COFT across models of smaller scales, we conduct additional experiments using GPT-2 small (124M), GPT-2 medium (355M), GPT-2 large (744M), and GPT-2 XL (1.5B) to calculate the self-information, respectively (Radford et al., 2019). As shown in Tables 17, 18, 19 and 20, COFT consistently exhibits superior performance across all baseline methods, which demonstrates the effectiveness and potentially broad applications to smaller models.

H. More In-depth Analysis of COFT

H.1. Comparison Results of Adding Special Prompt

We conduct experiments of the baseline methods with the addition of the prompt ”Please pay close attention to the most relevant content in the text” on the FELM benchmark for the knowledge hallucination task. As shown in Table 21 , we

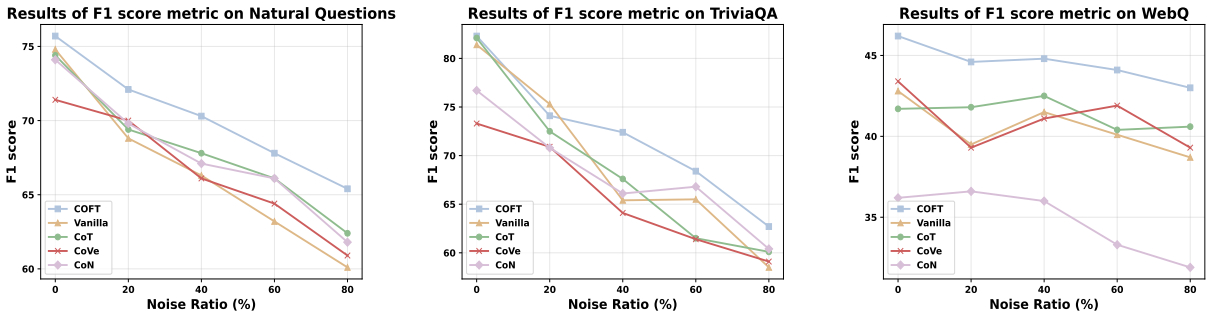

Figure 6. Evaluation on F1 score metric of noise robustness in question answering task, utilizing GPT4 as the backbone model: COFT demonstrates superior performance on all three open-domain QA benchmarks, especially at higher noise ratios.

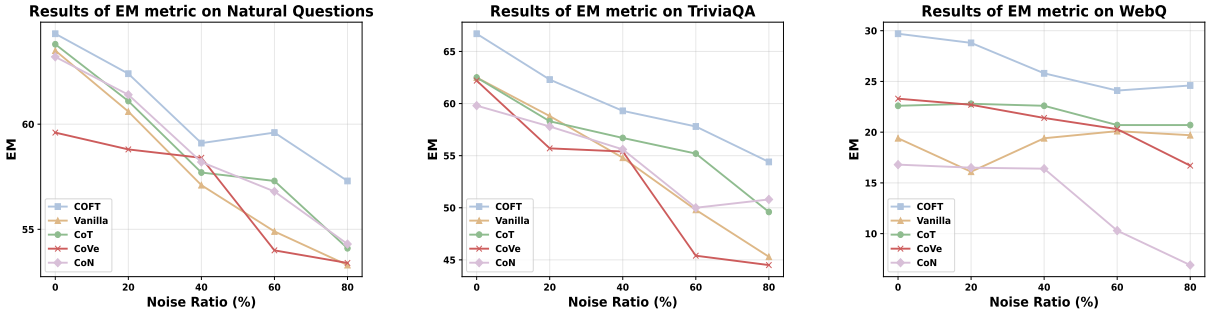

Figure 7. Evaluation on EM metric of noise robustness in question answering task, utilizing GPT4 as the backbone model: COFT demonstrates superior performance on all three open-domain QA benchmarks, especially at higher noise ratios.

find that while the inclusion of this prompt leads to marginal improvements in the overall performance, it can also result in a decline in performance in certain cases. This suggests that simply appending the prompt can not consistently enhance performance.

In contrast, COFT has demonstrated significant improvements, with an average increase of 30% and a maximum improvement of 60.5% in F1 scores across the knowledge hallucination benchmark, which further demonstrates the great benefits of COFT.

