Prompt Compression with Context-Aware Sentence Encoding for Fast and Improved LLM Inference

Barys Liskavets[1*], **Maxim Ushakov**[1], **Shuvendu Roy**[2], **Mark Klibanov**[3], **Ali Etemad**[2], **Shane K. Luke**[3]

[1]Alterra AI, Palo Alto, United States
[2]Queen's University, Canada
[3]Workday Inc.

Abstract

Large language models (LLMs) have triggered a new stream of research focusing on compressing the context length to reduce the computational cost while ensuring the retention of helpful information for LLMs to answer the given question. Token-based removal methods are one of the most prominent approaches in this direction, but risk losing the semantics of the context caused by intermediate token removal, especially under high compression ratios, while also facing challenges in computational efficiency. In this work, we propose context-aware prompt compression (CPC), a sentence-level prompt compression technique where its key innovation is a novel context-aware sentence encoder that provides a relevance score for each sentence for a given question. To train this encoder, we generate a new dataset consisting of questions, positives, and negative pairs where positives are sentences relevant to the question, while negatives are irrelevant context sentences. We train the encoder in a contrastive setup to learn context-aware sentence representations. Our method considerably outperforms prior works on prompt compression on benchmark datasets and is up to 10.93× faster at inference compared to the best token-level compression method. We also find better improvement for shorter length constraints in most benchmarks, showing the effectiveness of our proposed solution in the compression of relevant information in a shorter context. Finally, we release the code and the dataset for quick reproducibility and further development.

Code — https://github.com/Workday/cpc

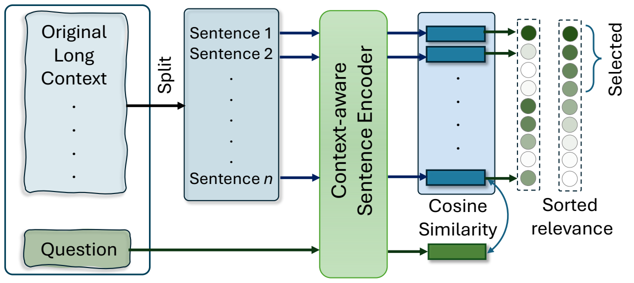


Figure 1: Overview of CPC. We first train a novel **context-aware** sentence encoder contrastively on a newly generated dataset of questions, positives, and negative pairs. Our model then uses this context-aware encoder to generate embeddings of the question and context sentences, and uses the similarity between them to select relevant sentences.

Method	Performance	Latency
LLMLingua	37.4	9.8×
LLMLingua-2	42.2	0.67×
LongLLMLingua	48.8	10.93×
CPC	**50.0**	1×

Table 1: Overall comparison of our method on performance and latency on LongBench. CPC outperforms LongLLMLingua (SOTA) on both performance and latency.

1 Introduction

The advent of large language models (LLMs) has triggered a surge of research into prompting techniques, including chain-of-thought (Wei et al. 2022), in-context learning (Dong et al. 2022), and retrieval augmented generation (Lewis et al. 2020), aimed at leveraging their generalization and reasoning capabilities for various downstream applications. In practice, providing suitable and descriptive (often long) prompts is essential for LLMs to generate useful responses for domain-specific tasks. However, longer prompts come at a significant inference expense for LLMs in terms of both time and cost. Therefore, striking a balance between the quality and the length of the prompt is a timely area of research in order to optimize inference performance vs. cost.

Recently, prompt compression has shown promise as a solution to the context length problem. The basic idea of such a method is to reduce the size of the prompt by removing less informative content. For instance, LLMLingua (Jiang et al. 2023b) proposed to utilize a well-trained language model to identify and remove non-essential tokens from the prompt. LongLLMLingua (Jiang et al. 2023c) enhanced LLMLingua for longer context by enabling question-aware compression. Later, LLMLingua-2 (Pan et al. 2024) proposed to formulate prompt compression as a token-level binary classification task that learns a task-agnostic prompt compression model by removing less important tokens. However, such token removal methods may result in non-coherent sentences, often

[*]Corresponding Author. Email: liskovets.borets@gmail.com

hampering the semantics of the prompt, especially when the compression ratio is relatively high. This results in a drop in the performance of the LLMs in answering questions. Additionally, most existing compression methods are usually computationally expensive as the inference time is proportional to the token length.

