2024-Winter Lab Seminar

LLMLingua
: Compressing Prompts for Accelerated Inference of Large Language Models

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, Lili Qiu

Microsoft Research

Yejin Yoon

HYU NLP Lab., Hanyang University

stillwithyou@hanyang.ac.kr

JAN. 17. 2024

#paper-of-the-day

Huiqiang Jiang, et al. (Microsoft) "**LLMLingua**: Accelerating and Enhancing LLMs in Long Context Scenarios via Prompt Compression" (EMNLP 2023)

Microsoft ***LLMLingua***: Compressing Prompts for Accelerated Inference of Large Language Models

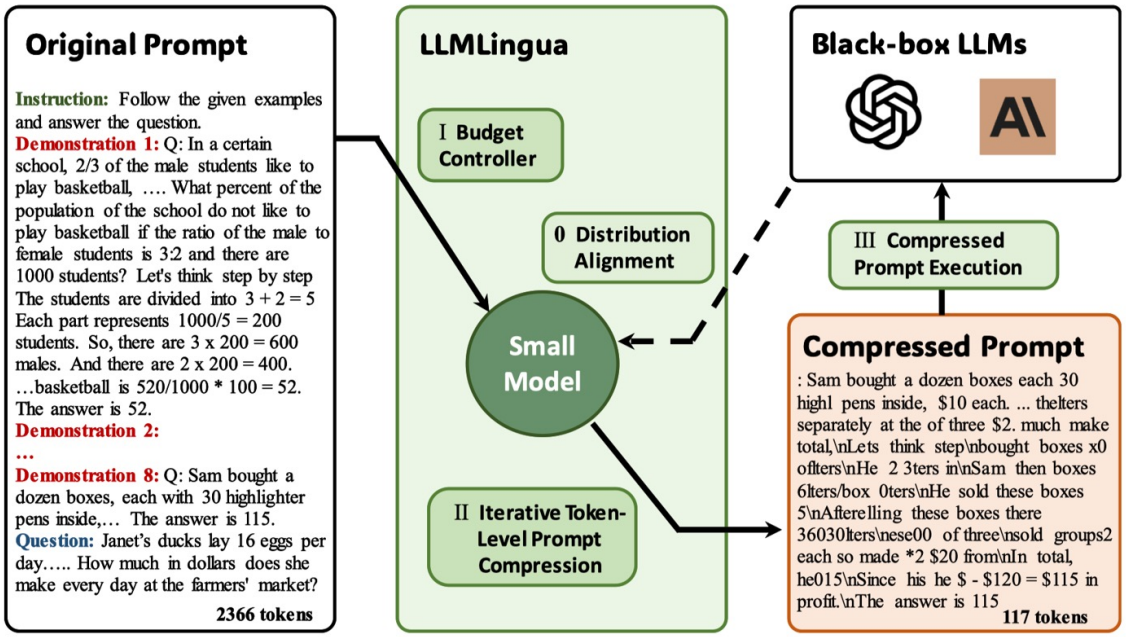


Figure 1: Framework of the proposed approach *LLMLingua*.

https://aka.ms/LLMLingua

Now you can use ***LLMLingua***!

A simple and efficient method to compress prompt **up to *20x***.

💰 **Saving cost**, not only prompt, but also the generation length;
📝 **Support longer contexts;**
⚖️ **Robustness**, no need any training for the LLMs;
🤷 **Keeping** the original prompt knowledge like ICL, reasoning, etc.
📜 **KV-Cache compression**, speedup inference;
🪃 **GPT-4 can recovery all key information from compressed prompt.**

#paper-of-the-day

Huiqiang Jiang, et al. (Microsoft Research) "**LongLLMLingua**: Accelerating and Enhancing LLMs in Long Context Scenarios via Prompt Compression" (ICLR2024 –ing)

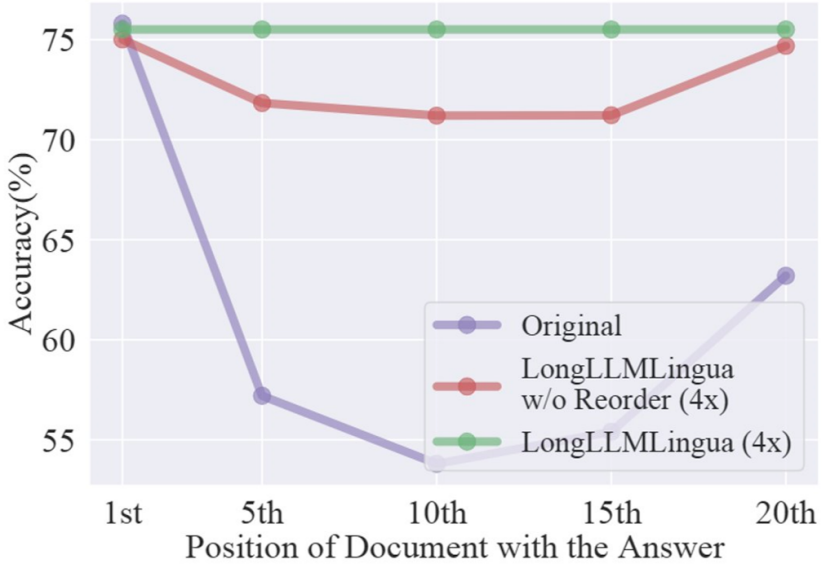
Microsoft

LongLLMLingua: Accelerating and Enhancing LLMs in Long Context Scenarios via Prompt Compression

The main ideas involve **compressing the prompt** in a two-stage process, as shown by the **red line**, which is a significant improvement over the original curve:

- *Coarse-grained compression through document-level perplexity*
- *Fine-grained compression of the remaining text using token perplexity*

Instead of fighting against the positional effects, we attempt to utilize them to our advantage through **document reordering**, as illustrated by the **green line**. *In this manner, the most critical passages are placed at the beginning and the end of the context.* Moreover, the entire process becomes more cost-effective and faster since it only requires handling 1/4 of the original context.



(b) Performance v.s. Key Information Position

🔍 **LongLLMLingua: Enhancing LLM's Perception of Key Information**

https://aka.ms/LLMLingua

Contents

What are Covered in this Presentation

- **Details of (Long)LLMLingua**
  - **LLMLingua :** Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, Lili Qiu. "LLMLingua: Compressing Prompts for Accelerated Inference of Large Language Models." (EMNLP2023)
  - **LongLLMLingua :** Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, Lili Qiu. " LongLLMLingua: Accelerating and Enhancing LLMs in Long Context Scenarios via Prompt Compression." (ICLR2023 submitted)

- **Intuitive concepts of predecessors**
  - **Selective Context :**
    - Yucheng Li (University of Surrey) "Unlocking Context Constraints of LLMs: Enhancing Context Efficiency of LLMs with Self-Information-Based Content Filtering."
    - Yucheng Li, Bo Dong, Chenghua Lin, Frank Guerin (University of Surrey) “Compressing Context to Enhance Inference Efficiency of Large Language Models” (EMNLP 2023)

1. Pre-Requisites

- Perplexity

Pre-Requisites

- **Perplexity (PPL)**
  - Perplexity measures the uncertainty of a LM in predicting the next token.
    - Entropy represents the level of disorder or uncertainty within the model's predictions.
    - Tokens with lower PPL contribute less to overall entropy, implying more predictable choices.
  - Lower PPL indicates higher prediction confidence.
  - It is defined as the exponentiated average negative log-likelihood of a sequence of words.
    - Normalized inverse probability of the test set

$$PPL(W) = P(w_1, w_2, \ldots, w_N)^{\frac{1}{N}} = \left(\prod_{i=1}^{N} \frac{1}{P(w_i | w_1, w_2, \ldots, w_{i-1})} \right)^{\frac{1}{N}}$$

overall probability of the word sequence

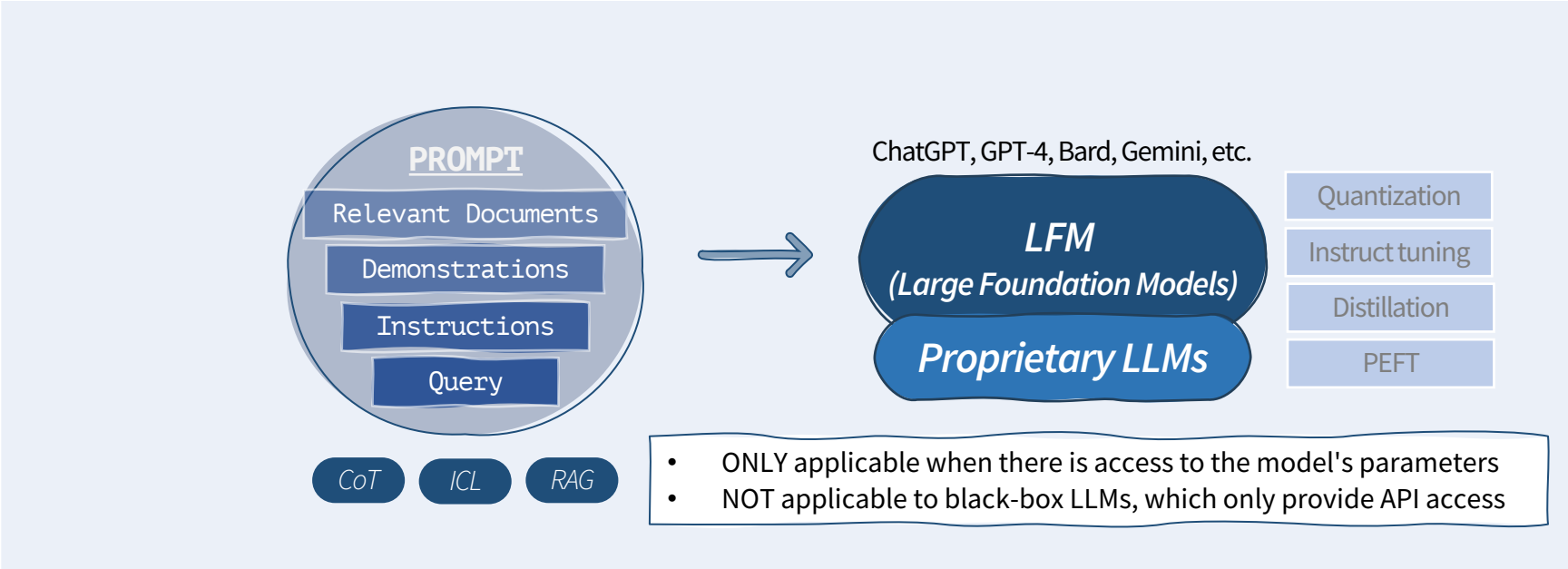
conditional probability of word w_i given all the preceding words $w_1, w_2, \ldots, w_{i-1}$