H.2. Two-hop Neighborhood Results of COFT

COFT exhibits excellent scalability and can be extended to multi-hop neighbor situations. COFT initially focuses on one-hop neighbors of candidate entities within the KG due to their intrinsic relevance and close association. And leveraging only a single hop from the neighbors results in an average increase of 30% and a maximum improvement of 60.5% in the F1 score on the knowledge hallucination task.

We conduct additional experiments incorporating two-hop neighbor information of COFT on the knowledge hallucination benchmark. As shown in Table 22, compared to the one-hop version of COFT, integrating two-hop neighbor information further enriches the input provided to the LLMs, leading to a moderate performance improvement over the one-hop scenario.

Despite the primary focus on one-hop neighbors, COFT maintains great performance. This demonstrates the effectiveness of extracting one-hop neighbors. When incorporating two-hop information, COFT further achieves a better result, which also demonstrates the flexibility and scalability of COFT. Therefore, for more complex question scenarios, there are also potential benefits of incorporating two-hop or even multi-hop neighbors to further increase the performance of COFT.

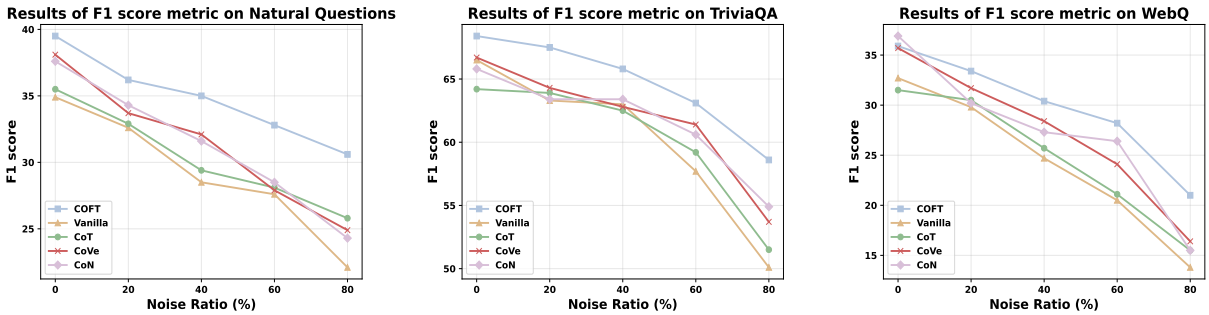

Figure 8. Evaluation on F1 score metric of noise robustness in question answering task, utilizing Vicuna-33B as the backbone model: COFT demonstrates superior performance on all three open-domain QA benchmarks, especially at higher noise ratios.

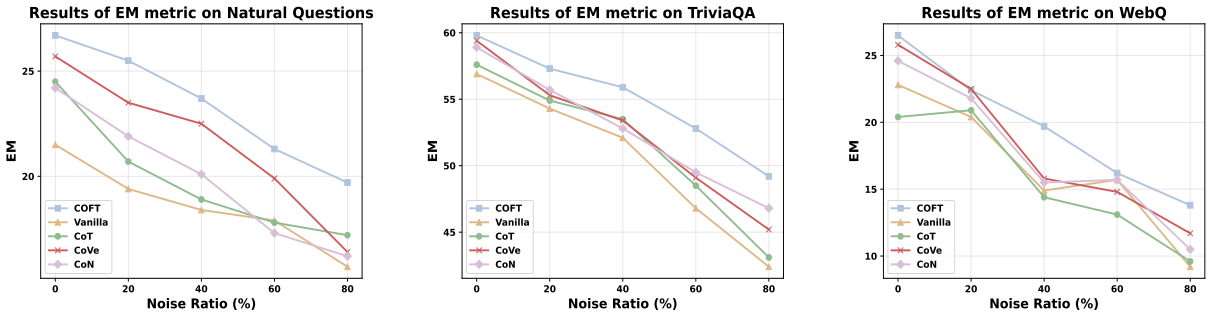

Figure 9. Evaluation on F1 score metric of noise robustness in question answering task, utilizing Vicuna-33B as the backbone model: COFT demonstrates superior performance on all three open-domain QA benchmarks, especially at higher noise ratios.

H.3. Only Input Highlights for LLM Inference

We incorporate specific symbols to highlight these units within the context to preserve the complete contextual semantics. The absence of complete contextual semantics may face inevitable information loss in real scenarios with more complex attention distributions (Miao et al., 2023b).

