From Reading to Compressing: Exploring the Multi-document Reader for Prompt Compression

Eunseong Choi, Sunkyung Lee, Minjin Choi, June Park, Jongwuk Lee

Sungkyunkwan University (SKKU), Republic of Korea

Introduction

Prompt Compression

Token Pruning

- Challenges

Multi-document Reader as a Compressor

Key Contributions

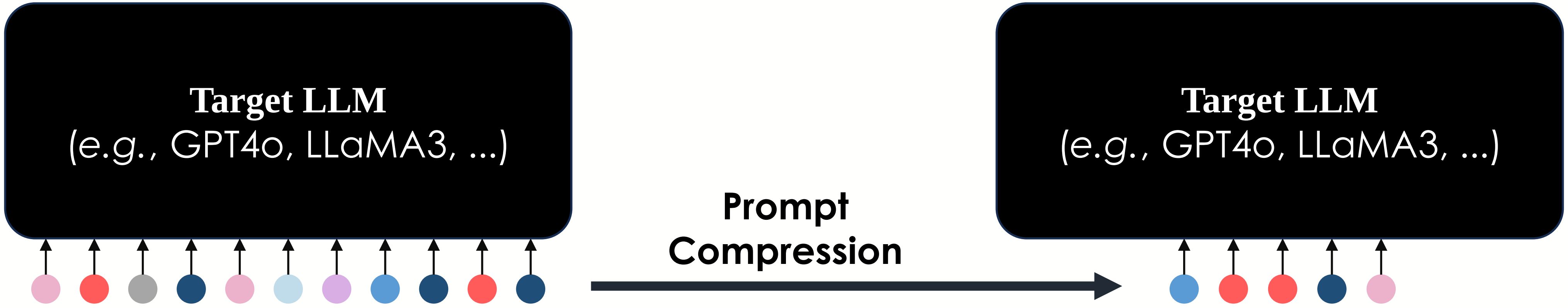
Prompt Compression

- **What is the prompt compression and why do we need it?**
  - Prompt compression aims to **reduce the computational overhead in LLMs, while preserving the essential information in the prompt**.

Prompt Compression

➢ **What is the prompt compression and why do we need it?**

- Prompt compression aims to reduce the computational overhead in LLMs, while preserving the essential information in the prompt.
- Basically, **it reduces the tokens to be input to the LLMs**.

