Say More with Less: Understanding Prompt Learning Behaviors through Gist Compression

Anonymous ACL submission

Abstract

Large language models (LLMs) require lengthy prompts as the input context to produce output aligned with user intentions, a process that incurs extra costs during inference. In this paper, we propose the **Gist CO**nditioned de**CO**ding (Gist-COCO) model, introducing a novel method for compressing prompts which also can assist the prompt interpretation and engineering. Gist-COCO employs an encoder-decoder based language model and then incorporates an additional encoder as a plugin module to compress prompts with inputs using gist tokens. It finetunes the compression plugin module and uses the representations of gist tokens to emulate the raw prompts in the vanilla language model. By verbalizing the representations of gist tokens into gist prompts, the compression ability of Gist-COCO can be generalized to different LLMs with high compression rates. Our experiments demonstrate that Gist-COCO outperforms previous prompt compression models in both passage and instruction compression tasks. Further analysis on gist verbalization results suggests that our gist prompts serve different functions in aiding language models. They may directly provide potential answers, generate the chain-of-thought, or simply repeat the inputs. All codes will be released via GitHub.

1 Introduction

Large Language Models (LLMs), such as GPT-4 (Achiam et al., 2023) and LLaMA (Touvron et al., 2023), have demonstrated their emergent capacity in handling various NLP tasks (Zhao et al., 2023; Wei et al., 2022b). To align user intentions with LLMs, existing work pays increasing attention to prompt engineering. They attempt to optimize prompts using LLMs themselves or manually craft prompts with meticulous care (Zhou et al., 2022; Cheng et al., 2023; Ye et al., 2023). Nevertheless, the challenge persists in deciphering user intentions from natural language by LLMs and providing explainable insights for prompt engineering.

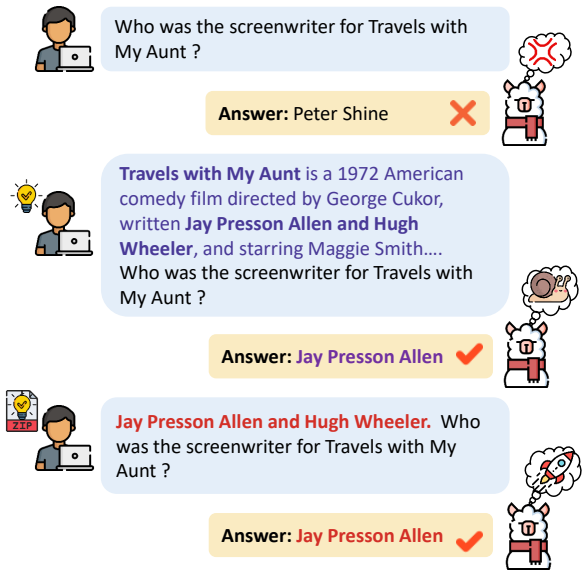


Figure 1: The Motivation of Our Gist Conditioned Decoding (Gist-COCO) Model. The user respectively utilizes prompts and compressed prompts to guide the generation of LLMs.

As shown in Figure 1, users typically collect or compose detailed prompts to assist LLMs in generating answers, making them more tailored and precise. However, with each user query, LLMs must iteratively encode these prompts and compute their self-attention (Vaswani et al., 2017), leading to increased computational time and memory usage (Mu et al., 2023). Reducing the length of prompts is a potent strategy to optimize these prompts. Existing work utilizes the theory of self-information (Shannon, 1948) to explain prompts and reduce them by filtering the contexts with low self-information in the prompts (Li, 2023). Mu et al. (2023) further compress task instructions by utilizing gist tokens and employing the resulting gist embeddings for instruction representation. Nevertheless, achieving interpretability and refinement in

prompt compression, which is crucial for prompt engineering and understanding LLMs' behavior, remains challenging yet.

To alleviate the problem, this paper introduces the **Gist COnditioned deCOding** (Gist-COCO) model, which targets on compressing prompts and generalizing compression to different LLMs. Our Gist-COCO model is inspired by information theory (Grünwald, 2007) and built upon an encoder-decoder based language model, such as FlanT5 (Chung et al., 2022). It employs an extra encoder model as a compression plugin module to compress prompts with inputs using a set of shorter gist tokens whose representations are utilized to replace the raw prompts of inputs. Specifically, these gist representations are contacted as prefixes with the input representations encoded by the vanilla encoder and fed into the vanilla decoder. Gist-COCO only finetunes the compression model to generate more effective gist representations, aiding the vanilla FlanT5 model in adhering closely to the raw prompts for the generation. Additionally, our Gist-COCO model incorporates a task disentangled gist modeling method to effectively compress various types of prompts, such as passages and instructions.

To generalize the compression capabilities of Gist-COCO across different LLMs, we propose the gist verbalization method, which can verbalize gist representations into some shorter gist prompts using the language model. By preprocessing the prompts with inputs using the compression module, the gist prompts refine the essential information from the raw prompts based on the inputs. Instead of using annotated summarization data to learn prompt compression (Vig et al., 2022; Xu et al., 2023), compression models, such as Gist (Mu et al., 2023) and Gist-COCO, compress prompts using gist tokens and optimize these gist representations using vanilla prompts from training data. Additionally, unlike baseline models (Mu et al., 2023; Chevalier et al., 2023), our Gist-COCO model freezes the parameters of language models and only finetunes the encoder model for compression, which can generalize its compression ability.