We further conduct experiments on the knowledge hallucination dataset, where we only use the highlighted lexical units as input to LLMs. As shown in Table 23, we find that even when only the highlighted lexical units are provided as reference context, the model achieves notable improvements over the baseline methods. This outcome demonstrates the efficacy of our COFT approach in accurately identifying and leveraging support facts within the reference text, thereby enhancing the inference performance. Meanwhile, compared to Table 1, we find that COFT uses only the highlighted lexical units as input is less competitive than the original version of COFT, which also demonstrates the effectiveness of our highlight mechanism.

H.4. More Results of Analyzing Position Bias of COFT

We further conduct experiments to explore the impact of position bias in the QA task. Specifically, each reference context comprised relevant documents and irrelevant documents. Drawing upon (Liu et al., 2023), we experiment by varying the positioning of the correct text from the first to the fifth position. As shown in Table 24, we find that COFT is less influenced by the position bias compared to Vanilla LLMs. This demonstrates COFT's robustness to the positioning of the correct text and implies the great potential to handle lengthy contexts in real-world scenarios.

H.5. More Results of Randomly Selecting Highlights of COFT

We also conduct additional experiments on the knowledge hallucination task. For each query, we randomly highlight lexical units (paragraphs, sentences, or words) that align with the number of highlighted key lexical units in original COFT.

As shown in Table 25, we find that randomly highlighting lexical units can not improve the results, which demonstrates the

Instruction:

I will show you a question, a list of text segments, and a reference text. All the segments can be concatenated to form a complete answer to the question. Your task is to assess whether each text segment contains factual errors or not based on the reference text. Please pay close attention to the content in the symbol ** ** in the reference text. Please generate using the following format.

Answer:
List the judgment results for each segments, such as: [True, False, ...]. Please only output the list, no more details. Note that the length of the list should be same as the number of segments.
Below are my inputs.

Question:

Have there been any US presidents shorter than 5ft?
Segments: ['Yes, there have been several US presidents who were shorter than 5 feet tall.', 'James Madison, the fourth president of the United States, was only 5 feet 4 inches tall.']

Reference text:

A record of the heights of the presidents of the ****United States**** and presidential candidates is useful for evaluating what role, if any, height plays in presidential elections. Some observers have noted that the taller of the two major-party candidates tends to prevail, and argue this is due to the public's preference for taller candidates. The tallest U.S. president was Abraham Lincoln at 6 ****feet**** 4 ****inches**** (193 centimeters), while the shortest was ****James Madison**** at 5 ****feet**** 4 ****inches**** (163 centimeters)…..

Figure 10. Prompt templates of knowledge hallucination task after highlighting the key lexical units. We use a sample prompt template across Vicuna-33B, ChatGPT, and GPT4.

effectiveness of COFT to identify key lexical units that are relevant to the query.

H.6. More Results of Joint-level Highlight Version of COFT

In practical applications, within a document, some paragraphs may be too short and might not require paragraph-level highlightintg, while others may be too long for word-level highlighting.

Therefore, we also conduct experiments using a joint-level highlighting version of COFT on the knowledge hallucination task. We highlight key lexical units by word-level granularity. If more than one-third of the words within a sentence were highlighted, we highlight the whole sentence. Similarly, if more than one-third of the sentences within a paragraph were highlighted, we would highlight the whole paragraph.

As shown in Table 26, COFT at the joint level yields an improvement over using single-level lexical units such as words, sentences, or paragraphs. We will include the joint level version of COFT in Table 1 of the main text to have a more comprehensive understanding of COFT. This suggests that exploring more joint-level highlighting strategies could be a promising direction for COFT.

I. Prompt Templates for Each Task

We list the prompt templates for different tasks to offer more visually intuitive results in Figures 10, 11, and 12 for knowledge hallucination, reading comprehension, and question answering, respectively. More detailed prompt information for the best performance of each task and dataset can be seen within the code.

Instruction:

Please choose the best option based on the article and question. Please pay close attention to the content in the symbol** **in the article. If the content in ** ** is not important, it can be ignored.