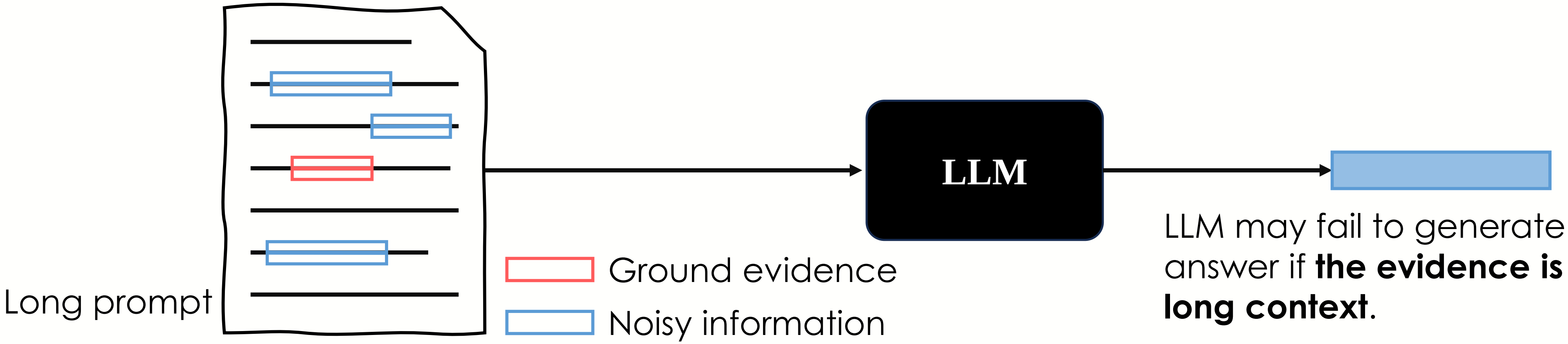


Prompt Compression

➢ **What is the prompt compression and why do we need it?**

1. Long contexts lead to high computational cost.
   - inference delays
   - API cost

2. Irrelevant information in long contexts can degrade the performance of LLMs.
   - Lost-in-the-Middle problem[1)]

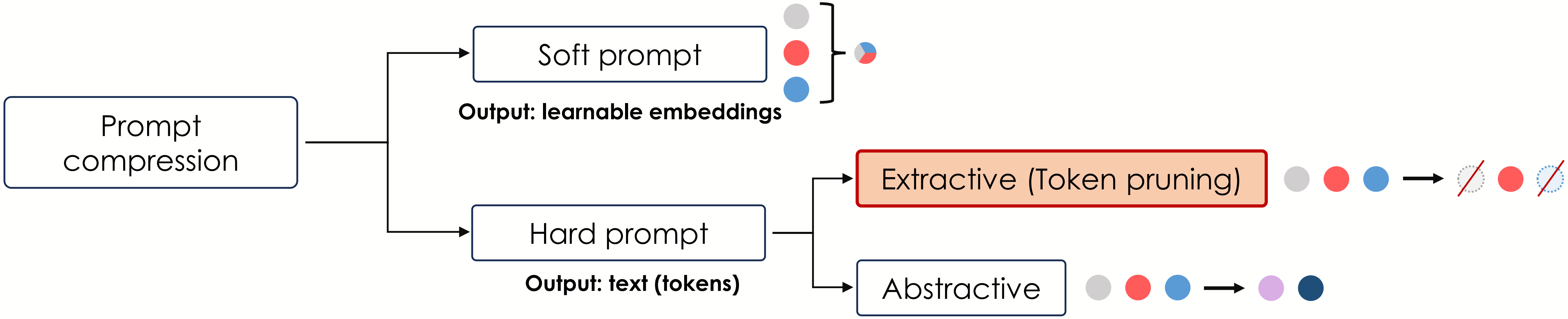


LLM may fail to generate the correct answer if **the evidence is hidden in the long context**.

1) Nelson F. Liu et al. "*Lost in the Middle: How Language Models Use Long Contexts.*" TACL 2023

Token Pruning for Prompt Compression

➢ **Taxonomy of Prompt Compression**

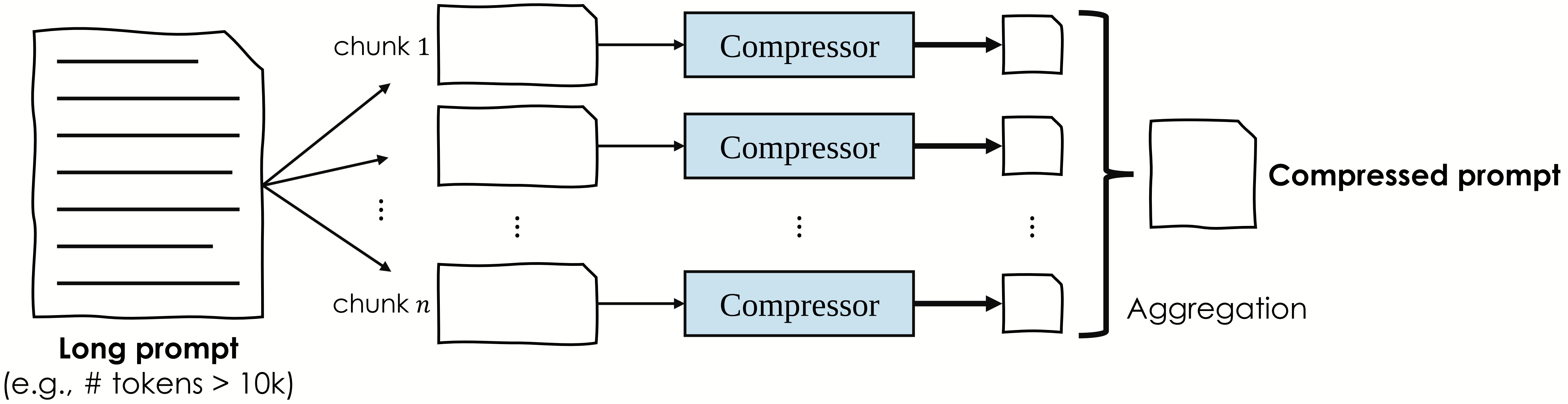


➢ **We aim to improve extractive prompt compression, *i.e.*, token pruning, which is**

- easily adaptable to any black-box LLMs: no need to train an LLM, even adapters for PEFT.
- more efficient than abstractive compression, where auto-regressive generation is required.