Our experiments demonstrate the effectiveness of the Gist-COCO model, surpassing prior prompt compression models in both passage and instruction compression tasks. Leveraging our gist verbalization method, Gist-COCO broadens its advantages to different language models, achieving an exceptionally high compression rate. Besides, the results of gist verbalization show that gist prompts serve diverse roles in assisting language models to comprehend human instructions, such as encompassing the formation of answers, generating the thought, and copying parts of contents from inputs or instructions for reinforcement.

2 Related Work

Large Language Models (LLMs) (Brown et al., 2020), typically finetune through instruction learning methods (Chung et al., 2022; OpenAI, 2022; Taori et al., 2023; Chiang et al., 2023), such as instruction tuning or Reinforcement Learning with Human Feedback (RLHF) (Ouyang et al., 2022), can enhance their ability to adhere to instructions or align with human preferences. Besides, finetuning language models on diverse instruction-response pairs enables language models to exhibit cross-task generalization (Wei et al., 2022a; Sanh et al., 2021). In this case, existing work focuses more on generating more instruction data (Wang et al., 2023; Wan et al., 2023; Mishra et al., 2022) or the task sensitive tasks (Kung et al., 2023) for supervised finetuning (SFT) LLMs.

To enhance the effectiveness of LLMs in downstream tasks, researchers are increasingly emphasizing prompt engineering (Liu et al., 2023). The prompts can serve as instructions to elucidate user intentions (Zhou et al., 2022) or provide the contextual knowledge to aid in the generation process (Izacard et al., 2023; Ram et al., 2023; Tonmoy et al., 2024; Shi et al., 2023). However, the prompts have demonstrated that they potentially exert a substantial influence on the LLMs' outputs (Lu et al., 2022) and necessitate meticulous designs (Chen et al., 2023; Kaddour et al., 2023).

To make prompts better guide the generation of LLMs, existing work focuses more on conducting more effective prompts in different ways. Zhou et al. (2022) use LLMs for automatic instruction generation and selection. Cheng et al. (2023) propose the Black-box Prompt Optimization (BPO) method, which optimizes the prompts to bridge the gap between humans and LLMs. Ye et al. (2023) further add the task-agnostic prefix to enhance the instruction. Nevertheless, it remains unclear which aspects of these provided prompts are favored by LLMs for comprehending human intentions.

Studying the characteristics of prompts in prompting LLMs has garnered much attention from researchers (Min et al., 2022; Beurer-Kellner et al., 2023). The researchers use the Turking Test (Efrat

and Levy, 2020) and the negated prompts (Jang et al., 2023) to analyze the instruction understanding and following ability of LLMs. Instead of evaluating such an ability of LLMs, inspired by the minimum description length (MDL) principle (Grünwald, 2007), we focus more on interpreting the role of prompts from a compression view. Some existing work has shown effectiveness in prompt compression, *e.g.* distilling the prompt understandings from teacher models to student models (Snell et al., 2022), compressing the prompts using a set of gist tokens (Mu et al., 2023; Ge et al., 2023; Chevalier et al., 2023) and generating some brief summaries (Vig et al., 2022; Xu et al., 2023). Based on these works, we aim to compress prompts as gist representations according to the need of language models and further verbalize them into gist prompts to interpret and understand the role of prompts.

3 Methodology

In this section, we first introduce prompt compression through the information theory (Sec. 3.1). We then describe our **Gist COnditioned deCOding** (Gist-COCO) model (Sec. 3.2). Finally, we show how to generalize the compression ability to different tasks and language models (Sec. 3.3).

3.1 Preliminary of Prompt Compression

Given an input x, existing work usually uses lengthy task instructions (Wang et al., 2023; Chung et al., 2022) or retrieved passages (Yu et al., 2023b; Shi et al., 2023) as prompts, denoted as c, to aid LLMs for the generation. To reduce inference cost, Gist-COCO compresses the raw long prompt c into a few gist representations $h^c = \{h^c_1, ..., h^c_N\}$, serving as condensed context for LLM inference.

Inspired by the compression viewpoint of the minimum description length (MDL) principle (Grünwald, 2007) in information theory (Wu et al., 2023), a good model should be able to represent the data with shorter descriptions and also generalize well to unseen data (Wu et al., 2023). The MDL principle indicates that the best compression model M^* can make the correct prediction y^* based on a shorter codelength:

$$M^*_\theta = \arg\min_\theta L(M_\theta) + L(y^*|M_\theta(c, x)), \quad (1)$$

where $L(M_\theta)$ is the codelength (model complexity) required by the model and $L(y^*|M_\theta(c, x))$ is the codelength to construct the correct prediction based on the compression result. The compression model

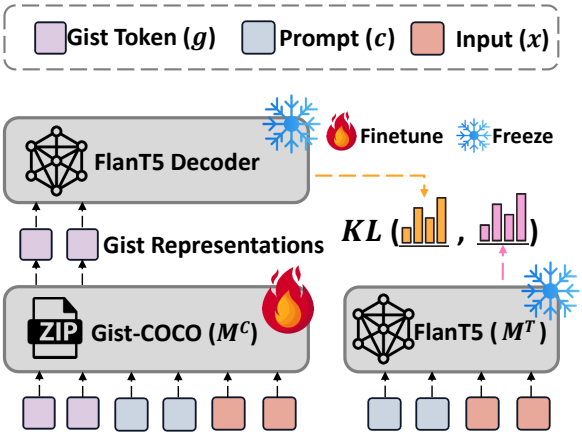


Figure 2: Training of Gist-COCO. Gist-COCO is trained to emulate the output distribution based on uncompressed inputs by producing gist representations.