Question:

We can infer from the passage that _ .
Options:
A. about 50% of first marriages end in divorce in the U.S, much higher than other parts of the world
B. never tie the knot before you make sure he or she will not divorce you
C. it usually takes 7-10 years to rebuild one's credit if it is broken for any reason
D. it's unusual for people in their 20's to have money these days

Reference text:

Being young is great. Most of the **parts** of your body still work great, you have a full head of hair, you're energetic, and you have a **world** of opportunity in front of you. However, there's going to come a time when you start to get older. And as you get older, you'll have new responsibilities, complete independence, and perhaps most importantly, less time to recover from mistakes. You see, we all make mistakes in life. Maybe you spent more **money** than you should have on a car, you passed up on a great job opportunity, or you didn't try as hard as you could have in school…..

Figure 11. Prompt templates of reading comprehension task after highlighting the key lexical units. We use a sample prompt template across Vicuna-33B, ChatGPT, and GPT4.

Table 17. Results of knowledge hallucination benchmark on WK (world knowledge), Sci/Tech (science and technology), and Wri/Rec (writing and recommendation) domains using GPT2-small (124M) to calculate the contextual weight. We denote COFT at the paragraph, sentence, and word levels as COFT_p, COFT_s, and COFT_w, respectively.

Backbone	Method	WK			Sci/Tech			Wri/Rec		
		F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
Vicuna-33B	COFT_w	63.7	62.4	65.1	69.1	63.8	75.3	74.6	69.8	80.1
	COFT_s	61.3	60.6	62.1	66.9	63.9	70.1	69.9	65.5	74.9
	COFT_p	64.2	65.3	63.2	69.2	66.6	72.1	75.9	69.9	83.1
ChatGPT	COFT_w	74.0	77.9	70.4	79.6	79.1	80.1	78.1	86.1	71.5
	COFT_s	71.8	70.5	73.1	74.3	75.5	73.1	75.6	77.9	73.5
	COFT_p	77.3	81.8	73.3	79.6	75.3	84.4	79.7	85.5	74.7
GPT-4	COFT_w	74.0	77.9	70.4	79.6	79.1	80.1	78.1	86.1	71.5
	COFT_s	71.8	70.5	73.1	74.3	75.5	73.1	75.6	77.9	73.5
	COFT_p	77.3	81.8	73.3	79.6	75.3	84.4	79.7	85.5	74.7

Instruction:

Please refer to the following text and answer the following question in simple words. Please note that no explanation should be provided except for the answer. Please pay close attention to the content in the symbol** **in the article. If the content in ** ** is not important, it can be ignored.

Question:

when does the miz and maryse show start

Reference text:

n late-2011, she announced plans for a clothing and jewelry line called House of **Maryse**, and later began working as a realtor. In November 2016, she began starring in the reality television series Total Divas on E! as part of the main cast. She and her husband The **Miz** will also star in their own reality **show** titled **Miz** & Mrs. that will premiere in 2018. off WWE programming. After two months of inactivity, **Maryse** was released from her WWE contract on October 28. On October 5, 2012, **Maryse** appeared at the Family Wrestling Entertainment (FWE) event "Back 2 Brooklyn", performing live commentary. She began appearing regularly for FWE, where she commentated during women's matches…..

Figure 12. Prompt templates of question answering task after highlighting the key lexical units. We use a sample prompt template across Vicuna-33B, ChatGPT, and GPT4.

Table 18. Results of knowledge hallucination benchmark on WK (world knowledge), Sci/Tech (science and technology), and Wri/Rec (writing and recommendation) domains using GPT2-medium (355M) to calculate the contextual weight. We denote COFT at the paragraph, sentence, and word levels as COFT$_p$, COFT$_s$, and COFT$_w$, respectively.