2. Background

- Selective-context
- Problem formulation

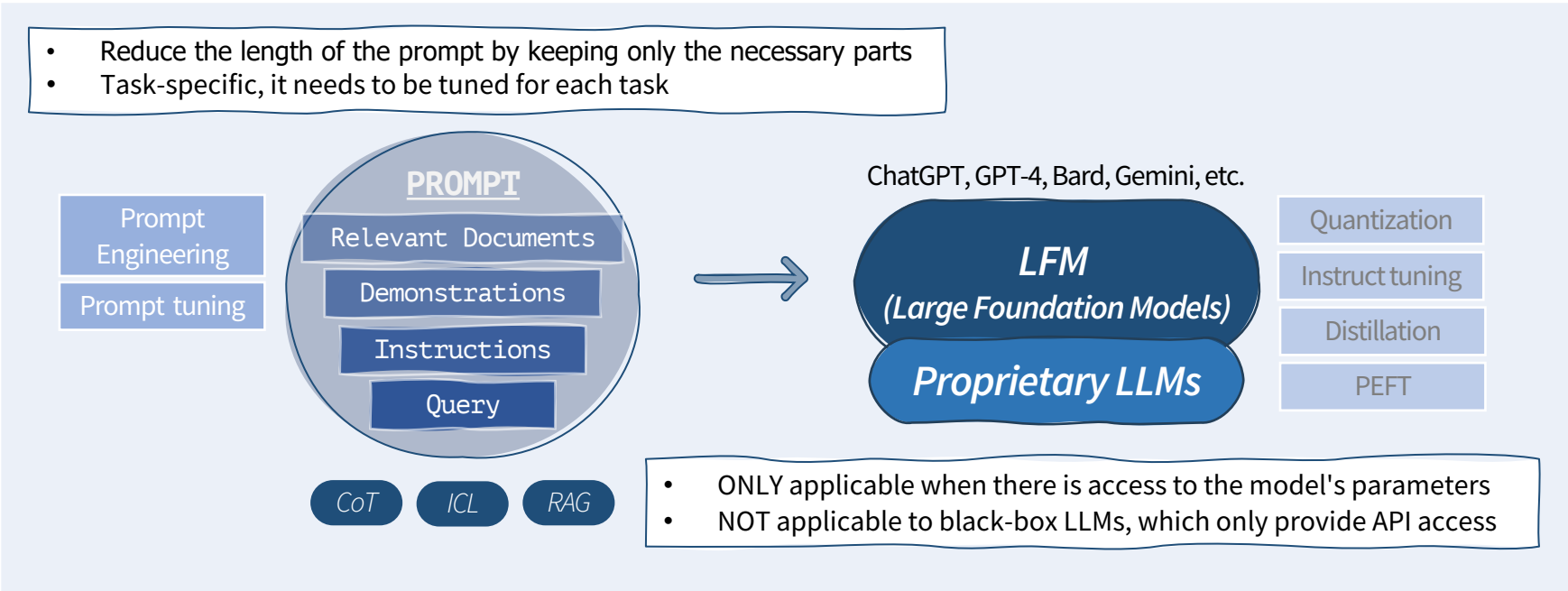
Background

- **Introduction**
  - ChatGPT is good at following instructions, but to make it perform better, various prompt techniques such as CoT, ICL, RAG, etc., are used.
  - The computing cost of LLMs and the size of the prompt significantly impact performance.



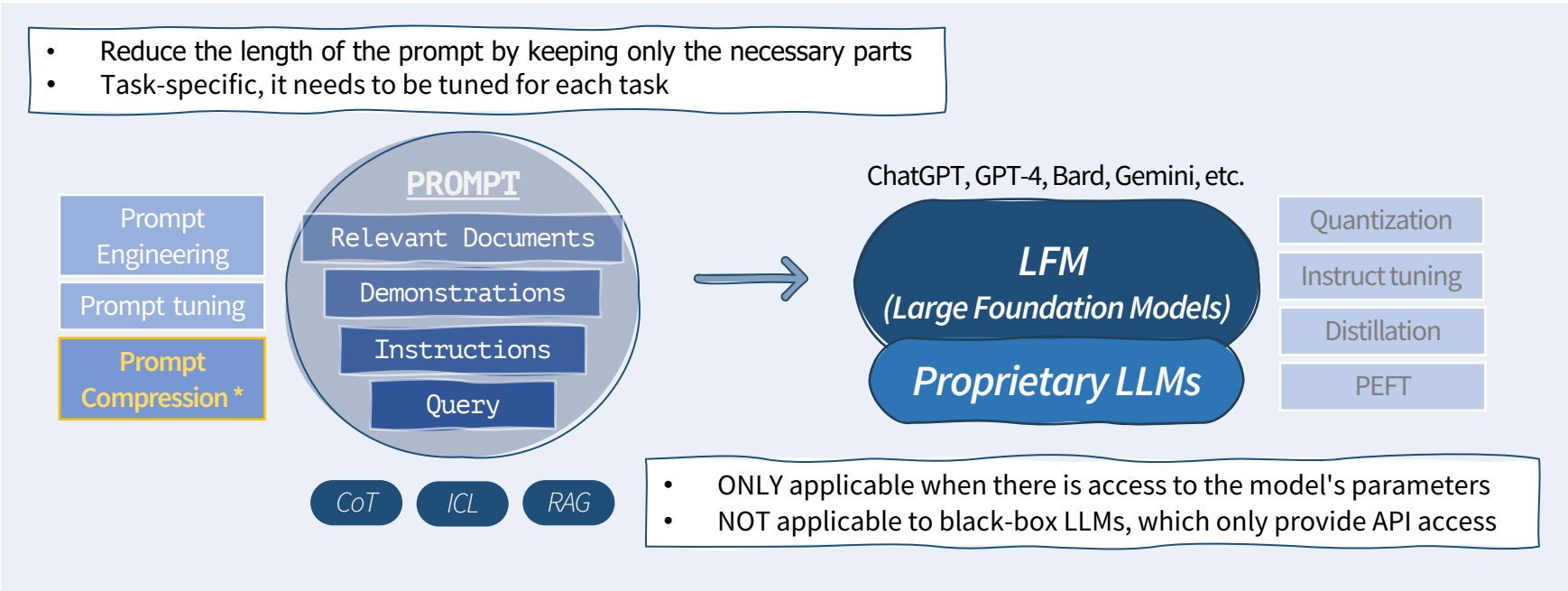
Background

- **Introduction**
  - ChatGPT is good at following instructions, but to make it perform better, various prompt techniques such as CoT, ICL, RAG, etc., are used.
  - The computing cost of LLMs and the size of the prompt significantly impact performance.
    - Some attempts to optimize inference costs in terms of input prompts



Background

- **Introduction**
  - ChatGPT is good at following instructions, but to make it perform better, various prompt techniques such as CoT, ICL, RAG, etc., are used.
  - The computing cost of LLMs and the size of the prompt significantly impact performance.
    - Some attempts to optimize inference costs in terms of input prompts.



Background

- **Related Work ~ Compression**
  - Premise: Natural language prompts are redundant.
    - No gradient flow is preferred.
  - ***Selective context*** (Li et al. EMNLP 2023)
    - Use sLLMs to calculate the lexical unit information of a prompt.

Self-information (based on entropy) $I(x_i) = -\log_2 P(x_i|x_0, x_1, ..., x_{i-1})$

    - Delete tokens with less information to compress the prompt.

Original: INTRODUCTION Continual Learning (~~CL~~) ~~,~~ ~~also known as Lifelong Learning~~ ~~,~~ ~~is~~ a promising learning paradigm to design models ~~that~~ have to ~~learn~~ how ~~to perform multiple tasks~~ across ~~different environments~~ over ~~their lifetime .~~ [To uniform the language and enhance ~~the readability of the paper we~~ adopt the unique term continual learning ~~(CL) .]~~ . Ideal CL models in ~~the real world~~ should ~~be~~ deal ~~with~~ domain shifts ~~,~~ researchers ~~have~~ recently started ~~to~~ sample tasks ~~from~~ two different datasets ~~.~~ For instance ~~,~~ proposed to train and evaluate ~~a model~~ on Imagenet first ~~and then~~ challenge ~~its performance on the~~ Places365 ~~dataset .~~ considers more scenarios ~~,~~ starting ~~with~~ Imagenet or Places365 ~~, and then moving on to~~ the VOC/CUB/Scenes datasets ~~.~~ Few works propose more advanced scenarios built ~~on~~ top ~~of~~ more than two datasets ~~.~~

Filtered: INTRODUCTION Continual Learning (a promising learning paradigm to design models have to how across overTo uniform the language and enhance adopt the unique term continual learning Ideal CL models in should deal domain shifts researchers recently started sample tasks two different datasets For instance proposed to train and evaluate on Imagenet first challenge Places365considers more scenarios starting Imagenet or Places365 the VOC/CUB/Scenes datasetsFew works propose more advanced scenarios built top more than two datasets

Figure 2: A visualisation of selective context. Darker colour indicates larger value of self-information.