M_θ encodes the prompt c into a fixed number of hidden states h^c, given c with the input x:

$$h^c = \{h^c_1, ..., h^c_N\} \leftarrow M_\theta(c, x). \quad (2)$$

As we fix $|h_c| = N$, the term $L(M_\theta)$ becomes constant in Eq. 1 and our goal is to minimize $L(y^*|M_\theta(c, x))$. In the next section, we introduce M_θ as well as its training and inference.

3.2 Prompt Compression via Gist Conditioned Decoding

Given the prompt c and input x, Gist-COCO is trained to minimize Eq. 1 to produce the optimal gist representations $h^c = \{h^c_1, ..., h^c_N\}$ for the prompt c. As shown in Figure 2, we propose to leverage the soft labels from a vanilla language model M^T with raw prompts to estimate the codelength with the help of Kullback-Leibler (KL) divergence between the uncompressed distribution and the compressed one:

$$L(y^*|M_\theta(c, x)) \approx \text{KL}(P(y^*|h^c, x)||Q(y^*|c, x)), \quad (3)$$

where $P(y^*|h^c, x)$ is the generation probability given the gist representations calculated from FlanT5-Decoder, and $Q(y^*|c, x)$ is the prior from the model given raw prompts, calculated from M^T (FlanT5):

$$\begin{aligned} P(y^*|h^c, x) &= \text{T5-Decoder}(h^c), \\ Q(y^*|c, x) &= M^T(c; x), \end{aligned} \quad (4)$$

where ; denotes concatenation. The parameters of M^T are frozen during training. h_c is encoded by the compression model M^C_θ, which is initialized with the same parameters as the model M^T:

$$h^c \leftarrow M^C_\theta(c, x) = \text{T5-Encoder}(\{g_1, ..., g_N\}; c; x), \quad (5)$$

where $\{g_1, ..., g_N\}$ are the gist tokens to compress the prompt c, whose weights are initialized from the special tokens of the FlanT5 model. h^c are the encoded representations of $\{g_1, ..., g_N\}$ using M^C.

During inference, following Eq. 5, we use the trained compression model M_θ^C to compress the prompt to obtain gist representations h^c, and feed them with the encoded input x into the decoder to obtain the output:

$$y = \text{T5-Decoder}(h^c; \text{T5-Encoder}(x)). \quad (6)$$

3.3 Compression Generalization for Different Prompts and Language Models

In this subsection, we generalize Gist-COCO to different tasks and language models by task disentangled gist modeling and prompt verbalization.

Task Disentangled Gist Modeling. We compress two types of prompts during modeling, including retrieved passages (Guu et al., 2020) and instructions (Mu et al., 2023), which are typically used in existing language models.

For instruction compression, we regard the task instruction as the prompt c and then use N instruction gist tokens $\{g_1^i, ..., g_N^i\}$ for compression:

$$h^c \leftarrow M^C(\{g_1^i, ..., g_N^i\}; c; x), \quad (7)$$

where $h^c = h^c(g^i)$. $h^c(g^i)$ represents the set of encoded representations of $\{g_1^i, ..., g_N^i\}$. In the retrieval-augmented generation (RAG) models, we regard the concatenation of retrieved passages and task instructions as the prompt c. Then we use both N passage gist tokens $\{g_1^p, ..., g_N^p\}$ and N instruction gist tokens $\{g_1^i, ..., g_N^i\}$ for compression:

$$h^c \leftarrow M^C(\{g_1^p, ..., g_N^p\}; \{g_1^i, ..., g_N^i\}; c; x), \quad (8)$$

where $h^c = \{h^c(g^p); h^c(g^i)\}$. $h^c(g^p)$ and $h^c(g^i)$ are the compressed representations of the passage gist tokens g^p and the instruction gist tokens g^i.

Gist Verbalization. To generalize the advantages of our Gist-COCO model to decoder-based language models, we use the vanilla FlanT5 decoder to decode the compressed hidden states h^c to get the gist prompts $v = \{v_1, ..., v_k\}$:

$$v = \text{T5-Decoder}(h^c). \quad (9)$$

We can assess the compression effectiveness of our Gist-COCO model by replacing the prompt c with the shorter gist prompts v when utilizing decoder-based language models. Besides, we can further observe and understand the effectiveness of prompt learning by analyzing the gist prompts v.

Split	Dataset	Setting	Total
Training	NVI2	Instruction	94,481
		Passage	92,607
Evaluation	PopQA	-	14,267
	KILT	NQ	2,837
		TrivialQA	5,359
		HotpotQA	5,600
	Alpaca+	Seen	1,000
		Unseen	1,000
		Human	252

Table 1: Data Statistics.

4 Experimental Methodology

This section describes the datasets, evaluation metrics, baselines, and implementation details.

Dataset. In our experiments, we use different datasets to build the training and evaluation benchmarks. All data statistics are shown in Table 1.

Training. During training Gist-COCO model, we use Natural Instruction v2 (NVI2) (Wang et al., 2022) dataset to build the training set for compression. The training dataset consists of instruction compression and retrieved passage compression tasks. For instruction compression, we filter out the non-English tasks and reserve 1,053 tasks. We randomly sample up to a maximum of 90 instances from each task, resulting in a total of 94,481 pieces of data. For the retrieved passage compression, we selected 30 tasks from NVI2 dataset, amounting to a total of 92,607 pieces of data. These selected tasks usually require external knowledge and we use T5-ANCE (Yu et al., 2023a,b) to retrieve passages from MS MARCO (Nguyen et al., 2016) for augmenting the language model.