In this work, we propose a sentence-level compression technique called Context-aware Prompt Compression (CPC) that compresses the prompt by removing the sentences that are less relevant to the given question. A key innovation of our proposed solution is a context-aware sentence encoder that is used to rank all the sentences in the context based on their relevance to the question. Here, the relevance is measured as the (context-aware) embedding similarity (cosine distance) between the question and each sentence in the context. An important characteristic of our method is that it performs context compression while preserving human readability, unlike token-based compressors such as the LLMLingua family (Pan et al. 2024; Jiang et al. 2023c,b). We train the context-aware sentence encoder by learning to distinguish between positive and negative sentences, where the positives are context sentences that contain relevant information to the given question, and the negatives are context sentences that do not contain any relevant information to the question. Along with the proposed method, we also introduce a new dataset with question, positive, and negative sentence pairs required to train our context-aware sentence encoder. Figure 1 provides an overview of our method.

We evaluate our model following the protocol established by prior works on prompt compression (Pan et al. 2024; Jiang et al. 2023c) on LongBench (Bai et al. 2023) and ZeroSCROLLS (Shaham et al. 2023). Extensive experiments show that our proposed method outperforms the existing state-of-the-art (LongLLMLingua (Jiang et al. 2023c)) on these two benchmarks by 1.5% and 1.3% on average on the 2,000 tokens constraint. Similarly, on the 3,000 token constraint, our method outperforms others by 1.2% and 2.1%, respectively. On the individual sub-tasks of LongBench, our method shows up to 8.3% improvements. We present detailed ablation studies on different components of our proposed solution, including the dataset collection and the context-aware sentence embedding module. Furthermore, our method is up to 10.93× faster than LongLLMLingua during inference, imposing minor overhead during prompt compression (see Table 5). In summary, our contributions are:

- We propose a new method for sentence-level prompt compression. Our method relies on the novel concept of context-aware sentence encoding, which enables it to compress prompts by removing sentences that do not contain relevant information for the given question.
- As part of our solution, we curate and release a new dataset that contains tuples of context, question, positive, and negatives to train the context-aware sentence encoder.
- Our method outperforms prior works and sets a new state-of-the-art for prompt compression on LongBench and ZeroSCROLLS. Additionally, our method is up to 10.93× faster during inference compared to existing prompt compressors. To enable reproducibility and contribute to the area, we release our code and dataset at: https://github.com/Workday/cpc.

2 Related Work

In this section, we discuss the literature relevant to our work from two perspectives. First, we review previous works on prompt compression. Next, we provide an overview of the literature on sentence embedding learning, focusing on aspects relevant to our context-aware embedding module.

2.1 Prompt Compression

Recent work on prompt compression aims to reduce the inference cost of LLMs. A prominent direction in this area is model-agnostic prompt compression, which utilizes a pre-trained model to act as an information-based compressor. For instance, Kim et al. (2022) introduced a token pruning method in the forward pass of the LLM. In Chen et al. (2023), a heuristic method was employed to recursively summarize the context. A limitation of these early works is that they require access to the pre-trained LLM as part of the compression method, which is not often feasible. Some of the more recent works developed solutions that do not require the LLM for compression. The most notable method in this direction is LLMLingua (Jiang et al. 2023b), which uses token-level perplexities to filter out semantically insignificant tokens that do not have high perplexity. Similar efforts have been performed by Selective-Context (Li et al. 2023a), with decoder-only models (LLaMa (Touvron et al. 2023), GPT-2 (Radford et al. 2019)) for token-wise perplexity calculation, which keeps a pre-defined number of highest-perplexity tokens to achieve the required compression. LongLLMLingua (Jiang et al. 2023c) further developed this idea for question-aware context compression by adding document-level question relevance estimation prior to employing LLMLingua. In general, these methods suffer from the lack of capability to adapt to new domains.