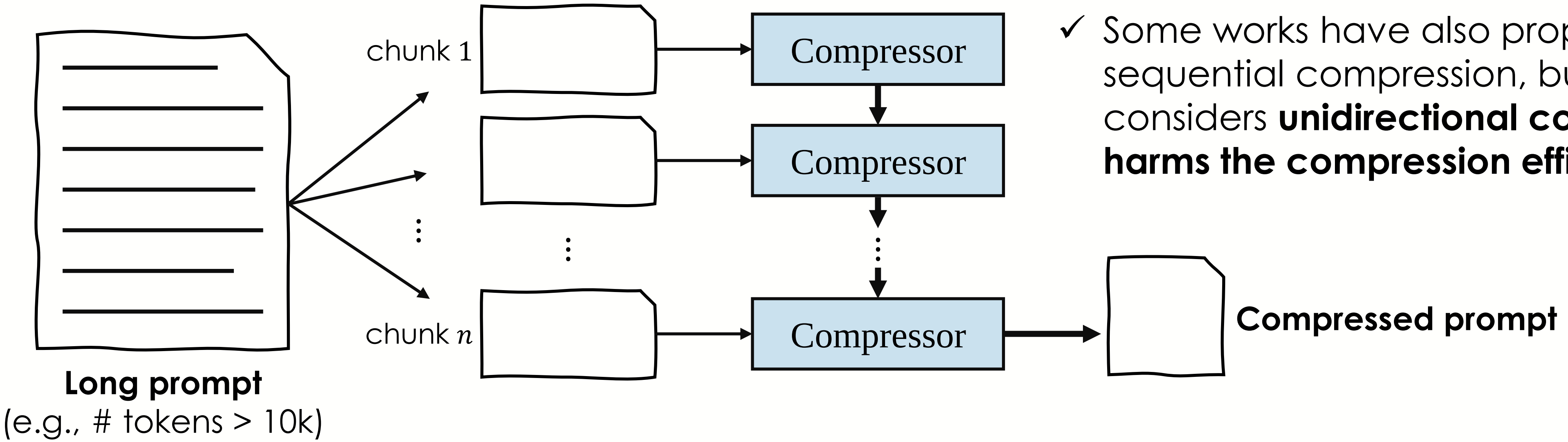
Challenges in Token Pruning ①

1. **How to extract essential information based on the global context?**
   - Existing work processes chunk by chunk, neglecting the global context.



Challenges in Token Pruning ①

1. **How to extract essential information based on the global context?**
   - Existing work processes chunk by chunk, neglecting the global context

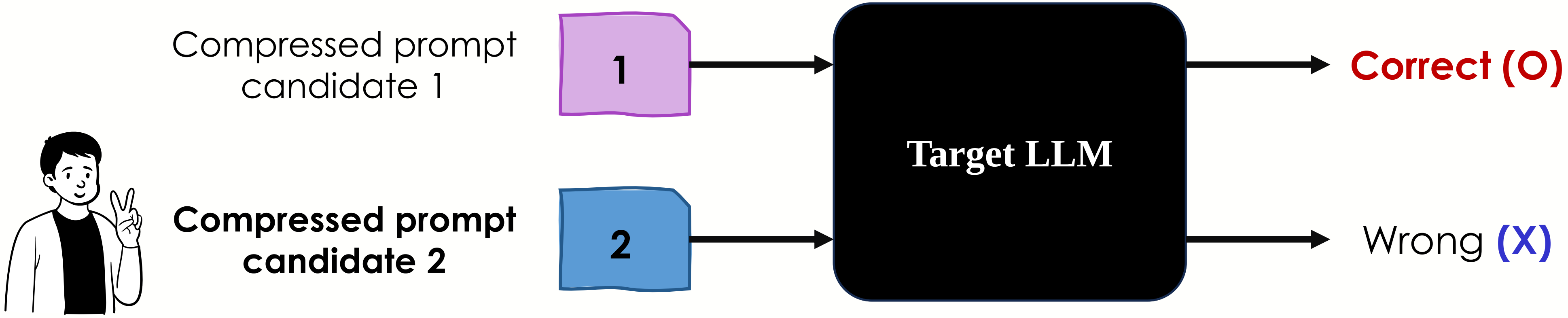


✓ Some works have also proposed sequential compression, but it only considers **unidirectional context** and **harms the compression efficiency**.

Challenges in Token Pruning ②

2. How to train a compressor effectively?

- The ground truth of the compressed prompt is difficult to define.