Evaluation. During evaluation, we use different datasets to estimate the effectiveness of retrieved passage compression and instruction compression.

Following previous work (Mu et al., 2023), we use Alpaca+ dataset (Mu et al., 2023) to evaluate the instruction compression effectiveness of Gist-COCO. The Alpaca+ dataset is a large instruction finetuning dataset, which combines both Self-Instruct (Wang et al., 2023) and Stanford Alpaca (Taori et al., 2023) datasets. To evaluate the effectiveness of retrieved passage compression, we use PopQA (Mallen et al., 2023) as well as NQ (Kwiatkowski et al., 2019), TrivialQA (Joshi et al., 2017) and HotpotQA (Yang et al., 2018) from KILT (Petroni et al., 2021) for evaluation, where we use the dev set for all tasks from KILT. The KILT-Wikipedia (Petroni et al., 2021) is regarded as the knowledge base for seeking knowledge. Then we

LLM	Method	Passage Compression				Instruction Compression		
		PopQA	KILT			Alpaca+		
			NQ	TriviaQA	HotpotQA	Seen	Unseen	Human
FlanT5-base	No Prompt	8.8	4.4	9.2	12.1	20.3	22.0	9.7
	AutoCompressor (2023)	8.3	4.8	9.4	12.2	20.3	22.7	7.8
	Gist (2023)	8.4	4.6	8.9	12.1	18.3	18.9	8.7
	Gist (Ours) (2023)	9.4	5.7	11.4	11.6	23.1	27.5	**14.1**
	Gist-COCO	**31.0**	**22.9**	**50.9**	**17.2**	**23.6**	**29.0**	12.1
	Full Prompt	43.9	30.0	61.9	23.2	23.9	29.8	15.2
FlanT5-large	No Prompt	7.3	8.3	19.0	14.6	19.2	18.4	10.3
	AutoCompressor (2023)	5.8	8.4	19.1	14.7	16.6	12.2	6.6
	Gist (2023)	9.1	8.2	18.9	14.6	21.4	19.0	10.7
	Gist (Ours) (2023)	11.6	8.4	19.1	13.0	24.3	29.4	**15.7**
	Gist-COCO	**32.0**	**27.0**	**57.3**	**20.6**	**25.7**	**30.1**	14.0
	Full Prompt	46.0	34.4	67.1	27.5	26.7	32.3	18.8

Table 2: Overall Performance of Different Prompt Compression Methods.

use T5-ANCE (Yu et al., 2023a,b) to retrieve passages from it for augmentation.

Baselines. In our experiment, we compare our Gist-COCO model with several baselines.

Two embedding based compression models are compared in our experiments, including AutoCompressor (Chevalier et al., 2023) and Gist (Mu et al., 2023). We directly use the AutoCompressor and Gist models to compress the prompts as representations and then train a linear layer to adapt the compressed representations to the FlanT5 model. AutoCompressor is an unsupervised model, which compresses long contexts into a set of summary vectors to facilitate different generation tasks. Different from AutoCompressor, Gist (Mu et al., 2023) is a supervised method, which finetunes the language models on the Alpaca+ instruction dataset and teaches the model to compress the instructions through the attention mask.

Besides, we also reimplement the Gist model, denoted as Gist (Ours), maintaining identical model architecture with Mu et al. (2023). We finetune this model using the same training dataset employed for our Gist-COCO model. Furthermore, we utilize the SEGENC model (Vig et al., 2022) as a baseline, which finetunes BART (Lewis et al., 2020) model using the query-focused summarization dataset.

Evaluation Metrics. Following previous work (Mu et al., 2023), we used the ROUGE-L metric to evaluate the performance of different models on instruction compression tasks. For the passage compression tasks, we use accuracy as an evaluation metric, which is similar to Yu et al. (2023b). We conduct string matching between the generated answer and the golden answer.

Experimental Details. This part describes the experiment details of Gist-COCO model.

We initialize Gist-COCO model with FlanT5-base and FlanT5-large checkpoints from Huggingface Transformers (Wolf et al., 2019). During training, we use the top-1 ranked passage from retrieval as the prompt to enhance the generation results for these passage compression tasks. In our experiments, we set the learning rate as 1e-4 and the training epoch as 8. During inference, we use the top-5 ranked passages from retrieval as the prompt for all passage compression tasks.

5 Evaluation Results

In this section, we first evaluate the performance of Gist-COCO on passage and instruction compression tasks. Subsequently, we conduct ablation studies and further analyze the characteristics of learned gist representations. Finally, the case studies are presented.

5.1 Overall Performance

The experiments show the effectiveness of Gist-COCO in the tasks of passage compression and instruction compression, utilizing both encoder-decoder-based language models and decoder-based language models for evaluation.