Backbone	Method	WK			Sci/Tech			Wri/Rec		
		F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
Vicuna-33B	COFT_w	65.2	62.0	68.8	70.8	66.3	76.0	72.6	62.8	85.9
	COFT_s	62.8	60.6	65.2	67.9	63.1	73.5	69.5	65.5	74.0
	COFT_p	66.9	66.8	67.1	65.0	60.6	70.1	71.2	62.7	82.4
ChatGPT	COFT_w	75.8	80.1	71.9	72.4	85.3	62.9	77.5	79.9	75.3
	COFT_s	70.9	77.9	65.1	73.7	80.7	67.9	74.8	85.1	66.8
	COFT_p	80.4	83.5	77.5	77.4	83.5	72.1	78.7	84.6	73.6
GPT-4	COFT_w	81.1	83.3	79.1	79.7	80.1	79.4	84.5	80.1	89.3
	COFT_s	79.1	85.1	73.9	76.9	80.3	73.7	80.8	83.2	78.5
	COFT_p	81.8	85.9	78.0	76.4	79.1	73.9	80.9	90.5	73.1

Table 19. Results of knowledge hallucination benchmark on WK (world knowledge), Sci/Tech (science and technology), and Wri/Rec (writing and recommendation) domains using GPT2-large (744M) to calculate the contextual weight. We denote COFT at the paragraph, sentence, and word levels as COFT$_p$, COFT$_s$, and COFT$_w$, respectively.

Backbone	Method	WK			Sci/Tech			Wri/Rec		
		F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
Vicuna-33B	COFT_w	63.3	61.1	65.7	69.8	63.2	78.0	75.8	65.9	89.3
	COFT_s	63.2	62.6	63.8	69.3	64.5	74.8	72.8	60.6	91.2
	COFT_p	63.6	60.4	67.2	67.0	61.3	73.9	70.3	64.8	76.9
ChatGPT	COFT_w	73.7	76.1	71.5	80.2	73.5	88.2	78.4	79.3	77.6
	COFT_s	75.2	69.9	81.3	74.1	83.1	66.9	73.9	74.6	73.3
	COFT_p	75.9	74.5	77.4	83.1	82.5	83.7	77.3	80.6	74.2
GPT-4	COFT_w	82.0	85.2	79.1	84.3	80.1	88.9	88.3	84.1	92.9
	COFT_s	78.2	88.5	70.1	75.8	78.4	73.3	86.3	85.3	87.4
	COFT_p	82.9	87.9	78.4	80.6	82.8	78.5	86.5	88.2	84.9

Table 20. Results of knowledge hallucination benchmark on WK (world knowledge), Sci/Tech (science and technology), and Wri/Rec (writing and recommendation) domains using GPT2-XL (1.5B) to calculate the contextual weight. We denote COFT at the paragraph, sentence, and word levels as COFT$_p$, COFT$_s$, and COFT$_w$, respectively.

Backbone	Method	WK			Sci/Tech			Wri/Rec		
		F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
Vicuna-33B	COFT$_w$	63.5	62.9	64.1	69.5	65.5	74.1	78.6	71.3	87.5
	COFT$_s$	66.0	73.1	60.1	65.2	71.3	60.1	74.5	66.9	84.1
	COFT$_p$	65.2	69.1	61.7	69.4	67.2	71.8	71.5	65.1	79.3
ChatGPT	COFT$_w$	73.5	81.5	66.9	79.4	74.2	85.4	74.7	86.4	65.8
	COFT$_s$	76.1	83.3	70.1	78.1	73.9	82.7	76.7	80.5	73.3
	COFT$_p$	77.9	80.5	75.4	81.3	77.8	85.2	78.7	84.1	73.9
GPT4	COFT$_w$	82.1	83.8	80.5	83.8	79.2	89.0	88.1	88.3	87.9
	COFT$_s$	80.1	82.9	77.5	80.1	82.9	77.4	82.7	79.6	86.1
	COFT$_p$	85.0	89.4	81.1	82.7	79.9	85.7	85.0	84.7	85.3

Table 21. Results of knowledge hallucination benchmark on WK (world knowledge), Sci/Tech (science and technology), and Wri/Rec (writing and recommendation) domains by adding the prompt "Please pay close attention to the most relative content in the text".