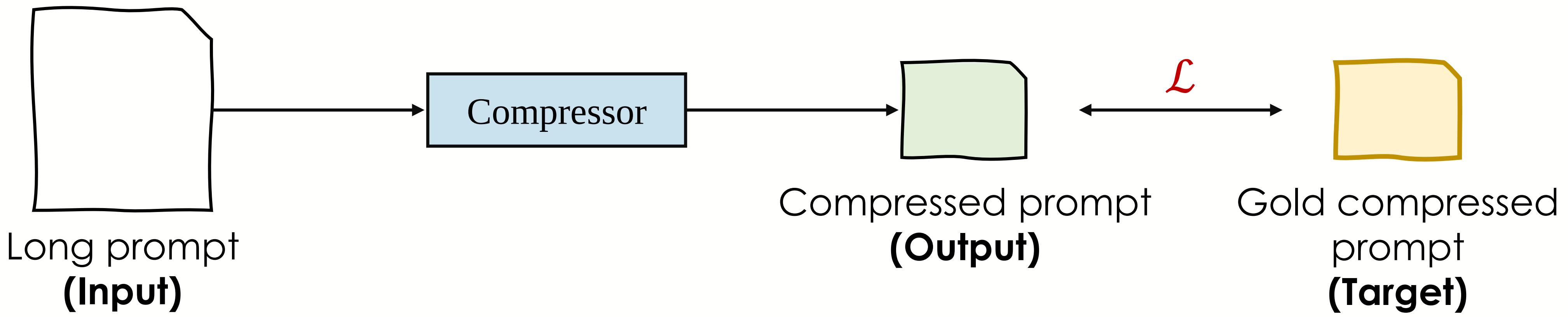


It is hard to know which compressed prompt is **the best for the target LLM.**

Challenges in Token Pruning ②

2. How to train a compressor effectively?

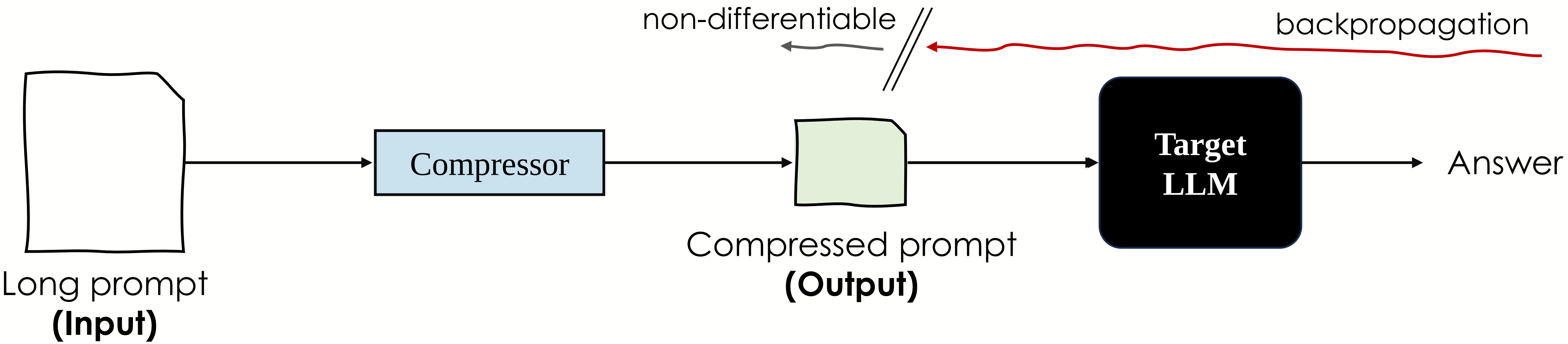
- The ground truth of the compressed prompt is difficult to define.



Challenges in Token Pruning ②

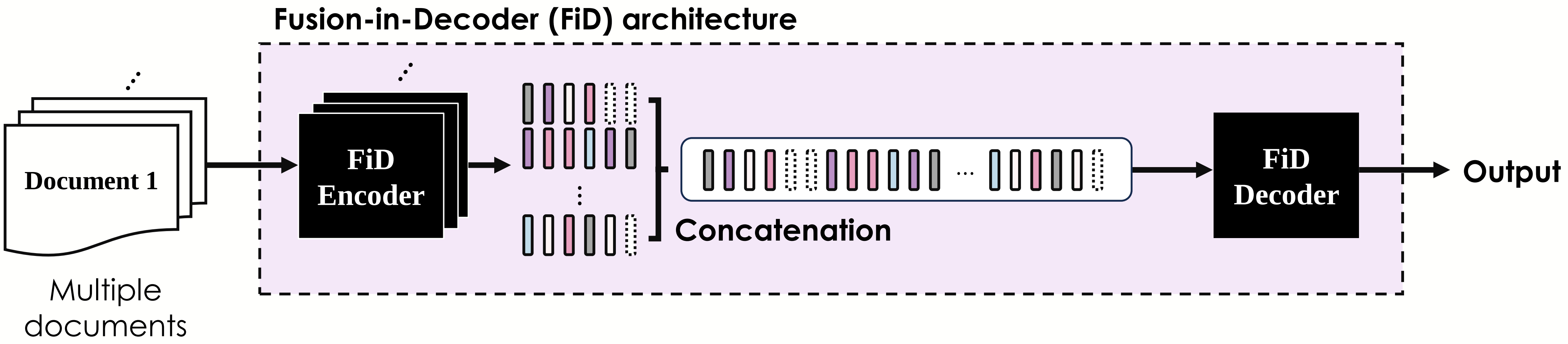
2. How to train a compressor effectively?

- The ground truth of the compressed prompt is difficult to define.



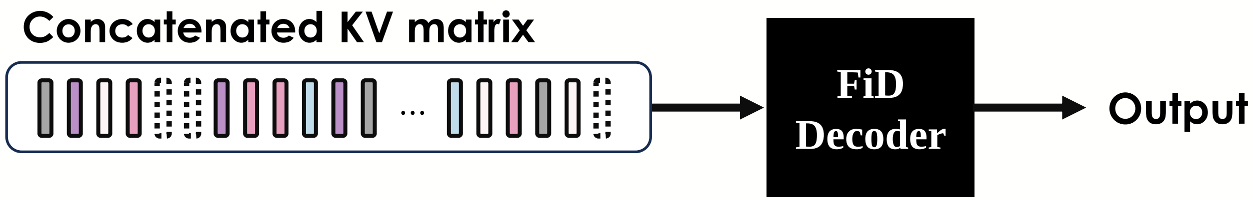
Multi-document Reader as a Compressor

- **We employ a multi-document reader, *i.e.*, Fusion-in-Decoder architecture, to address the challenges.**
  1) Aggregation of evidence across multiple documents
  2) Efficient processing of multiple documents using the cross-attention mechanism