The representation-based prompt compression performance is shown in Table 2. In our experiments, we implement the Gist and AutoCompressor models by training a linear layer to adapt the compressed representations to the FlanT5 model. When compared to the fully finetuned compression model, Gist (Ours), they demonstrate comparatively less effectiveness in assisting FlanT5 to comprehend the knowledge and user intent conveyed through the prompts. This suggests that representation-based compression models still require finetuning to tailor them to different language models, limiting the

LLM	Method	Passage Compression					Instruction Compression			
		PopQA	KILT			Ratio	Alpaca+			Ratio
			NQ	TrivialQA	HotpotQA		Seen	Unseen	Human	
Llama-7b	No Prompt	22.8	20.5	61.7	18.2	-	21.7	21.1	4.6	-
	SEGENC (2022)	25.9	24.6	63.9	19.9	97.6%	**25.6**	25.0	8.1	22.9%
	Gist-COCO	**34.9**	**28.9**	**69.6**	**22.6**	99.1%	24.7	**25.3**	**8.8**	35.9%
	Full Prompt	43.3	33.5	75.1	25.4	-	36.0	34.4	12.5	-
Llama2-7b	No Prompt	26.0	24.0	67.8	20.9	-	21.3	20.8	6.2	-
	SEGENC (2022)	29.9	29.2	70.6	21.8	97.6%	**26.1**	24.7	**8.6**	22.9%
	Gist-COCO	**35.9**	**30.8**	**71.9**	**24.4**	99.1%	22.7	**24.9**	8.4	35.9%
	Full Prompt	45.2	35.0	75.4	27.9	-	35.5	32.8	12.3	-
Llama-13b	No Prompt	27.6	27.2	72.9	22.0	-	22.3	18.7	4.0	-
	SEGENC (2022)	31.3	29.2	70.5	22.7	97.6%	**27.7**	**26.2**	9.5	22.9%
	Gist-COCO	**36.9**	**30.8**	**74.5**	**24.4**	99.1%	24.8	26.0	**9.6**	35.9%
	Full Prompt	45.7	36.0	77.6	29.3	-	37.6	38.0	14.4	-

Table 3: Effectiveness of Prompt Compression on Decoder-based Language Models.

generalization ability of these baseline models.

The evaluation results show that Gist-COCO outperforms all compression baseline models, demonstrating its ability to learn more tailored gist representations for prompt compression. Notably, Gist-COCO achieves more than a 20% improvement on the passage compression task, showing its effectiveness in distilling some necessary information from the raw prompts to the gist representations. Different from the baseline models, such as Gist (Ours), Gist-COCO freezes the parameters of language models and only finetunes an additional encoder model specifically for prompt compression, which helps to preserve the capabilities of vanilla language models. It breaks the limitation of compression generalization by directly using the decoder module of vanilla language models to verbalize the gist representations into gist prompts for aiding different LLMs.

We then extend the evaluation of Gist-COCO's compression efficacy to decoder-based language models by using gist prompts (Eq. 9) to replace raw prompts. The evaluation results are shown in Table 3. Overall, Gist-COCO enhances the generation accuracy of Llama-7b/13b by furnishing compressed prompts, demonstrating their ability to extract essential information from raw prompts. In comparison to the query-focused passage compression model, SEGENC, Gist-COCO achieves competitive or even superior performance in both passage and instruction compression tasks. This highlights the capacity of leveraging the language model itself for prompt compression and selecting informative contents in an unsupervised manner.

5.2 Ablation Studies

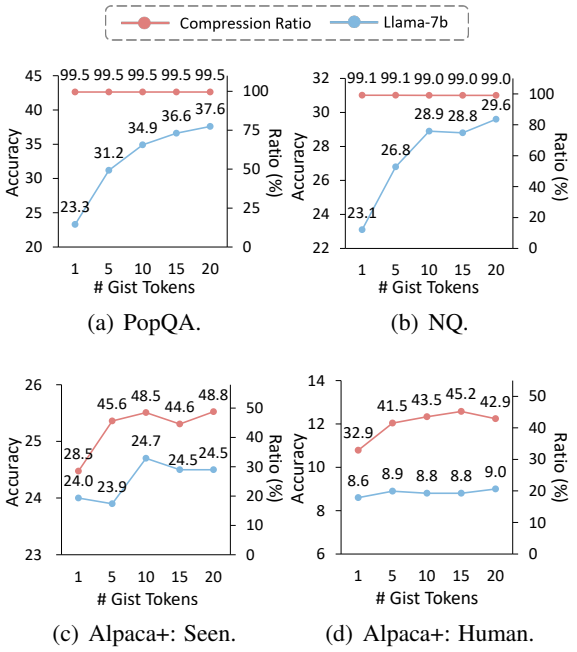
This experiment conducts ablation studies to demonstrate the effectiveness of Gist-COCO with varying numbers of gist tokens and explores the impact of employing unified gist tokens. More ablation studies are shown in Appendix A.2.

Setting	#Token	PopQA	KILT	Alpaca+
Unified	5	24.3	30.3	24.7
	10	26.6	32.1	24.6
	20	**30.9**	**35.8**	**26.5**
Gist-COCO (Disentangled)	1	16.8	24.2	19.2
	5	27.4	33.1	24.7
	10	32.0	36.2	26.3
	15	34.5	37.2	**26.8**
	20	**35.8**	**38.0**	26.7

Table 4: Ablation Studies. We employ varying numbers of gist tokens to encode prompts as hidden states and feed them to FlanT5-large for evaluating the compression effectiveness.

As shown in Table 4, we conduct the Unified and Gist-COCO (Disentangled) settings to train the model to compress the prompts into gist tokens, separately. In the unified setting, we utilize all gist tokens to compress both passages and instructions. Our Gist-COCO model uses disentangled gist tokens that are allocated in equal numbers for compressing passages and instructions. For example, the number of gist tokens in the decomposition setting is 5 signifies that we use 5 gist tokens to compress passages and another 5 gist tokens to compress instructions.