Brief Introduction

- **In terms of** information theory**,**
  tokens with low PPL **contribute minimally to** ***the overall entropy of a LM*****.**

- **Coarse-to-fine Prompt Compression Method**
  - Budget controller
  - ITPC (Iterative Token-level Prompt Compression)
  - Distribution Alignment

Problem Formulation

- **Notation**
  - Original prompt

$$x = (x^{ins}, x^{deoms}, x^{que})$$

  - Compressed prompt

$$\tilde{x} = \{\tilde{x}_i\}_{i=1}^{\tilde{L}}$$

  - Length

$$L = L^{ins} + L^{deoms} + L^{que}$$

  - Compression ratio

$$\tau = \frac{\tilde{L}}{L}$$

    (The smaller τ, the lower inference cost)
  - $\tilde{x}$ as input to LLM should produce results similar to those produced by taking x as input.

$$\min_{\tilde{x},\tau} \mathrm{KL}\left(P(\tilde{x}_G|\tilde{x}), P(x_G|x)\right)$$

A prompt compression system is designed to generate a compressed prompt $\widetilde{\boldsymbol{x}} = \{\widetilde{x}_i\}_{i=1}^{\widetilde{L}}$ from a given original prompt $\boldsymbol{x} = (\boldsymbol{x}^{\text{ins}}, \boldsymbol{x}^{\text{dems}}, \boldsymbol{x}^{\text{que}})$, where $\boldsymbol{x}^{\text{ins}} = \{x_i^{\text{ins}}\}_{i=1}^{L^{\text{ins}}}$, $\boldsymbol{x}^{\text{dems}} = \{x_i^{\text{dems}}\}_{i=1}^{L^{\text{dems}}}$, and $\boldsymbol{x}^{\text{que}} = \{x_i^{\text{que}}\}_{i=1}^{L^{\text{que}}}$ denote the instruction, demonstrations, and the question in the original prompt $\boldsymbol{x}$. $\widetilde{L}$, L_{ins}, L_{dems}, and L_{que} represent the numbers of tokens in $\widetilde{\boldsymbol{x}}$, $\boldsymbol{x}^{\text{ins}}$, $\boldsymbol{x}^{\text{dems}}$, and $\boldsymbol{x}^{\text{que}}$, respectively. Let $L = L_{\text{ins}} + L_{\text{dems}} + L_{\text{que}}$ denote the total sequence length of $\boldsymbol{x}$, the compression rate is defined as $\tau = \widetilde{L}/L$, $\tau \in [0, 1]$, and the compression ratio is $1/\tau$. A smaller value of τ implies a lower inference cost, which is preferable. Let $\widetilde{\boldsymbol{x}}_G$ represent the LLM-generated results derived by $\widetilde{\boldsymbol{x}}$ and $\boldsymbol{x}_G$ denotes the tokens derived by $\boldsymbol{x}$, the distribution of $\widetilde{\boldsymbol{x}}_G$ is expected to be as similar to $\boldsymbol{x}_G$ as possible. This can be formulated as:

$$\min_{\widetilde{\boldsymbol{x}},\tau} \mathrm{KL}(P(\widetilde{\boldsymbol{x}}_G|\widetilde{\boldsymbol{x}}), P(\boldsymbol{x}_G|\boldsymbol{x})), \qquad (1)$$

3. Suggestion

- LLMLingua

  1) Budget controller

  2) ITPC

  3) Distribution Alignment

Framework of *LLMLingua*

Microsoft ***LLMLingua***: Compressing Prompts for Accelerated Inference of Large Language Models

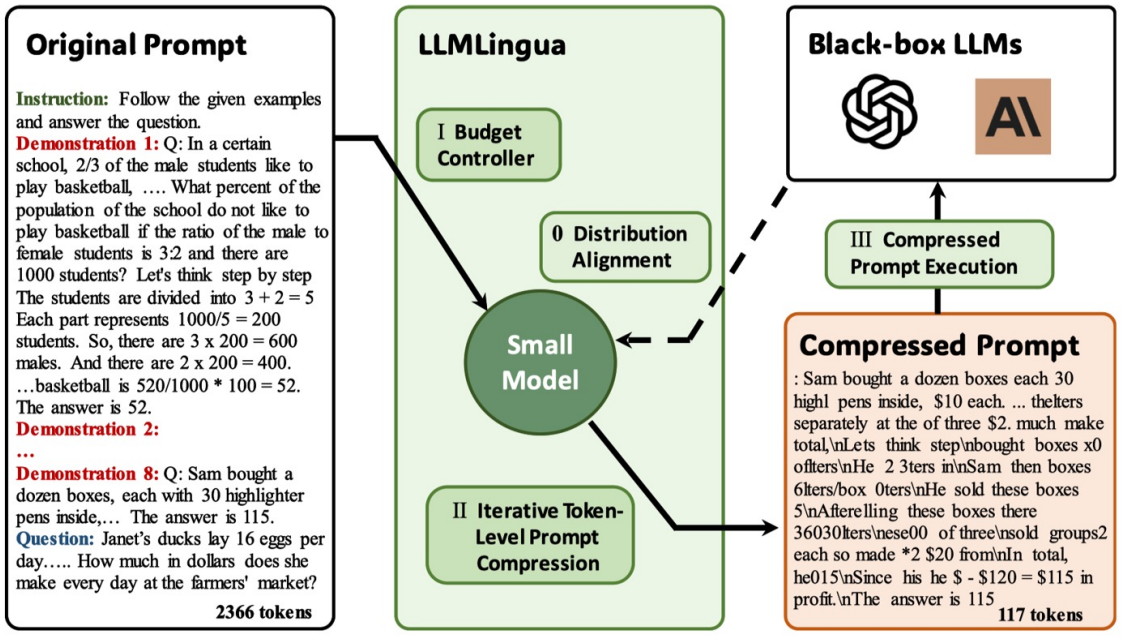


Figure 1: Framework of the proposed approach *LLMLingua*.

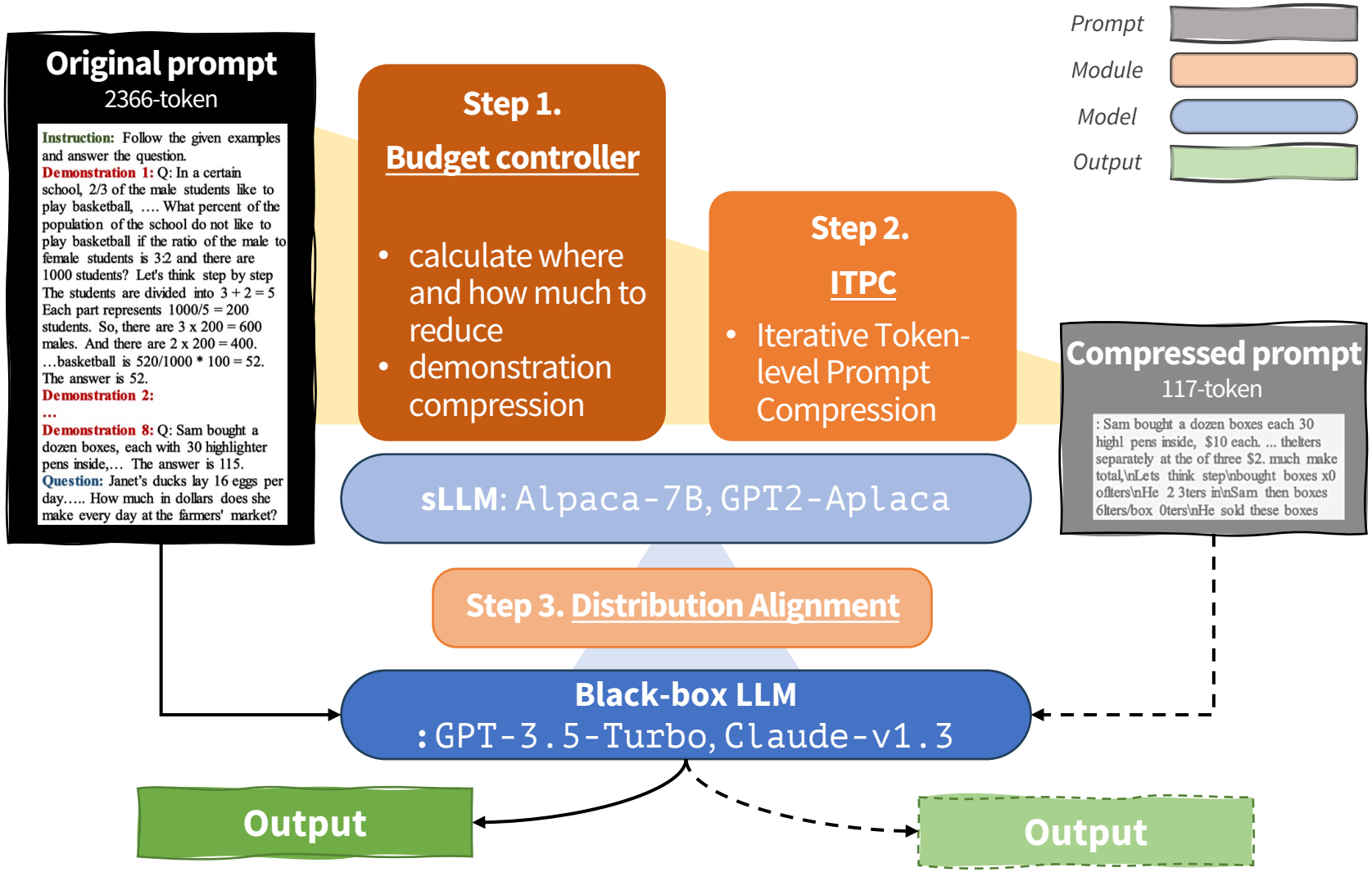
https://aka.ms/LLMLingua

Now you can use ***LLMLingua***!