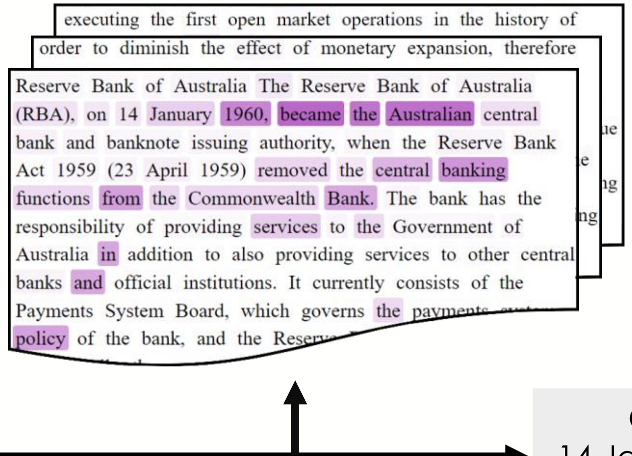


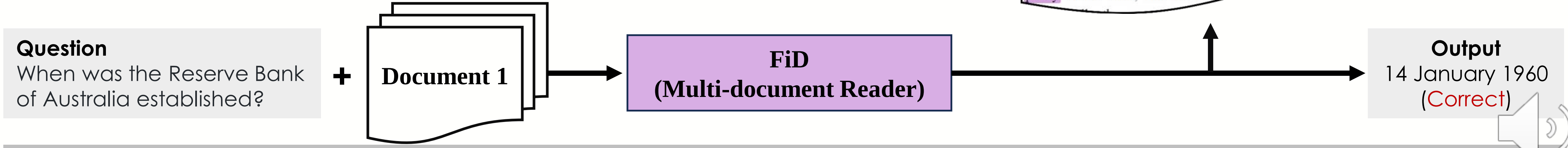
Multi-document Reader as a Compressor

- Addressing challenge ① **capturing contextual dependencies across the long context.**
  - FiD can capture global context **by using the concatenated KV matrix** during the decoding.



- Addressing challenge ② **absence of ground truth**
  - **By learning to answer the question**, *i.e.*, QA task, FiD learns to capture core information.
  - Avoiding the need for pseudo compressed prompts





Key Contributions

1. We introduce the prompt compression connected with the FiD to **capture the global semantics** over chunks aggregated in the decoder.

2. We **align the question-answering task with prompt compression**.
    - The cross-attention scores trained to answer the question provide the effective **approximation of the tokens that the target LLM focuses on**.

3. R2C achieves efficiency by using the cross-attention scores computed in generating only the first token, avoiding auto-regressive generation.