Another common direction of prompt compression is trainable prompt compression methods, including soft prompt methods, sequence-to-sequence methods, and reinforcement learning methods, among others. Soft prompt methods directly pre-train or fine-tune a language model as the compressor (Wang et al. 2024; Bulatov, Kuratov, and Burtsev 2022; Chevalier et al. 2023; Ge et al. 2024; Wang and Xiao 2024), which usually yield a high compression rate but have little interpretability or control over the compression rate. Sequence-to-sequence models capture the entire context as input and directly generate the compressed sentence (Xu, Shi, and Choi 2024). However, these methods have high latency due to the autoregressive nature of generation. Reinforcement learning for prompt compression, such as in Laban et al. (2021), introduces a novel reward function to simplify complex text by jointly optimizing simplicity, fluency, salience, and guardrails. Another approach utilizes compression ratio as a reward function, conditioned on meeting a ROUGE metric threshold (Jung and Kim 2024), but may compromise on question-aware context compression tasks. Furthermore, reinforcement learning has been employed for token classification in efficient unsupervised sentence compression by fine-tuning pre-trained encoder (Ghalandari, Hokamp, and Ifrim 2022). Among more recent methods, Pan et al. (2024) trains a model that is designed to evaluate the information value of each specific lexical unit, which is then sorted by these values and pruned. However, the token-level compression results in a non-coherent sentence that compromises the semantics of the compressed prompt, resulting in a sub-optimal performance of the LLM. In this work, we focused on prompt compression by identifying and removing less relevant *sentences* from an input context.

2.2 Text Embedding Learning

The goal of text embedding learning is to generate text embeddings that capture the semantic meaning in a high-dimensional vector space, enabling various natural language processing tasks, such as text classification, clustering and similarity search. Earlier works in text embedding learning, such as GloVe (Pennington, Socher, and Manning 2014) and Word2Vec (Church 2017), focused on learning a word or token-level representation, while more recent works have explored sentence-level representation learning. For instance, Reimers and Gurevych (2019) proposed to fine-tune BERT-like architecture (Vaswani et al. 2017) for extraction of sentence embeddings that can be subsequently utilized to measure text similarities. Later Li et al. (2023b); Wang et al. (2022); Beltagy, Peters, and Cohan (2020) further improved the efficacy of the text embedding, especially for longer contexts. More recently, BehnamGhader et al. (2024) utilized the vast knowledge of pre-trained LLMs to develop a strong sentence encoder. However, the sentence representation learned by such models is not context-aware, and no prior works focused on context-aware sentence encoding that is required for our prompt compression approach.

3 Method

3.1 Problem Definition

Let $x = \{x_i\}_{i=1}^{L}$ be an input context of length L tokens. The goal of the prompt compression is to produce a compressed prompt $\tilde{x} = \{\tilde{x}_i\}_{i=1}^{\tilde{L}}$, where $\tilde{L} < L$. The compression ratio can be denoted as $\tau = \tilde{L}/L$, where $\tau \in [0, 1]$. The goal of an efficient compression method is to generate a compressed prompt (with smaller τ) while keeping the relevant context preserved so that the performance of the LLM on the compressed prompt $\tilde{x}$ matches that of the original prompt x. However, existing solutions often rely on token-level compression, which can come with a key drawback. The removal of intermediate tokens from a sentence may result in a non-coherent and grammatically incorrect sentence, often hampering the semantics of the input prompt and causing a drop in the performance of LLM. In this work, we propose a sentence-level compression method that removes sentences from the given context based on their relevance to the input question. The key innovation of our proposed method is a context-aware sentence encoder, which is utilized to find the relevance of each sentence in the context, given a question. Our method requires a dataset of context, question, positive, and negative tuples for training our proposed context-aware sentence encoder, which we discuss below. Subsequently, we discuss our proposed method for training the context-aware sentence encoder and prompt compression pipeline during inference.