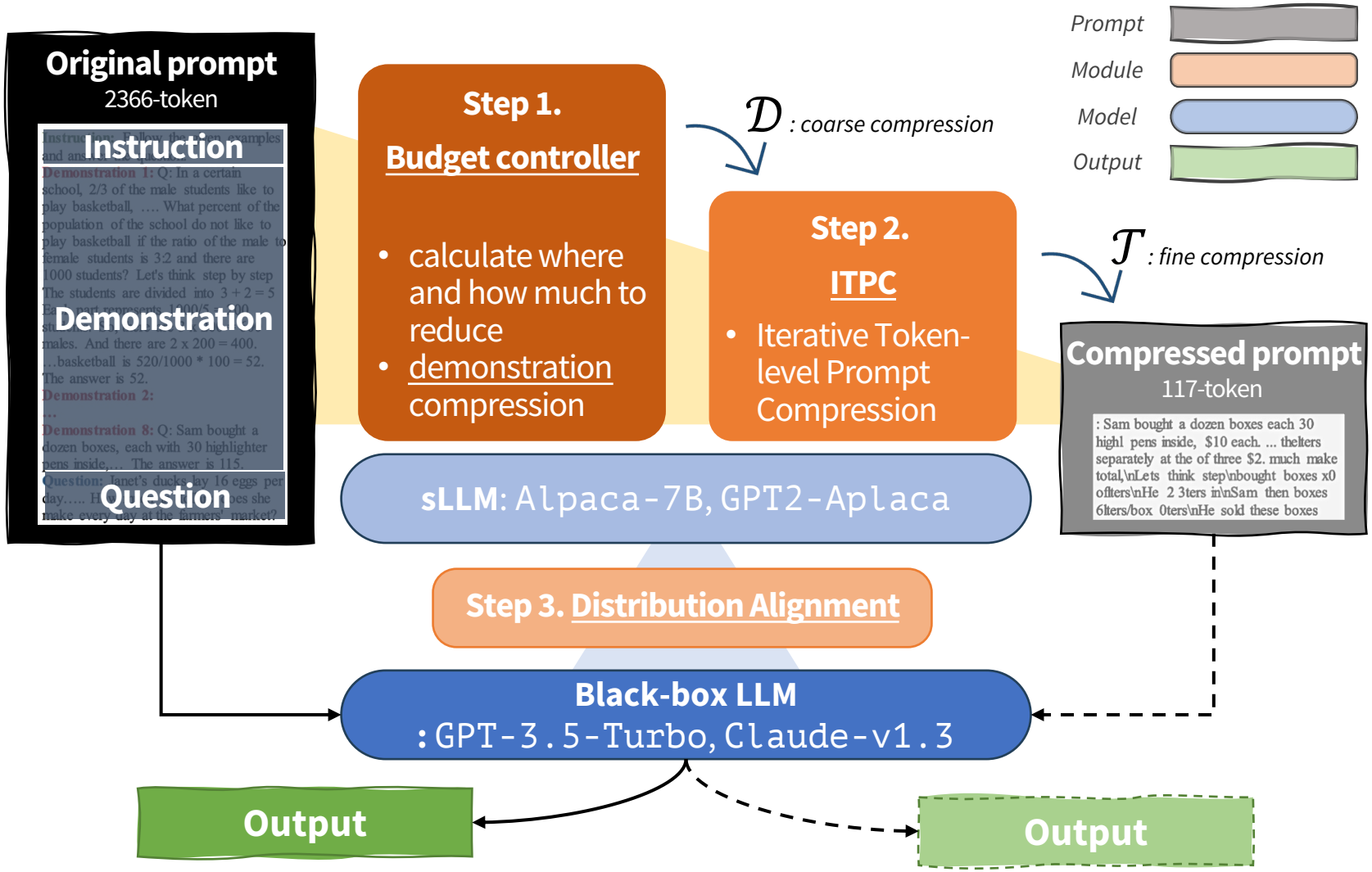
A simple and efficient method to compress prompt **up to *20x***.

💰 **Saving cost**, not only prompt, but also the generation length;
📝 **Support longer contexts;**
⚖️ **Robustness**, no need any training for the LLMs;
🤷 **Keeping** the original prompt knowledge like ICL, reasoning, etc.
📜 **KV-Cache compression**, speedup inference;
🪃 **GPT-4 can recovery all key information from compressed prompt.**

Framework of *LLMLingua*



Framework of *LLMLingua*



1) Budget Controller

- **Demonstration –level compression + $\Delta\tau$ per each component**

1. Demonstration-level compression
   - Maintain critical information: instruction and question
     - Allocate a larger portion of the budget to preserve the essence of 'Instruction' and 'Question'.
     - Reduce redundancy in 'Demonstration' sections to economize space.
2. Sentence –level compression
   - When a high compression ratio is required, token-level dropout risks excessive loss of semantic information; granularity concern
   - Implement sentence-level dropout as a preliminary measure.

Algorithm 1 Pseudo code of Budget Controller.

Input: A small language model $\mathcal{M}_s$; the original prompt $\boldsymbol{x} = (\boldsymbol{x}^{\text{ins}}, \boldsymbol{x}^{\text{dems}}, \boldsymbol{x}^{\text{que}})$.

Set the selected demonstration set $\mathcal{D} = \phi$.
Get demonstration compression rate τ_{dem} by Eq.(2).
Calculate the perplexity of each demonstration via $\mathcal{M}_s$.
Rank all demonstrations in descending order of their perplexity as a list $(\boldsymbol{x}^{\text{dem}}_{(1)}, ..., \boldsymbol{x}^{\text{dem}}_{(N)})$, where N is the number of demonstrations, $\boldsymbol{x}^{\text{dem}}_{(i)}$ is the i-th demonstration.
for $i = 1$ **do**
    if $\widetilde{L}_{\mathcal{D}} > k \cdot \tau_{\text{dems}} L_{\text{dems}}$ **then**
        Break.
    end if
    Append $\boldsymbol{x}^{\text{dem}}_{(i)}$ to $\mathcal{D}$.
    $i = i + 1$
end for
Allocate remaining budget to $\boldsymbol{x}^{\text{ins}}$ and $\boldsymbol{x}^{\text{que}}$ via Eq. (3).

Output: The subset of demonstrations $\mathcal{D}$ obtained from coarse-grained compression; Additional budget $\Delta\tau_{\text{ins,que}}$ for the instruction and the question.

1) Budget Controller

- **Derive compression ratio for demonstrations for each component**

1. Demonstration-level compression
   - Maintain critical information: instruction and question
     - Allocate a larger portion of the budget to preserve the essence of 'Instruction' and 'Question'.
     - Reduce redundancy in 'Demonstration' sections to economize space.
2. Sentence –level compression
   - When a high compression ratio is required, token-level dropout risks excessive loss of semantic information; granularity concern
   - Implement sentence-level dropout as a preliminary measure.

$$\tau_{\text{dems}} = \frac{\tau L - (\tau_{\text{ins}} L_{\text{ins}} + \tau_{\text{que}} L_{\text{que}})}{L_{\text{dems}}}.$$

Derive compression ratio for demonstrations ▲

$$\Delta\tau = \frac{k \cdot \tau_{\text{dems}} L_{\text{dems}} - \widetilde{L}_{\mathcal{D}}}{L_{\text{ins}} + L_{\text{que}}},$$

Adjust compression ratios for instruction and question ▲

2) Iterative Token-level Prompt Compression

- **ITPC; Token-level Compression**
  - The intrinsic limitation to utilize perplexity: independent assumption; similar to shortcomings of MLM

Demonstration-level compressed prompt

$$p(\tilde{x}) = \prod_{i=1}^{\tilde{L}} p(\tilde{x}_i | \tilde{x}_{<i}) \approx p(x') = \prod_{i=1}^{L'} p(x_i | \tilde{x}_{<i}, \bar{x}_{<i})$$

  - Divide Prompt into segments and calculate PPL in those segments (preserving inter-dependencies)

$$p(\tilde{s}_j) = \prod_{i=1}^{\sum_k^j \tilde{L}_{s,k}} p(\tilde{s}_{j,i} | \tilde{s}_{j,<i}, \tilde{s}_{<j}) \approx \prod_{i=1}^{L_{s,j}+\sum_k^{j-1} \tilde{L}_{s,k}} p(s_{j,i} | s_{j,<i}, \tilde{s}_{<i})$$

Divide prompt into segments

i-th token in j-th segment

2) Iterative Token-level Prompt Compression

- **ITPC; Token-level Compression**

1. Get the conditional probabilities (PPL distribution)
2. Get dynamic threshold for segment compression ratio

$$\tau_{s_j} = \begin{cases} \tau_{\text{ins}} + \Delta\tau, & \text{if } s_j \text{ from } x^{\text{ins}}, \\ \tau_{\text{dems}}, & \text{if } s_j \text{ from } x^{\mathcal{D}}, \\ \tau_{\text{que}} + \Delta\tau, & \text{if } s_j \text{ from } x^{\text{que}}. \end{cases}$$