		WK			**Sci/Tech**			**Wri/Rec**		
Backbone	**Methods**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**
Vicuna-33B	Vanilla	35.1	29.6	43.1	26.2	17.9	48.8	30.3	19.2	72.0
	CoT	35.0	31.1	40.1	24.5	15.9	53.3	30.5	19.9	65.4
	RALM	47.2	44.2	50.6	36.1	27.3	53.1	31.2	19.4	80.1
	CoVe	46.7	45.1	48.5	47.7	41.1	56.7	64.4	59.2	70.5
	CoN	54.3	55.1	53.5	59.7	56.0	63.9	66.2	60.3	73.3
ChatGPT	Vanilla	12.8	27.9	8.3	4.3	6.3	3.3	2.1	5.5	1.3
	CoT	8.3	30.9	4.8	7.9	23.9	4.7	6.3	11.9	4.3
	RALM	25.7	33.7	20.8	18.4	18.8	18.0	23.1	58.3	14.4
	CoVe	18.8	46.7	11.8	18.8	13.7	29.8	21.1	60.4	12.8
	CoN	18.0	63.0	10.5	21.9	24.4	19.9	31.3	28.9	34.2
GPT-4	Vanilla	39.7	80.3	26.4	21.7	63.5	13.1	25.6	84.4	15.1
	CoT	53.2	82.1	39.3	27.6	61.0	17.8	27.0	84.5	16.1
	RALM	53.9	78.8	41.0	31.4	55.7	21.9	48.4	70.3	36.9
	CoVe	49.2	50.5	47.9	70.1	85.5	59.4	50.9	51.2	50.7
	CoN	55.0	50.1	61.0	68.7	80.3	60.1	69.3	75.1	64.3

Table 22. Results of knowledge hallucination benchmark on WK (world knowledge), Sci/Tech (science and technology), and Wri/Rec (writing and recommendation) domains by incorporating two-hop neighbor information of COFT. We denote COFT at the paragraph, sentence, and word levels by COFT_p, COFT_s, and COFT_w, respectively.

		WK			Sci/Tech			Wri/Rec		
Backbone	Methods	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
Vicuna-33B	COFT_p	69.5	73.3	66.0	67.0	63.5	71.0	69.2	64.3	75.0
	COFT_s	61.9	60.6	63.2	72.1	69.5	74.9	67.2	61.5	74.1
	COFT_w	63.7	63.5	64.0	68.3	63.3	74.2	77.0	74.8	79.4
ChatGPT	COFT_p	78.4	79.5	77.4	83.5	83.1	83.9	78.7	90.5	69.7
	COFT_s	76.1	77.1	75.1	78.3	75.5	81.3	79.2	94.9	67.9
	COFT_w	82.2	86.8	78.1	86.4	83.9	89.0	83.7	92.8	76.3
GPT4	COFT_p	88.2	90.0	86.5	89.7	92.6	87.0	92.6	88.4	97.3
	COFT_s	84.1	88.1	80.5	78.7	80.5	77.0	88.3	91.2	85.5
	COFT_w	92.4	93.3	91.5	81.4	90.7	73.9	87.8	93.8	82.5

Table 23. Results of knowledge hallucination benchmark on WK (world knowledge), Sci/Tech (science and technology), and Wri/Rec (writing and recommendation) domains by only making use of the selected key lexical units as input to the LLM. We denote COFT at the paragraph, sentence, and word levels as COFT_p, COFT_s, and COFT_w, respectively.

Backbone	Method	WK			Sci/Tech			Wri/Rec		
		F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
Vicuna-33B	COFT_p	69.0	69.8	68.3	61.3	57.7	65.3	65.4	60.6	71.1
	COFT_s	61.1	60.1	62.1	61.8	60.5	63.2	61.7	60.0	63.5
	COFT_w	60.1	57.1	63.5	61.9	62.8	61.1	70.1	63.1	78.8
ChatGPT	COFT_p	74.2	80.0	69.2	77.3	78.6	76.1	70.9	76.8	65.8
	COFT_s	71.1	69.4	72.8	74.5	76.7	72.4	70.3	77.5	64.3
	COFT_w	69.7	69.9	69.5	74.3	72.3	76.5	70.9	75.5	66.9
ChatGPT	COFT_p	77.3	75.4	79.3	86.6	80.8	93.3	85.7	80.3	91.9
	COFT_s	74.4	84.5	66.4	75.4	81.1	70.4	81.3	84.4	78.5
	COFT_w	69.9	70.1	69.8	74.3	83.5	66.9	78.6	85.1	73.0

Table 24. Results of question answering tasks in the Natural Questions benchmark. Performance of the vanilla LLM and COFT is evaluated in terms of EM and F1 score under different positions of the correct document.