Proposed Method

Overall Framework

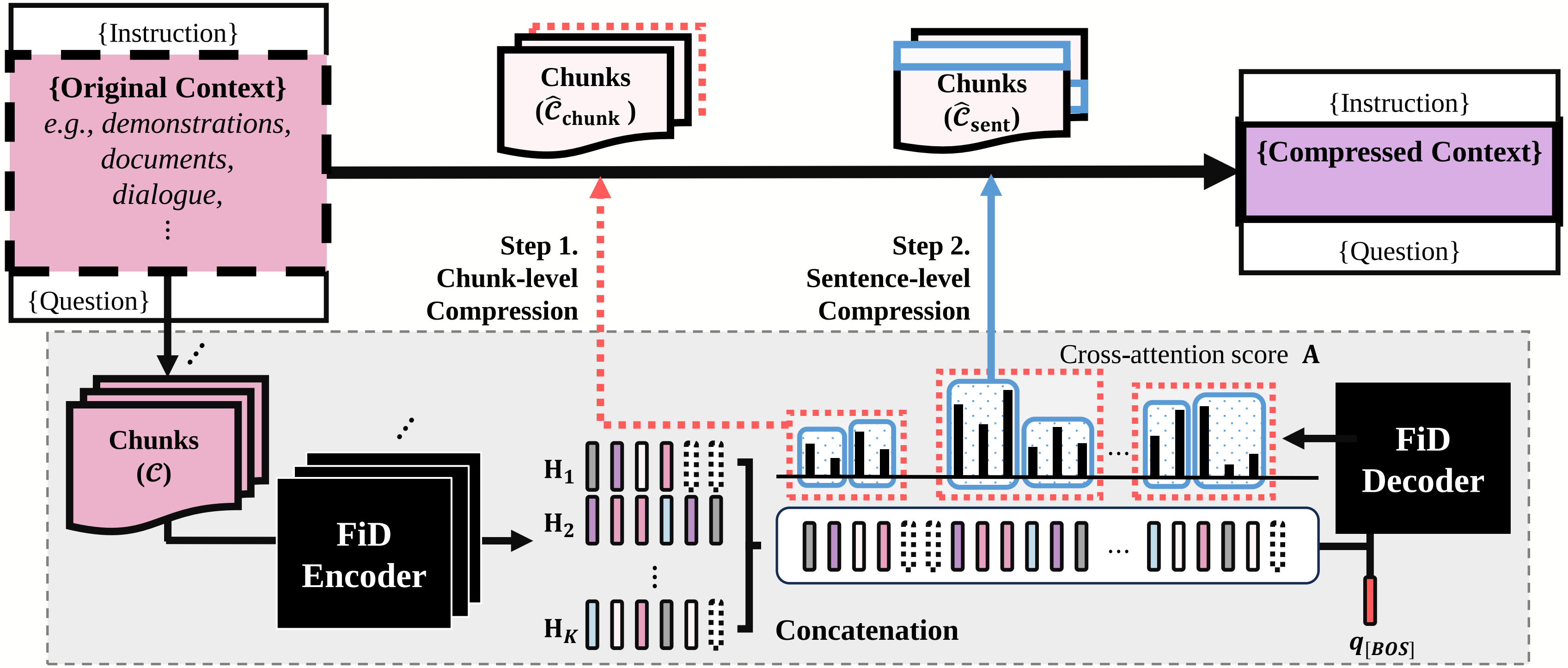
Identifying Importance in Context

Aggregating Unit Importance

Hierarchical Compression

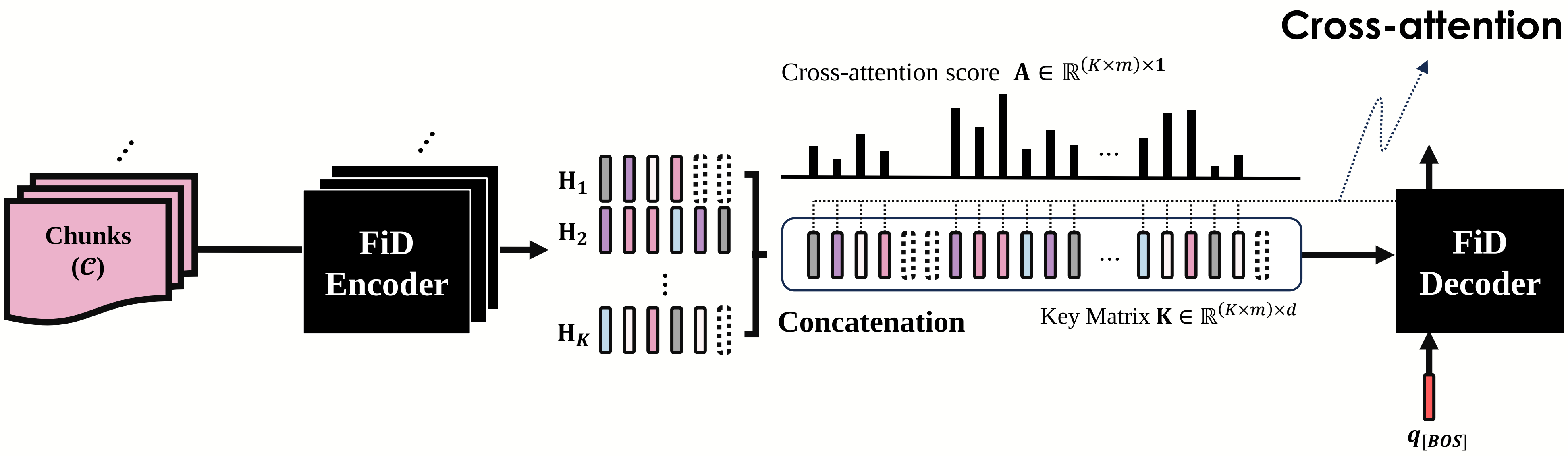
Overall Framework

- **Reading to Compressing (R2C)**
  - Cross-attention scores over concatenated multiple chunks → global context
  - Coarse-to-fine compression
  - Training: QA task



Identifying Importance in Context

- **R2C first divides long context into multiple chunks and puts them into the FiD-encoder.**
- **By summing the cross-attention scores over all layers and heads in the FiD-decoder, R2C gets the token-level importance.**
  - Token-level importance of j-th token in i-th chunk $t_{i,j} = \sum_{l=1}^{L} \sum_{h=1}^{H} \mathbf{A}_{i,j}^{(l,h)}$.



Aggregating Unit Importance

- **Token-level compression may neglect the semantic integrity of the text.**

She moved to Los Angeles, where she studied drama at the Lee Strasberg Theatre and Film Institute

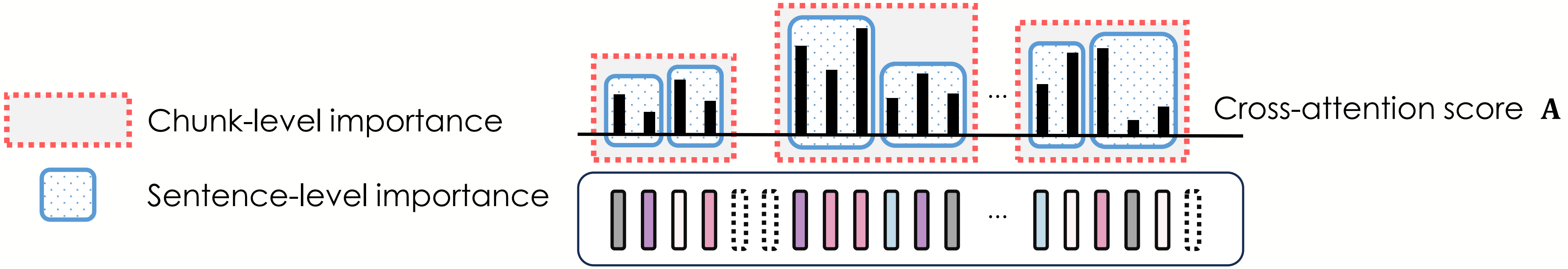
→ **Token-level compression**

Los Angeles studied drama at the Lee Strasberg Theatre and Film Institute

Image by GPT4o

Aggregating Unit Importance

- **Instead, R2C adopts two coarser granularity compression units, chunks and sentences.**
  - R2C averages the token importance to compute the chunk and sentence importance.