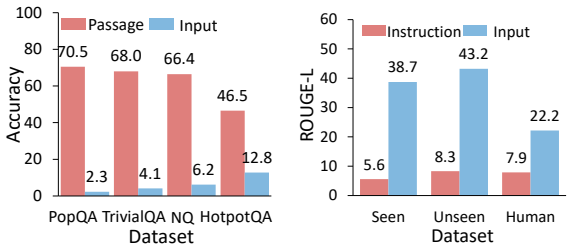
The evaluation results show that, disentangling the gist tokens for various compression tasks typically leads to improvements, highlighting the necessity of utilizing distinct gist tokens to represent various tasks. As the number of gist tokens increases, the compression performance strengthens accordingly. This indicates that additional gist tokens can capture and convey more information from prompts and inputs, thereby enhancing the language model generation process. However, it's noteworthy that



(a) PopQA. (b) NQ.

(c) Alpaca+: Seen. (d) Alpaca+: Human.

Figure 3: Effectiveness of Gist Verbalization Results. We use different numbers of compression tokens.



(a) Passage Compression. (b) Instruction Compression.

Figure 4: Text Similarity between the Gist Verbalization Results with Inputs and Prompts.

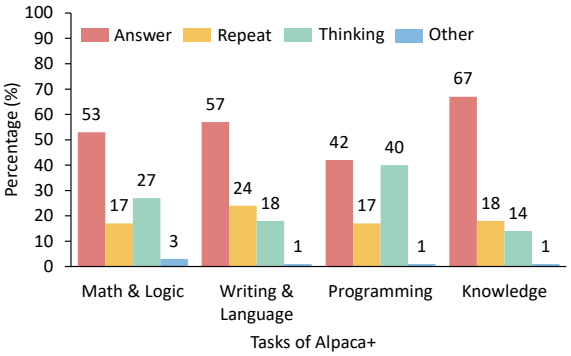


Figure 5: Distribution of Categorizations of Gist Verbalization Results. We categorize Alpaca+ tasks into distinct groups and present the categorization outcomes of verbalization results across various tasks.

the performance improvement tends to plateau after reaching a gist token count of 10. Consequently, we opt for 10 as the optimal gist token count for compressing both passages and instructions.

Then we show the effectiveness of the verbalization outputs produced by Gist-COCO, as depicted in Figure 3, utilizing the Llama-7b model. Evaluation results indicate that the verbalized outputs from Gist-COCO consistently enhance the performance of Llama-7b as the number of gist tokens increases. Conversely, performance remains almost unchanged across instruction compression tasks. This illustrates that passages typically encompass more compressible information, while 10 gist tokens are adequate for instruction compression. Moreover, the compression ratio remains stable across different numbers of gist tokens, indicating that prompts are typically treated as short prefixes for language models, and certain tokens play a more crucial role in aiding language models.

5.3 Characteristics of Learned Gist Representations

In this experiment, by verbalizing these gist representations into gist prompts, we further analyze the knowledge learned by gist tokens.

As shown in Figure 4, we first evaluate the text similarity between the gist prompts and both inputs and prompts. Regarding the passage compression tasks, the gist prompts exhibit a notably high resemblance to the passages rather than the inputs. This observation underscores that the primary objective of passage compression is to extract essential knowledge from the passage to facilitate question answering. In contrast, for the tasks in Alpaca+, the gist prompts demonstrate much higher similarity to the inputs. This suggests that our Gist-COCO model engages in a more profound analysis of the queries using the provided instructions.

Then we explore the roles of gist prompts across various tasks in Figure 5. We firstly employ GPT-3.5 to categorize the data within the Alpaca+ dataset into four distinct groups: Match & Logic, Writing & Language, Programming, and Knowledge. Detailed categorization statistical information is shown in Appendix A.3. Subsequently, we randomly select 100 instances from each task group and assign labels to the sampled data using GPT-3.5. These labels include Answer, Repeat, Thinking, and Other. The "Answer" label denotes that the gist prompts provide potential answers to the input. The "Thinking" label signifies that the gist prompts serve as a form of thought process. Meanwhile, the "Repeat" label indicates that the gist prompts

Passage Compression	
PopQA	**Passage:** Page 3 (film) Page 3 is a 2005 Indian drama film directed by **Madhur Bhandarkar** and produced by Bobby Pushkarna and Kavita Pushkarna about the Page 3 culture and media in the city of Mumbai. It stars Konkona Sen Sharma, Atul Kulkarni, Sandhya Mridul, Tara Sharma, Anju Mahendru, and Boman Irani. The film won three National Film Awards ... **Input:** Who was the director of Page 3? **Compression:** **Madhur Bhandarkar**
TriviaQA	**Passage:**...The screenplay by Robert E. Sherwood and Joan Harrison, and adaptation by Philip MacDonald and Michael Hogan, were based on the 1938 novel of the same name by **Daphne du Maurier**. The film stars Laurence Olivier as the brooding... **Input:** The Alfred Hitchcock films Rebecca and The Birds were based on novels by which author? **Compression:** **Daphne du Maurier**
Instruction Compression	
Knowledge	**Instruction:** Classify this sentence into one of the topics: education, politics, **technology**, sports **Input:** Apple's new Iphone was released today. **Compression:** **technology**
Match & Logic	**Instruction:** What is the best way to get from point a to point b? explain why you chose that method. **Input:** Point A: (0, 0) and Point B: (10, 10) **Compression:** The first step is to get to the point where you want to go.
Programming	**Instruction:** You are given a programming problem and its implementation. Analyze the problem and implementation and explain the algorithm and approach used to solve the problem. **Input:** Table: Person \n| Column Name | Type |\n | personId | int |\n | lastName | varchar |\ n | firstName | varchar |... **Compression:** SELECT T1.name FROM Person AS T1 JOIN Address AS T2 ON T1.name = T2.name JOIN Person AS T
Writing & Language	**Instruction:** The topic of YouTube post has been described and based on the information, you need to write a hook for starting the post. A catchy hook will keep readers interested so they keep reading. **Input:** **A video showing how to make a tasty** cup of **coffee.** **Compression:** **A video showing how to make a tasty coffee.**