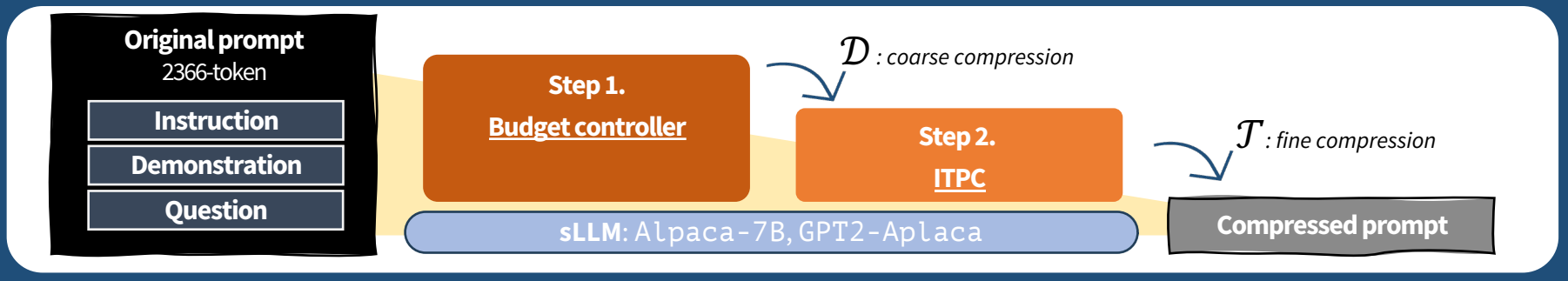
$$\widetilde{s}_j = \{s_{j,i} | p(s_{j,i}) > \gamma_j\}$$

Algorithm 2 Pseudo code of Iterative Token-level Prompt Compression (ITPC).

Input: A small language model $\mathcal{M}_s$; the prompt from budget controller $x' = (x^{\text{ins}}, x^{\mathcal{D}}, x^{\text{que}})$; target compression rate τ, adjusted compression rate $\Delta\tau_{\text{ins,que}}$.

Set the selected token set $\mathcal{T} = \phi$
Get segment set $\mathcal{S}$.
for $i = 1, 2, \ldots, m$ **do**
    Get the conditional probabilities $p(s_i)$ via Eq.(5)
    Get the compression threshold γ_i with Eq. (6).
    Append the compressed token to $\mathcal{T}$ via Eq.(7).
end for
Concatenate all tokens in $\mathcal{T}$ as $\widetilde{x}$.

Output: The compressed prompt $\widetilde{x}$.



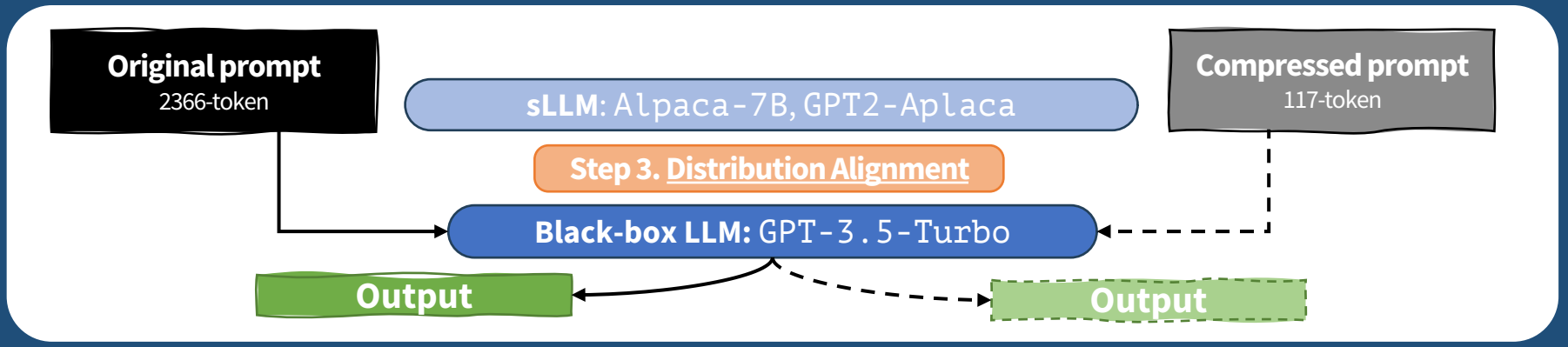
3) Distribution Alignment

- **Instruction tuning $\mathcal{M}_s$**

$$\min_{\boldsymbol{\theta}_s} \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}\left(\mathbf{x}_i, \mathbf{y}_{i,\mathrm{LLM}}; \boldsymbol{\theta}_{\mathcal{M}_s}\right)\right],$$

parameters of sLLM

- Begin with a pre-trained small language model $\mathcal{M}_s$
- Utilize datasets generated by a LLM to perform instruction tuning on $\mathcal{M}_s$
- Minimize the expected value of the loss function

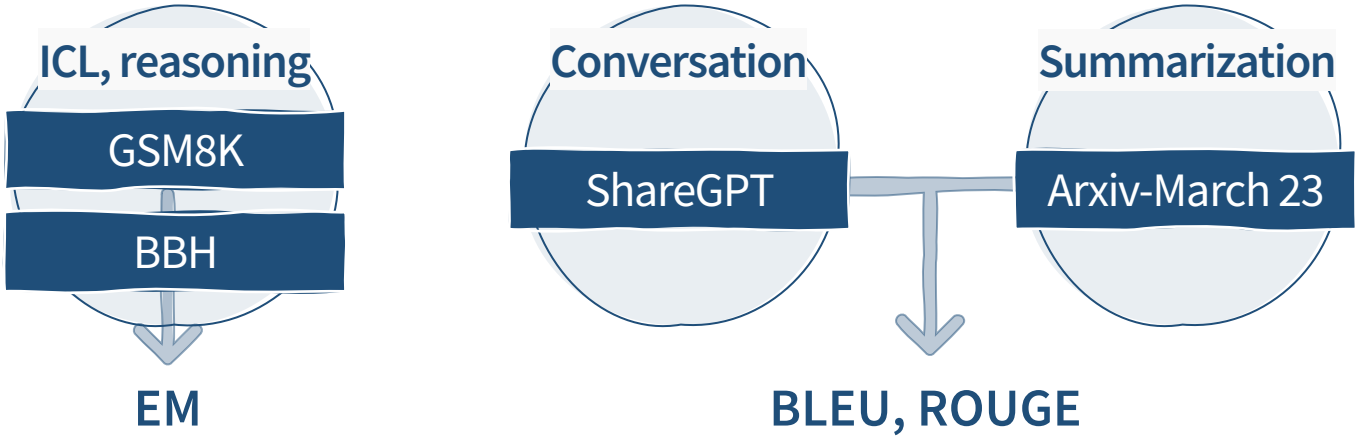


4. Effect

- Experiment
- Discussion

Experiments

- **Datasets**



- **Baseline Models**
  - Target LLMs
    : `GPT-3.5-Turbo`, `Claude-v1.3`
  - sLLMs ($\mathcal{M}_s$): `Alpaca-7B`, `GPT2-Aplaca`
  - Hyperparameters
    - $k = 2, \tau_{ins} = 0.85, \tau_{que} = 0.9$
      * k: granular control coefficient
    - ITPC #(segment) = 100

- **Baseline Methods**
  - GPT4-Genereation
  - Random Selection
  - Selective-Context

Experiments – ICL & Reasoning

- **Summary:**
  - Increased overall difference
  - Best score in all experiments
- **Details**
  - A slight performance drop when compression ratios were increased (1/2-shot or 1/4-shot)
    - GSM8K EM scores on 14x and 20x dropped by 1.44 and 1.52 respectively.
  - The performance of selective-context is generally poor.
    - By limiting the prompt to phrase-level compression, it may discard some of the important reasoning information in some cases.

Methods	GSM8K			BBH		
	EM	Tokens	$1/\tau$	EM	Tokens	$1/\tau$
Full-shot	78.85	2,366	-	70.07	774	-
1-shot constraint						
1-shot	77.10	422	6x	69.60	284	3x
Selective-Context	53.98	452	5x	54.27	276	3x
GPT4 Generation	71.87	496	5x	27.13	260	3x
Ours	**79.08**	446	5x	**70.11**	288	3x
half-shot constraint						
Sentence Selection	72.33	230	10x	39.56	175	4x
Selective-Context	52.99	218	11x	54.02	155	5x
GPT4 Generation	68.61	223	11x	27.09	161	5x
Ours	**77.41**	171	14x	**61.60**	171	5x
quarter-shot constraint						
Sentence Selection	66.67	195	12x	46.00	109	7x
Selective-Context	44.20	157	15x	47.37	108	7x
GPT4 Generation	56.33	188	20x	26.81	101	8x
Ours	**77.33**	117	20x	**56.85**	110	7x
zero-shot	48.75†	11	215x	32.32	16	48x
Simple Prompt	74.9	691	3x	-	-	-

Table 2: Performance of different methods under different target compression ratios on the GSM8K mathematical reasoning and Big-bench Hard (BBH) datasets. †We also include the instruction of the prompt in zero-shot experiments for a vertical comparison.