Hierarchical Compression

- **R2C compresses the prompt hierarchically**, given the multi-granularity unit importance, *i.e.*, chunk and sentence.
  - Target number of tokens after compression T
  - Hierarchical ratio between two levels of compression ρ
  - → Number of tokens to compress $E_{\mathrm{comp}} = |\mathrm{P_C}| - T$, where $|\mathrm{P_C}|$ is the original length.

  - R2C **iteratively removes chunks or sentences** until the number of compressing tokens meets the E_{chunk} and E_{sent}.
    1. Chunk-level compression
       - $E_{\mathrm{chunk}} = \rho \cdot E_{\mathrm{comp}}$
    2. Sentence-level compression
       - $E_{\mathrm{sent}} = (1 - \rho) \cdot E_{\mathrm{comp}}$
       - For the i-th chunk, R2C multiplies the normalized inverse chunk-level importance to $E_{sent,i}$, to reserve more information in salient chunks, reflecting the global context.

Experiments

Experimental Setup

Baselines

Experimental Results

Compression Efficiency

Experimental Setup

➢ **We validate the compression performance of R2C through in- and extensive out-of-domain evaluation.**

1) In-domain evaluation

Dataset	Task	Source	Average length	Metric	# samples (train/valid/test)
Natural Questions	Single-document QA	Wikipedia	3,018 (20 candidate passages)	Span EM	79,168 / 8,757 / 3,610

2) Out-of-domain evaluation
   - LongBench benchmark with 5 tasks and 15 different datasets.
   - Tasks
     - Single-document QA (SingleDoc): NarrativeQA, Qasper, MultiFieldQA-en
     - Multi-document QA (MultiDoc): HoppotQA, 2WikiMultihopQA, MuSiQue
     - Summarization (Summ.): GovReport, QMSum, MultiNews
     - Few-shot Learning (FewShot): TREC, TriviaQA, SAMSum
     - Code Completion (Code): LCC, RepoBench-P

Experimental Setup

- **Target LLM – We use one relatively small LLM and a powerful API-based LLM to validate the compressed prompts.**
  1) Llama2-7b-chat-hf (LLaMA2-7B)[1)]
  2) GPT-3.5-turbo-1106 (GPT-3.5)[2)]

- **Details**
  1) **Retriever** (for NQ dataset): DPR[3)]
  2) **Backbone**: FiD-base trained on the NQ, using 20 passages for each question
  3) **Compression hyperparameters** – randomly sampled 20% of the NQ dev set for tuning
     - Hierarchical ratio ρ: 0.8 , *i.e.*, 80% of the compression is done at the chunk-level
     - Target tokens T: 500 for NQ (6x compression), 2k for LongBench (5x compression)
  4) **Chunk size**: 128 tokens

1) Hugo Touvron et al. "Llama 2: Open foundation and fine-tuned chat models"
2) https://chatgpt.com/
3) Gautier Izacard, Edouard Grave. "Distilling Knowledge from Reader to Retriever for Question Answering." ICAL 2021

Baselines

- **Two retrieval-based models** (chunk-level compression only)
  - We follow the settings from LongLLMLingua, where instructions are used as questions if there the question does not exist in the dataset.

  1) BM25
  2) DPR
     - We use DPR trained with knowledge distillation on the NQ dataset

- **Five compression-based models**
  1) Selective-Context
  2) LLMLingua
  3) LongLLMLingua
  4) LLMLingua-2
  5) RECOMP

In-domain Evaluation

- **Accuracy of FiD and two target LLMs on the Natural Questions (NQ) test set.**
  1) R2C achieves the **best performance among existing compression methods**.
     - Improvements of 5.6% and 11.1% over the most effective baseline.

Target LLM	Compression	NQ test (Span EM)	# tokens
FiD	-	50.5	-
GPT-3.5	Original	66.7	3018.0
	BM25	49.4	534.0
	DPR	<u>63.0</u>	501.0
	Selective-Context	44.4	501.0
	LLMLingua	41.9	478.0
	LLMLingua-2	52.1	510.0
	LongLLMLingua	55.6	489.0
	RECOMP	<u>63.0</u>	500.0
	R2C (ours)	**66.5**	482.0
LLaMA2-7B	Original	52.8	3018.0
	BM25	41.8	534.0
	DPR	<u>54.3</u>	501.0
	Selective-Context	38.1	501.0
	LLMLingua	32.7	478.0
	LLMLingua-2	42.5	510.0
	LongLLMLingua	49.0	489.0
	RECOMP	53.7	500.0
	R2C (ours)	**59.7**	48[illegible]

In-domain Evaluation

- **Accuracy of FiD and two target LLMs on the Natural Questions (NQ) test set.**

1) R2C achieves the best performance among existing compression methods.
   - Improvements of 5.6% and 11.1% over the most effective baseline.