Table 5: Case Studies. The matched text phrases are **highlighted**.

reiterate the content of queries or instructions.

The evaluation results indicate that directly generating answers is the predominant behavior across different tasks. It demonstrates that compression models usually serve as a form of information preprocessing to give the answer-like results to aid language models. Across all tasks, Gist-COCO tends to repeat prompts or inputs more frequently in the Writing & Language tasks, underscoring the significance of user intent in the task. Moreover, there is a preference for generating a chain of thought to aid Match & Logic and Programming tasks, highlighting the critical role of the thought process in dealing with these tasks (Wei et al., 2022c; Li et al., 2023; Huang et al., 2023).

5.4 Case Studies

Finally, we show several cases in Table 5 to analyze the gist prompts of Gist-COCO.

In the first two cases, the gist prompts like "Madhur Bhandarkar" and "Daphne du Maurier" indicate that the extracted segments from the passage can directly answer the question. It demonstrates the compression module's tendency to directly generate answers for simpler questions, highlighting its preprocessing capabilities. For the third and fourth cases, involving mathematical and programming tasks, strategic planning and critical thinking are necessary. Gist-COCO shows its effectiveness in generating preliminary thoughts or code snippets as prompts to assist language models in comprehending and solving such problems. It confirms that the chain-of-thought and program thought indeed have the ability to improve the model's effectiveness on these tasks. The final case illustrates a writing and language task, where the results indicate Gist-COCO's inclination to replicate the input, suggesting the continued challenge in verbalizing and analyzing such instructions.

6 Conclusion

This paper introduces Gist-COCO, a prompt compression approach utilizing gist conditioned decoding. Our experiments demonstrate that Gist-COCO surpasses existing compression models across various prompt compression tasks and extends its effectiveness to different language models. Further analyses provide some opportunities to understand the prompt behaviors in language models, facilitating a deeper understanding of their functionality.

Limitations

Although Gist-COCO has demonstrated considerable success in compression prompts, it encounters inherent limitations. Existing prompt compression is still difficult to achieve the same results as the original prompt with a high compression ratio, and there are still different degrees of information loss in the prompt compression process. To mitigate this, Gist-COCO attempts to increase the number of gist tokens, but the improvement is limited.

Besides, there are some instructions that are hard to compress, making Gist-COCO repeat the contents in the inputs. In this case, it is still challenging to interpret which contents can really assist the language models to follow the given instruction.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report.

Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. 2023. Prompting is programming: A query language for large language models. *Proceedings of the ACM on Programming Languages*, (PLDI):1946–1969.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*.

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2023. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *ArXiv preprint*.

Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2023. Black-box prompt optimization: Aligning large language models without model training. *ArXiv preprint*.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *Proceedings of EMNLP*, pages 3829–3846.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models.

Avia Efrat and Omer Levy. 2020. The turking test: Can language models understand instructions?

Tao Ge, Jing Hu, Xun Wang, Si-Qing Chen, and Furu Wei. 2023. In-context autoencoder for context compression in a large language model.

Peter D Grünwald. 2007. *The minimum description length principle*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of ICML*, pages 3929–3938.

Dong Huang, Qingwen Bu, and Heming Cui. 2023. Codecot and beyond: Learning to program and test like a developer.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43.

Joel Jang, Seonghyeon Ye, and Minjoon Seo. 2023. Can large language models truly understand prompts? a case study with negated prompts. In *Transfer Learning for Natural Language Processing Workshop*, pages 52–62. PMLR.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of ACL*, pages 1601–1611.

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

Po-Nien Kung, Fan Yin, Di Wu, Kai-Wei Chang, and Nanyun Peng. 2023. Active instruction tuning: Improving cross-task generalization by training on prompt sensitive tasks. In *Proceedings of EMNLP*, pages 1813–1829.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*, pages 7871–7880.

Jia Li, Ge Li, Yongmin Li, and Zhi Jin. 2023. Structured chain-of-thought prompting for code generation.

Yucheng Li. 2023. Unlocking context constraints of llms: Enhancing context efficiency of llms with self-information-based content filtering.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, (9):1–35.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of ACL*, pages 8086–8098.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of ACL*, pages 9802–9822.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of EMNLP*, pages 11048–11064.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of ACL*, pages 3470–3487.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2023. Learning to compress prompts with gist tokens.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset. In *CoCo@ NIPs*.

OpenAI. 2022. Chatgpt.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, pages 27730–27744.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. 2021. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

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

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

Charlie Snell, Dan Klein, and Ruiqi Zhong. 2022. Learning by distilling context.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS*, pages 5998–6008.