Backbone	Methods	1st		2nd		3rd		4th		5th	
		EM	F1 Score	EM	F1 Score	EM	F1 Score	EM	F1 Score	EM	F1 Score
Vicuna-33B	COFT	21.5	32.3	20.9	32.0	19.5	30.6	20.5	31.7	21.4	32.0
	Vanilla	18.3	26.3	16.3	23.6	11.7	18.6	13.3	24.5	16.2	25.8
ChatGPT	COFT	35.6	53.8	35.3	51.2	31.0	50.3	32.7	51.4	35.2	52.8
	Vanilla	28.1	40.1	26.3	37.3	20.5	32.1	23.3	33.4	26.9	38.1
GPT-4	COFT	58.5	66.8	57.7	66.5	56.9	64.8	57.5	65.9	57.9	65.5
	Vanilla	54.5	63.5	53.7	62.7	51.6	58.4	53.6	61.1	53.9	62.8

Table 25. Results of knowledge hallucination task. We denote randomly selecting version of COFT as Random COFT and the original version of COFT as Original COFT.

		WK			**Sci/Tech**			**Wri/Rec**		
Backbone	**Methods**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**
Vicuna-33B	Random COFT_p	59.4	67.3	53.2	50.4	51.1	49.8	55.5	50.1	62.1
	Original COFT_p	69.3	71.9	66.9	67.9	62.9	73.8	70.4	66.8	74.4
	Random COFT_s	49.9	49.7	50.1	50.9	47.9	54.3	55.1	48.9	63.1
	Original COFT_s	62.0	63.1	60.9	68.7	67.1	70.4	66.2	64.7	67.7
	Random COFT_w	35.8	33.1	38.9	37.7	29.3	52.9	45.1	34.9	63.7
	Original COFT_w	64.4	61.7	67.4	70.9	65.7	77.2	77.3	67.9	89.8
ChatGPT	Random COFT_p	49.5	53.8	45.9	45.8	55.4	39.0	51.0	59.2	44.8
	Original COFT_p	78.6	83.8	74.0	83.9	81.2	86.8	77.5	85.9	70.5
	Random COFT_s	37.3	48.1	30.4	46.4	45.5	47.3	39.6	48.1	33.6
	Original COFT_s	76.8	75.7	77.9	74.6	79.1	70.5	76.8	84.4	70.5
	Random COFT_w	52.6	53.6	51.7	57.4	49.3	68.7	47.0	55.3	40.8
	Original COFT_w	81.6	85.5	77.9	84.4	80.9	88.4	81.1	93.7	71.5
GPT-4	Random COFT_p	64.1	59.1	70.0	66.0	60.1	73.3	65.0	78.9	55.3
	Original COFT_p	83.1	79.7	86.8	89.9	84.4	96.1	91.8	85.5	99.1
	Random COFT_s	66.0	75.0	58.9	61.6	68.1	56.3	66.9	75.4	60.1
	Original COFT_s	80.0	92.3	70.6	76.6	84.9	69.8	85.5	89.2	82.1
	Random COFT_w	62.4	80.5	51.0	71.6	77.1	66.9	68.8	78.4	61.3
	Original COFT_w	87.3	94.8	80.9	77.9	86.0	71.3	84.7	92.9	77.9

Table 26. Results of knowledge hallucination benchmark. We denote the joint-level version of COFT as COFT-joint and the original version of COFT as Original COFT. The best results for each LLM backbone are highlighted in **bold**.

		WK			**Sci/Tech**			**Wri/Rec**		
Backbone	**Methods**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**
Vicuna-33B	Original COFT_p	69.3	71.9	66.9	67.9	62.9	73.8	70.4	66.8	74.4
	Original COFT_s	62.0	63.1	60.9	68.7	67.1	70.4	66.2	64.7	67.7
	Original COFT_w	64.4	61.7	67.4	70.9	65.7	77.2	77.3	67.9	89.8
	COFT-joint	71.2	73.3	69.2	70.7	66.8	75.1	79.1	69.8	91.2
ChatGPT	Original COFT_p	78.6	83.8	74.0	83.9	81.2	86.8	77.5	85.9	70.5
	Original COFT_s	76.8	75.7	77.9	74.6	79.1	70.5	76.8	84.4	70.5
	Original COFT_w	81.6	85.5	77.9	84.4	80.9	88.4	81.1	93.7	71.5
	COFT-joint	81.1	87.1	75.9	86.2	83.5	89.0	84.8	92.5	78.3
GPT4	Original COFT_p	83.1	79.7	86.8	89.9	84.4	96.1	91.8	85.5	99.1
	Original COFT_s	80.0	92.3	70.6	76.6	84.9	69.8	85.5	89.2	82.1
	Original COFT_w	**87.3**	94.8	80.9	77.9	86.0	71.3	84.7	**92.9**	77.9
	COFT-joint	89.7	95.0	85.0	90.0	88.0	92.0	92.1	90.7	93.5