Experiments - Dialogue / Summarization

Methods	ShareGPT							Arxiv-March23						
	BLEU	Rouge1	Rouge2	RougeL	BS F1	Tokens	$1/\tau$	BLEU	Rouge1	Rouge2	RougeL	BS F1	Tokens	$1/\tau$
Constraint I	*2x constraint*							*350 tokens constraint*						
Sentence Selection	**28.59**	46.11	**31.07**	37.94	88.64	388	1.5x	22.77	50.1	25.93	33.63	88.21	379	4x
Selective-Context	25.42	46.47	29.09	36.99	88.92	307	1.9x	21.41	51.3	27.94	36.73	89.60	356	4x
Ours	27.36	**48.87**	30.32	**38.55**	**89.52**	304	1.9x	**23.15**	**54.21**	**32.66**	**42.74**	**90.33**	345	4x
Constraint II	*3x constraint*							*175 tokens constraint*						
Sentence Selection	18.94	35.17	18.96	26.75	85.63	255	2.3x	12.41	38.91	14.25	26.72	87.09	229	7x
Selective-Context	15.79	38.42	20.55	28.89	87.12	180	3.3x	12.23	42.47	19.48	29.47	88.16	185	8x
Ours	**19.55**	**40.81**	**22.68**	**30.98**	**87.70**	177	3.3x	**13.45**	**44.36**	**24.86**	**34.94**	**89.03**	176	9x

Table 1: Performance of different methods under different target compression ratios on the conversation (ShareGPT) and summarization (Arxiv-March23) task.

Discussion

	EM	Tokens	$1/\tau$
Ours in 1-shot constraint	83.51	439	5x
Ours in half-shot constraint	82.61	171	14x
Simple Prompt	81.8	691	3x

Table 4: Ours method on GSM8K using Claude-v1.3.

	EM	Tokens	$1/\tau$
Ours with GPT2 in 1-shot constraint	77.02	447	5x
Ours with GPT2 in half-shot constraint	76.42	173	14x
Ours with GPT2 in quarter-shot constraint	76.27	128	18x

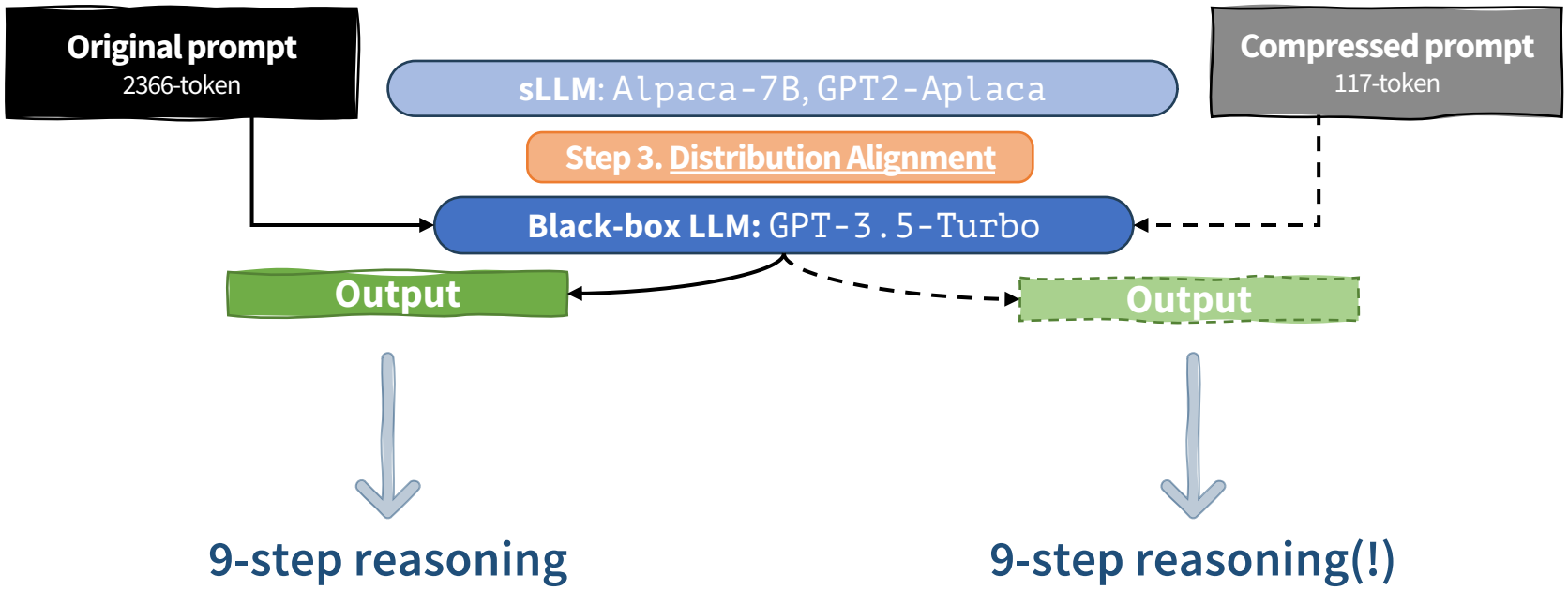
Table 5: Our method on GSM8K with GPT2-Alpaca as the small language model.

- (Table 3) Ablation study on GSM8K
- (Table 4) Same trend for other models besides GPT-3.5-Turbo
- (Table 5) This doesn't seem to work as well for smaller models like the GPT-2(1.5B).

	EM	Tokens	$1/\tau$
Ours	**79.08**	439	5x
- w/o Iterative Token-level Prompt Compression	72.93	453	5x
- w/o Budget Controller	73.62	486	5x
- w/o Dynamic Compression Ratio	77.26	457	5x
- w/ Random Selection in Budget Controller	72.78	477	5x
- w/o Distribution Alignment	78.62	452	5x
- w/ Remove Stop Words	76.27	1,882	1.3x

Table 3: Ablation study on GSM8K in 1-shot constraint.

Discussion



- (Appendix E) When querying LLM with a compressed prompt that was barely recognizable to humans, LLM generate a multi-step by step answer just like the original!

Discussion

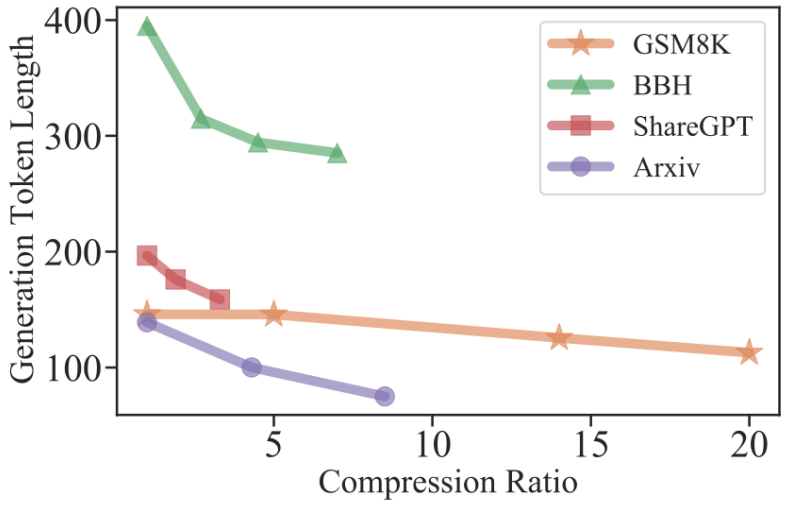


Figure 2: The distribution of generated token lengths at varying compression ratios ($1/\tau$).

$$c = (L + kL/\tau + L/\tau) \cdot c_{\text{small}} + L/\tau \cdot c_{\text{LLMs}},$$

$1/\tau$	1x	2x	5x	10x
End-to-End w/o LLMLingua	8.6	-	-	-
End-to-End w/ LLMLingua	-	4.9(1.7x)	2.3(3.3x)	1.3(5.7x)
LLMLingua	-	0.8	0.3	0.2

Table 6: Latency (s) comparison on GSM8K.

- (Figure 2) the relationship between the compression ratio▲ and the length of the generated text▼
- (Table 6) Latency Efficiency

Discussion

- **Recovering the Compressed Prompt using LLMs (GPT-4)**
  - LLM effectively understands the semantic information in the compressed prompt and restores it appropriately. (Even if humans can't)
  - How much GPT-4 recovers depends on the compression ratio and whether it uses a sLLM.
  - Alpaca-7B restored the complete 9-step reasoning process, while GPT2-Alpaca only restored 7-step reasoning, with errors.

Compressed Prompt:
: Sam bought a dozen boxes each 30 highl pens inside, $10 each. He reanged five of boxes into of six each $3 per. He sold the thelters separately at the of three $2. much make total,
Lets think step
bought boxes x0 oflters
He 2 3ters in
Sam then boxes 6lters/box 0ters
He sold these boxes 5
Afterelling these boxes there 36030lters
ese00 of three
sold groups2 each so made *2 $20 from
In total, he015
Since his he $ - $120 = $115 in profit.
The answer is 115

Discussion

- **Recovering the Compressed Prompt using LLMs (GPT-4)**
  - LLM effectively understands the semantic information in the compressed prompt and restores it appropriately. (Even if humans can't)
  - How much GPT-4 recovers depends on the compression ratio and whether it uses a sLLM.
  - Alpaca-7B restored the complete 9-step reasoning process, while GPT2-Alpaca only restored 7-step reasoning, with errors.

Original Prompt(9-steps Chain-of-Thought):
Question: Sam bought a dozen boxes, each with 30 highlighter pens inside, for $10 each box. He rearranged five of these boxes into packages of six highlighters each and sold them for $3 per package. He sold the rest of the highlighters separately at the rate of three pens for $2. How much profit did he make in total, in dollars?

Let's think step by step
Sam bought 12 boxes x $10 = $120 worth of highlighters.
He bought 12 * 30 = 360 highlighters in total.
Sam then took 5 boxes × 6 highlighters/box = 30 highlighters.
He sold these boxes for 5 * $3 = $15
After selling these 5 boxes there were 360 - 30 = 330 highlighters remaining.
These form 330 / 3 = 110 groups of three pens.
He sold each of these groups for $2 each, so made 110 * 2 = $220 from them.
In total, then, he earned $220 + $15 = $235.
Since his original cost was $120, he earned $235 - $120 = $115 in profit.
The answer is 115

Recovered Compressed Prompt(9-steps Chain-of-Thought):
Question: Sam purchased a dozen boxes, each containing 30 highlighter pens, for $10 each. He rearranged five of these boxes to have six pens each and sold them for $3 per box. He then sold the remaining highlighters individually in groups of three for $2 each. Let's calculate the total profit.