Jesse Vig, Alexander R. Fabbri, Wojciech Kryscinski, Chien-Sheng Wu, and Wenhao Liu. 2022. Exploring neural models for query-focused summarization. In *Findings of the Association for Computational Linguistics: NAACL*, pages 1455–1468.

Fanqi Wan, Xinting Huang, Tao Yang, Xiaojun Quan, Wei Bi, and Shuming Shi. 2023. Explore-instruct: Enhancing domain-specific instruction coverage through active exploration. In *Proceedings of EMNLP*, pages 9435–9454.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language

models with self-generated instructions. In *Proceedings of ACL*, pages 13484–13508.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of EMNLP*, pages 5085–5109.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022b. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022c. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, pages 24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of ACL*, pages 1423–1436.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP*, pages 2369–2380.

Seonghyeon Ye, Hyeonbin Hwang, Sohee Yang, Hyeongu Yun, Yireun Kim, and Minjoon Seo. 2023. Investigating the effectiveness of task-agnostic prefix prompt for instruction following. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Shi Yu, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2023a. Openmatch-v2: An all-in-one multi-modality plm-based information retrieval toolkit. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3160–3164.

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023b. Augmentation-adapted retriever improves generalization of language models as generic plug-in. In *Proceedings of ACL*, pages 2421–2436.

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

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

LLM	Setting	#Token	Passage Compression				Instruction Compression		
			PopQA	KILT			Alpaca+		
				NQ	TrivialQA	HotpotQA	Seen	Unseen	Human
FlanT5-large	Unified	5	24.3	21.2	47.6	18.3	24.0	28.1	14.0
		10	26.6	22.8	50.7	19.1	23.6	28.4	13.3
		20	30.9	26.0	56.6	20.8	24.3	29.6	14.1
	Disentangled	1	16.8	15.1	36.9	16.7	20.4	20.2	10.4
		5	27.4	23.7	52.4	19.3	24.8	27.4	13.6
		10	32.0	27.0	57.3	20.6	25.7	30.1	14.0
		15	34.5	27.8	58.8	21.3	25.4	**31.3**	14.9
		20	**35.8**	**28.6**	**59.9**	**21.8**	**25.6**	30.6	**15.2**
Llama-7b	Unified	5	29.2	25.9	67.4	21.5	24.5	24.9	7.7
		10	31.0	25.4	66.2	21.1	24.0	24.9	7.3
		20	33.5	27.8	68.6	22.0	21.7	25.2	7.9
	Disentangled	1	23.3	23.1	62.8	19.4	24.0	23.8	8.6
		5	31.2	26.8	68.5	22.2	23.9	25.2	8.9
		10	34.9	28.9	69.6	22.6	**24.7**	25.3	8.8
		15	36.6	28.8	70.1	22.8	24.5	25.4	8.8
		20	**37.6**	**29.6**	**70.3**	**22.8**	24.5	**25.4**	**9.0**
Llama2-7b-hf	Unified	5	30.6	28.5	70.2	23.5	22.3	23.8	7.6
		10	32.8	28.7	69.8	22.9	21.9	24.2	7.6
		20	34.4	29.7	71.2	24.1	21.6	24.5	6.9
	Disentangled	1	25.3	26.4	67.8	22.0	20.7	21.7	7.5
		5	32.7	29.4	71.2	23.9	21.6	24.2	7.4
		10	35.9	30.8	71.9	24.4	**22.7**	**24.9**	8.4
		15	37.6	31.4	72.3	**24.8**	22.7	24.7	7.8
		20	**38.4**	**31.4**	**72.7**	24.7	22.0	25.1	**8.8**
Llama-13b	Unified	5	32.4	28.0	73.7	23.5	24.6	25.7	8.6
		10	34.0	28.4	73.0	23.7	24.2	25.4	8.0
		20	35.8	29.8	74.4	24.1	23.4	**26.1**	7.6
	Disentangled	1	27.0	26.0	71.7	23.2	24.0	23.7	9.0
		5	34.0	29.3	74.0	24.0	24.7	25.6	9.0
		10	36.9	30.8	74.5	24.4	**24.8**	26.0	9.6
		15	38.4	30.8	74.3	**24.6**	24.6	25.8	**9.7**
		20	**39.3**	**31.0**	**74.5**	24.4	24.4	25.7	9.6

Table 6: Additional Ablation Studies. For FlanT5-large, we employ the embedding-based compression modeling method, as well as verbalize the gist representations as prompts for other models.

A Appendix

A.1 License

We show the licenses of the datasets that we use. PopQA, MS MARCO and KILT use MIT license. Alpaca+ and NVI2 use Apache license. All of these licenses and agreements allow their data for academic use.

A.2 Additional Ablation Studies on Gist-COCO

We conduct additional ablation studies to delicately explore the compression effectiveness on different models with different gist modeling methods.

As shown in Table 6, the disentangled gist modeling method is more effective than the unified gist modeling method, when generalizing the compression capabilities of Gist-COCO to different LLMs. With an increase in the number of gist tokens, there is an enhancement in gist verbalization performance. However, once the number of gist tokens surpasses 10, the rate of improvement slows down, impacting performances on certain tasks.

Type Name	Total
Math & Logic Problems	504
Writing & Language Problems	956
Programming Problems	273
Knowledge	519

Table 7: Data Statistics of Different Classifications of Alpaca+ Data.

A.3 Data Classification of Alpaca+ Data

We employ ChatGPT-3.5 to categorize the data within the Alpaca+ dataset into four distinct groups. The data statistics are shown in Table 7.