2) Among the models that use NQ for training, *i.e.*, DPR, RECOMP, R2C, R2C shows better performance.
   - **Learning to answer the question directly contributes to capture importance context.**
   - A benefit of using **cross-attention scores**

Target LLM	Compression	NQ test (Span EM)	# tokens
FiD	-	50.5	-
GPT-3.5	Original	66.7	3018.0
	BM25	49.4	534.0
	DPR	63.0	501.0
	Selective-Context	44.4	501.0
	LLMLingua	41.9	478.0
	LLMLingua-2	52.1	510.0
	LongLLMLingua	55.6	489.0
	RECOMP	63.0	500.0
	R2C (ours)	**66.5**	482.0
LLaMA2-7B	Original	52.8	3018.0
	BM25	41.8	534.0
	DPR	54.3	501.0
	Selective-Context	38.1	501.0
	LLMLingua	32.7	478.0
	LLMLingua-2	42.5	510.0
	LongLLMLingua	49.0	489.0
	RECOMP	53.7	500.0
	R2C (ours)	**59.7**	48[illegible]

Out-of-domain Evaluation

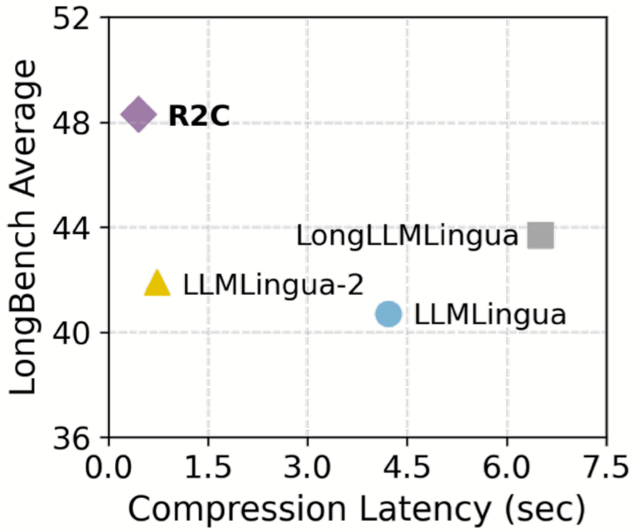
- **Out-of-domain evaluation validates that the QA task proves to be an effective alternative for training compressor.**
  - R2C shows superior performance on all tasks including 2 QA tasks.
  - BM25 with chunk-level filtering outperforms R2C, indicating the need for different levels of granularity for effective compression depending on the prompts.

Target LLM	Compression	SingleDoc	MultiDoc	Summ.	FewShot	Code	Average	# tokens
GPT-3.5	Original	43.2	46.1	25.2	69.2	64.4	49.6	9,881
	BM25	34.9	41.0	23.3	**68.1**	49.6	43.4	1,949
	Selective-Context	30.4	31.4	20.9	66.0	55.0	40.7	1,830
	LLMLingua	29.8	35.4	22.1	52.4	45.0	36.9	2,009
	LLMLingua-2	36.2	40.9	23.2	61.5	47.6	41.9	2,023
	LongLLMLingua	37.0	44.9	22.0	65.1	49.4	43.7	1,743
	RECOMP	40.1	48.1	-	-	-	-	-
	R2C (ours)	**43.5**	**48.7**	**24.9**	66.9	**57.6**	**48.3**	1,976

Compression Efficiency

- **R2C dramatically improves compression efficiency, while enhancing the accuracy.**
  - Although R2C uses a generative model, it only uses the cross-attention scores from the first token.

Model	Backbone	# parameters
LLMLingua	LLaMA2-7B	7B
LongLLMLingua	LLaMA2-7B	7B
LLMLingua-2	XLM-RoBERTa-large	355M
R2C	T5-base	220M



Compression Efficiency

- **The overall end-to-end inference time can also be accelerated with R2C.**

Compression	Compression latency	API latency	End-to-end latency
-	0s	1.52s	1.52s (100.0%)
R2C (5x)	0.45s	0.88s	1.33s (87.5%)
R2C (10x)	0.44s	0.68s	**1.11s (74.0%)**

Conclusion

Conclusion

- We propose ***R**eading **To C**ompressing* (**R2C**), a novel prompt compression method that uses Fusion-in-Decoder to **capture global context across multiple chunks**.

- R2C trained on question-answering datasets, **identifies key tokens without noisy pseudo-labels**.

- We extensively validate R2C on both in- and out-of-domain evaluations and show that it outperforms existing methods by preserving semantic integrity.

Thank you

E-mail: eunseong@skku.edu

Paper: https://arxiv.org/abs/2410.04139

Code: https://github.com/eunseongc/R2C