Initially, Sam bought 12 boxes with 30 highlighters each, which equals 360 highlighters. He spent $10 per box, so his total cost was $120.
Sam then rearranged 5 boxes to have 6 highlighters per box, which equals 30 highlighters. He sold these boxes for $3 each, making $15.
After selling these boxes, he had 330 highlighters left (360 - 30). He sold these in groups of three, which equals 110 groups. He sold each group for $2, making $220 from this.

In total, Sam made $235 ($15 + $220). Since his initial cost was $120, he earned a profit of $115 ($235 - $120).

The answer is $115.

5. Conclusion

Conclusion

- **Developed a coarse-to-fine algorithm for prompt compression in three modules called LLMLingua**
  - Budget Controller
  - Iterative Token-level Compression
  - Distribution Alignment

- **SOTA performance on all datasets. Only 1.5 point drop with 20x compression.**

- **LLMs are able to effectively restore the compressed prompts and reduce the length of their output.**

6. LongLLMLingua

2024-Winter Lab Seminar

LongLLMLingua
: Accelerating and Enhancing LLMs in Long Context Scenarios via Prompt Compression

Huiqiang Jiang, Qianhui Wu, Xufang Luo,
Dongsheng Li, Chin-Yew Lin, Yuqing Yang, Lili Qiu

Microsoft Research

Yejin Yoon

HYU NLP Lab., Hanyang University

stillwithyou@hanyang.ac.kr

JAN. 17. 2024

Framework of *LongLLMLingua*

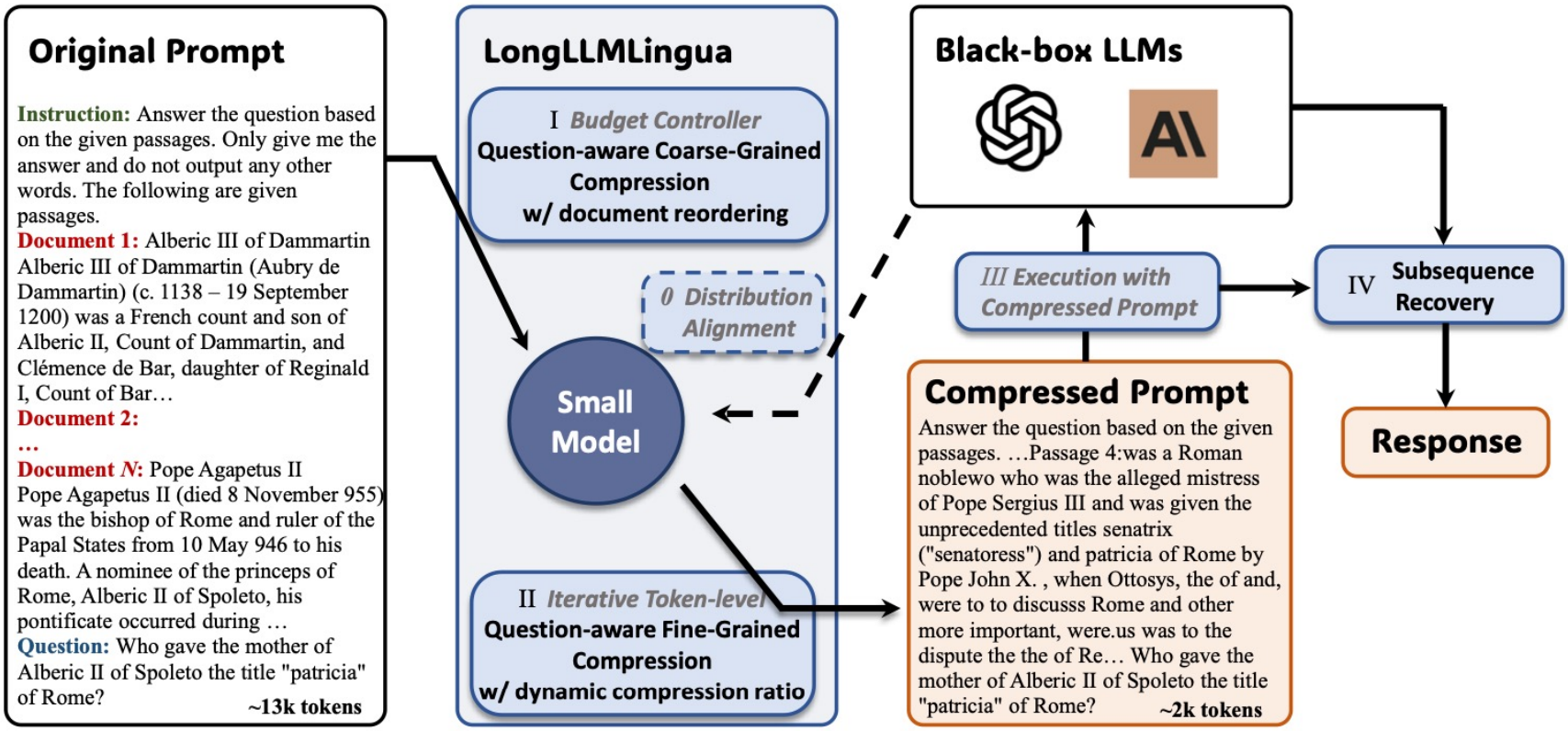
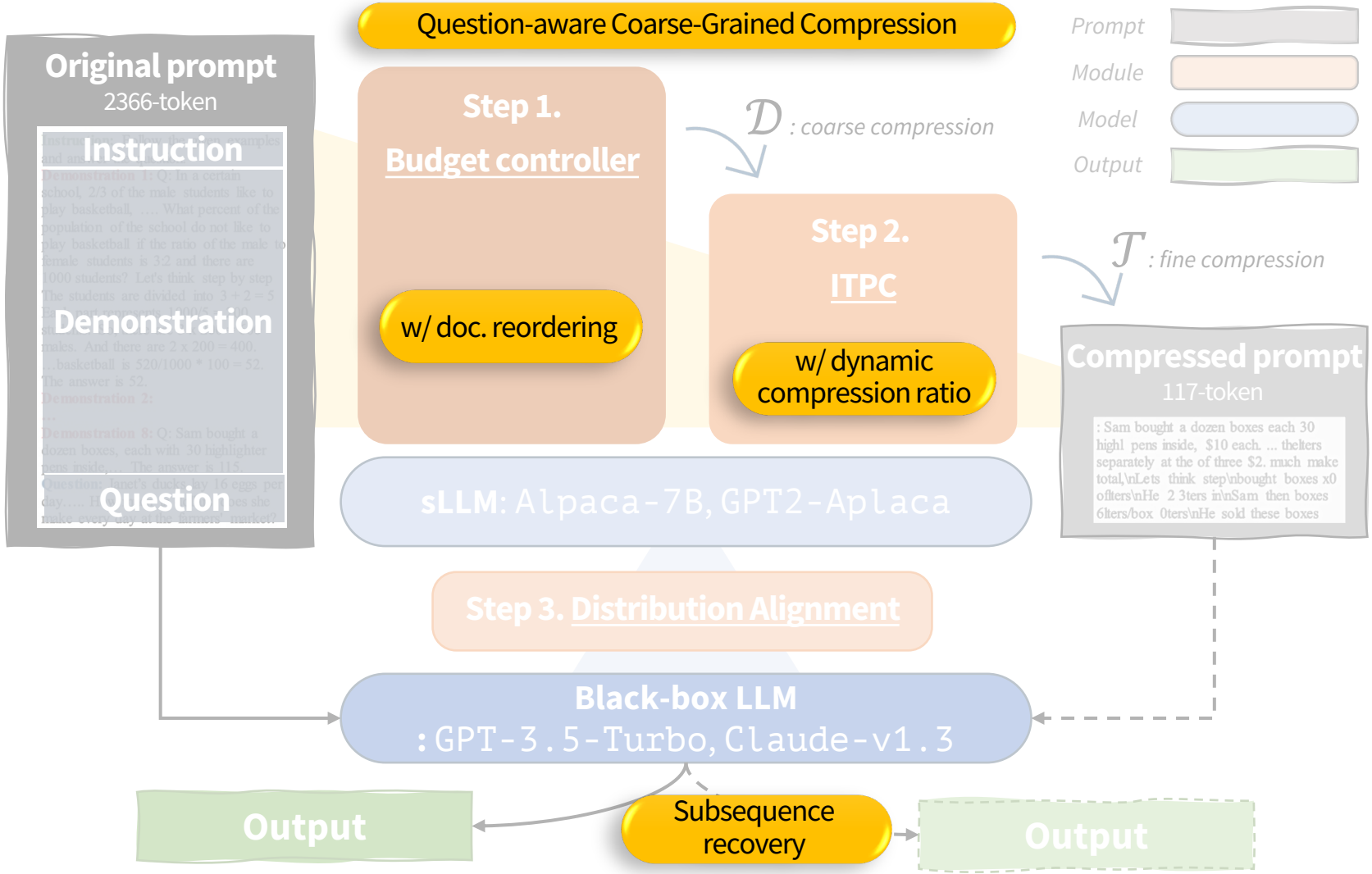
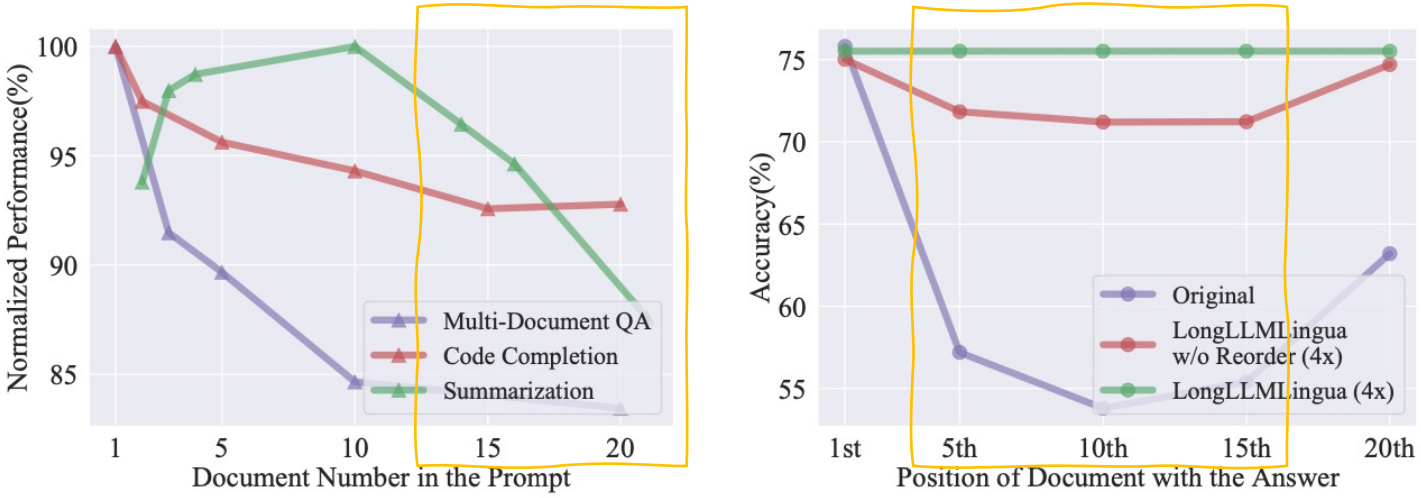