J. More Discussions On COFT

J.1. Why is it necessary to use both TF-ISF and self-information to measure the importance of entities?

We propose *context weights* to assess the importance of each entity within a given context. Firstly, we consider the frequency and distribution of entities in the reference context by calculating the **T**erm **F**requency-**I**nverse **S**entence **F**requency (TF-ISF), which helps to distinguish entities that are frequently mentioned yet not common words. Such entities play a significant role in understanding the semantics of sentences. Subsequently, we take into account the amount of information an entity contributes when responding to a query within the reference context by computing self-information, which helps to identify potentially significant, highly informative entities. These entities may carry unique or crucial information essential for understanding the entire text.

TF-ISF measures the distinctiveness of words across sentences and self-information assesses their information value in probabilistic terms. By multiplying these two metrics, we obtain a comprehensive indicator, *context weights*, that more effectively captures the importance of words in context. Moreover, the *context weights* may potentially be beneficial to consider additional factors or optimize the combination method of TF-ISF score and self-information to get improved results and we leave the exploration as a future work.

J.2. Why does COFT only search for one-hop neighbor entities in the open-source KG?

Retrieving neighbor entities of a target entity in a knowledge graph is a common method for finding entities related to the target. For example, the one-hop neighbor entities of a celebrity often represent attributes like their family, friends, nationality, and workplace, which contain significant and comprehensive information about the given entity. We could also opt to retrieve the two-hop neighbor entities. If the average degree of nodes in the KG is high, this approach can introduce more relevant nodes but may also bring in a large number of irrelevant nodes.

Therefore, the decision to search for one-hop, two-hop, or more distant neighbor entities should be dynamically adjusted based on the task and the specific characteristics of the KG. For COFT, we observe that retrieving only one-hop neighbors yields satisfactory results. Considering the potential noise introduced by retrieving higher-hop neighbors, we only retrieve one-hop neighbors to enrich the key entity candidates for COFT in all mentioned tasks.

J.3. Why highlighting key lexical units in three different granularity levels from coarse to fine?

COFT represents a key entity-driven highlighting approach. It captures potential key entities from the perspective of a knowledge graph (world knowledge) and evaluates the importance of entities based on the TF-ISF and self-information scores. After identifying the final key entities, a straightforward method is to highlight these corresponding entities in the reference context, i.e., the word-level highlighting COFT. However, considering practical applications, such as cases where the reference context contains a small number of entities that appear multiple times, word-level highlighting provides limited information as well. In these scenarios, sentence-level highlighting or paragraph-level highlighting may be more appropriate.

Moreover, for certain queries, it is crucial to focus on the core paragraphs of the reference text, rather than just sentences or words. Therefore, to enhance the versatility of COFT, we have proposed three different levels of granularity of highlighting selections: paragraph, sentence, and word. Table 1 corroborates the effectiveness of multi-granular highlighting as well.

J.4. What named entity recognition method is employed in COFT, and does it have any tailored designs?

Given the relative maturity of named entity recognition (NER) in the fields of natural language processing and knowledge graphs, we do not elaborate on it in the main text. Considering the need for rapid deployment and ease of implementation, we have utilized the Spacy[11] open-source library for the NER component. Moreover, we employ noun phrase extraction from the NLTK library[12] to retain some non-named yet significant nouns in queries. We also reference the entity list from Wikidata for entity recognition. The ablation study results in Tables 3, 8, and 9, demonstrate the simplicity and effectiveness of the NER component in our method. We also think that other specific optimized NERs are promising to improve COFT.

[11] https://spacy.io/
[12] https://www.nltk.org/