Figure 2: Framework of *LongLLMLingua*. Gray *Italic* content: As in LLMLingua.

Framework of *LongLLMLingua*



Intuition (document reordering + dynamic budget)

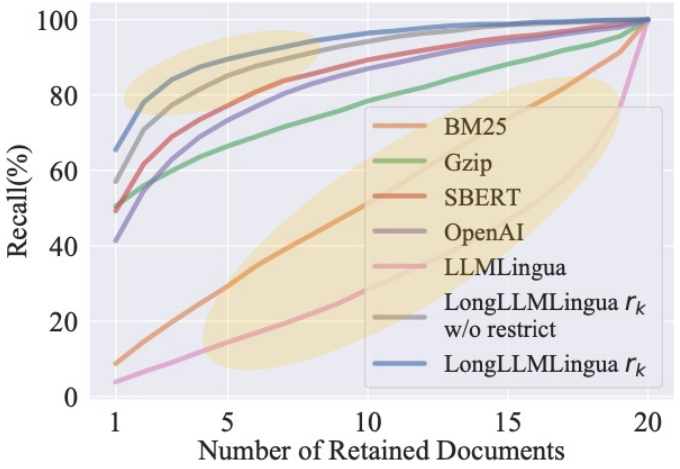


(a) Performance v.s. Document Number (b) Performance v.s. Key Information Position

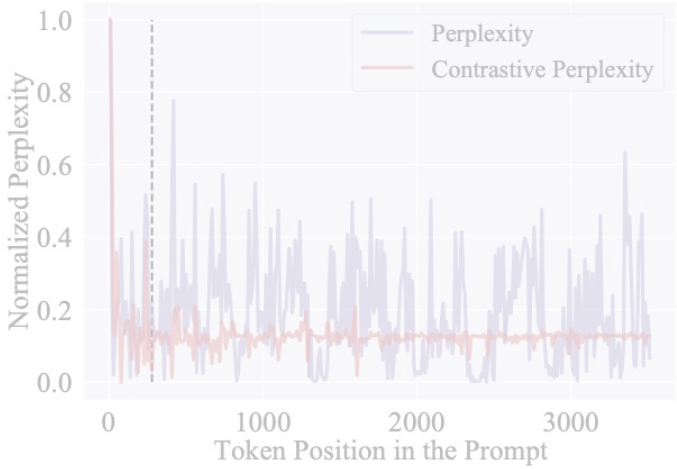
Figure 1: (a) LLMs' performance in downstream tasks may decrease as the noisy information in the prompt increases. In this case, we keep k most relevant documents/paragraphs based on the ground truth or LongLLMLingua r_k. A larger k implies more noise introduced into the prompt. To improve the key information density in the prompt, we present question-aware coarse-to-fine compression. (b) LLMs' ability to capture the relevant information depends on their positions in the prompt. To reduce information loss in the middle, we introduce a document reordering mechanism.

$$\tau_k^{\text{doc}} = \max(\min((1 - \frac{2I(r_k)}{N_d})\delta\tau + \tau^{\text{doc}}, 0), 1),$$

Intuition (query-aware)



(a) Recall Distribution



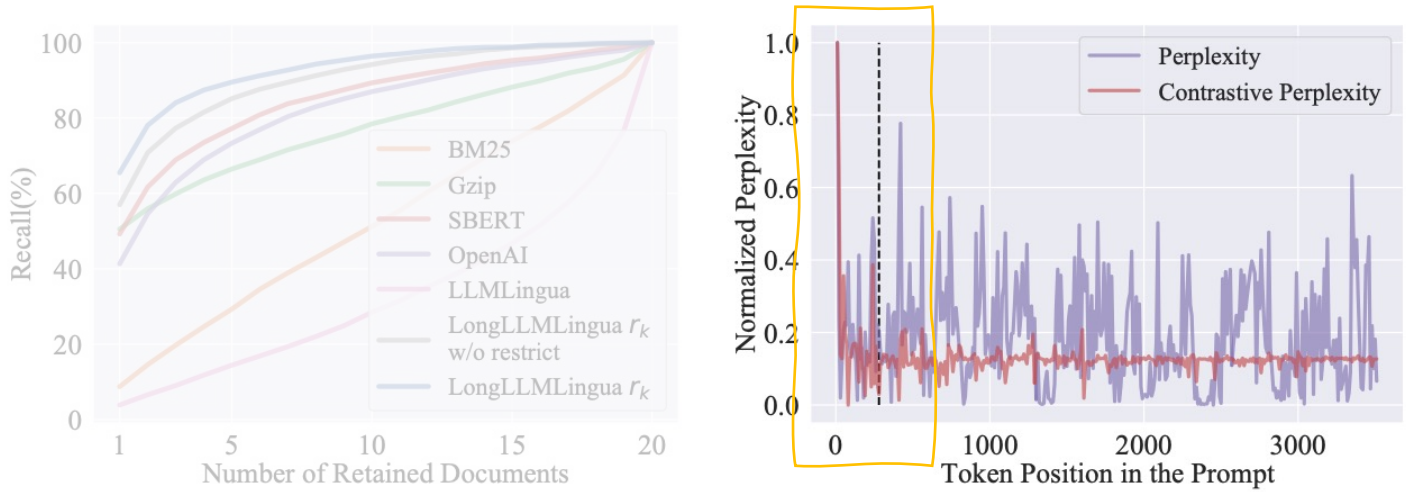
(b) Perplexity Distribution

Figure 3: (a) Comparison of recall on NaturalQuestions Multi-documemnt QA dataset. (b) Comparison between perplexities and contrastive perplexities of tokens in the prompt from Multi-documemnt QA dataset. The document with the ground truth is located on the left side of the dashed line.

$$r_k = \frac{1}{N_c} \sum_{i}^{N_c} p(x_i^{\text{que,restrict}} | \mathbf{x}_k^{\text{doc}}) \log p(x_i^{\text{que,restrict}} | \mathbf{x}_k^{\text{doc}}), k \in \{1, 2, \cdots, K\},$$

We can get the answer to this question in the given documents.

Intuition (query-aware)



(a) Recall Distribution

(b) Perplexity Distribution

Figure 3: (a) Comparison of recall on NaturalQuestions Multi-documemnt QA dataset. (b) Comparison between perplexities and contrastive perplexities of tokens in the prompt from Multi-documemnt QA dataset. The document with the ground truth is located on the left side of the dashed line.

$$s_i = \text{perplexity}(x_i|x_{<i}) - \text{perplexity}(x_i|x^{\text{que}}, x_{<i}).$$

↳ Contrastive perplexity

Intuition (subsequence recovery)

Original Prompt	Compressed Prompt	LLMs' Response
Document [1](Title: List of Nobel laureates in Physics) The first Nobel Prize in Physics was awarded in 1901 to {Wilhelm Conrad Röntgen}{Wilhelm Con rad Rö nt gen}, of Germany,...	Document [1](Title: List of Nobelates in Physics) The first Nobel1 {Wilhelmgen}{Wilhelm gen}, of, who received, ...	{Wilhelmgen} {Wilhelm gen}

Figure 4: The example of Subsequence Recovery, the red text represents the original text, and the blue text is the result after using the LLaMA 2-7B tokenizer.

- **Harry ~~Styles~~ vs. Harry ~~Poter~~**

1. Identify the longest matching substring in LLM's output

2. Find a subsequence of the original prompt and swap

- Prefix tree or sequence automata

{ End Page }

Thank you :D

Yejin Yoon

HYU NLP Lab.

Dept. of Artificial Intelligence Application,
Hanyang University

stillwithyou@hanyang.ac.kr