3.2 Dataset Curation

To train the context-aware sentence encoder, we first create a dataset containing tuples of (long) contexts, questions, positive sentences, and negative sentences, called Context-aware Question-Relevance (CQR) Dataset. The ‘context’ is the text input that contains all the relevant information to answer the ‘question’, which is a meaningful query that asks some key information regarding the ‘context’. The ‘positives’ are defined as sentences within the ‘context’ that contain *some*, but not necessarily all relevant information to answer the question. Finally, the ‘negatives’ are ‘context’ sentences that contain no information relevant to answering the question. To generate a dataset of such tuples, we use a two-step approach where we first generate a set of questions, positives, and negatives, followed by a filtering step. We discuss these two steps below.

We start with the WikiText dataset (Merity et al. 2017) as the seed dataset for our synthetic data generation. This dataset consists of long Wikipedia pages, each containing factual information about historical/scientific concepts. First, we take a document from the original dataset which we consider as context C. This document consists of sentences $\{S_i\}_{i=1}^{K}$. Next, we sample ‘positive’ sentence P from S_i. During the sampling procedure, we ensure that P is a consistent English sentence that contains coherent information. We consider a sentence to be consistent and coherent if at least $\theta\%$ of its words are from the English dictionary and consist of ASCII-only characters. Then, we prompt a pre-trained LLM ψ to generate synthetic question-answer pairs (Q_j, A_j) for this particular sentence P while also considering its context C. This is performed using Prompt 1 below.

```
Prompt 1 (Question Prompt):
Here is a text to consider: TEXT: "text"
Read the sentence in double brackets,
namely, [[sentence]].
Ask questions to this sentence, and make
sure the question is not answerable from
this sentence alone without knowing the
context.
Reply in this format:
Q: {question 1}
A: {answer 1}
Q: {question 2}
A: {answer 2}
```

Note that while some sentences may not directly contain the information required to answer the question, they may provide key contextual clues needed to do so indirectly. For example:

```
Q: How many children does John have?
C: (1) John and Mary have been married for
10 years. (2) They have their anniversary
now and they have decided to travel to
Spain for a month. (3) Mary is worried
that her two children are too small to
travel on the plane.(4) So she has asked
her sister to look after them.
```

In the example above, sentences (1) and (3) do not contain sufficient information to answer question Q, but together they form a subset of sentences that contains all the necessary information. Note that if we evaluate the sentence embedding similarity of the question and each of these sentences, the similarity between sentence (3) and Q would be relatively low due to the absence of any direct reference to John. Accordingly, we need our compression model to train on questions that can not be answerable solely based on P. To ensure the question is not fully answerable from the sentence P alone, we use the LLM to verify each sentence/question/answer triplet with Prompt 2 as follows:

```
Prompt 2 (Verification Prompt):
You are given a piece of text, a question
and an answer. Verify whether it is
possible to derive such an answer by
considering only the given piece of text
(you should rely only on the piece of
text). Think step by step and finish
your thoughts with one word: "Yes" or
"No". Answer "Yes" if and only if ALL the
necessary information is contained in the
text. If anything is missing, then state
what is missing and answer "No". Answer
"Yes" ONLY if there is no such information
in the answer that is missing in the text.
Otherwise, answer "No"!!
{A demonstration}
Text: {context sentence}
Question: {question}
Answer: {answer}
Verification result: Yes/No
```

Here the $Q\&A$ generated pairs that have passed the verification step (obtained verification result "No") along with P and C are kept in the dataset. Next, to gather the negative sentences, we employ a pre-trained sentence transformer (Reimers and Gurevych 2019), denoted as $\mathcal{E}$. This transformer generates embeddings for each sentence S_i in the context, represented as $\{E_i = \mathcal{E}(S_i)\}_{i=1}^{K}$. Additionally, it produces embeddings for the question Q and the positive sentence P, denoted as $E_q = \mathcal{E}(Q)$ and $E_p = \mathcal{E}(P)$, respectively. We define a similarity score η as the cosine similarity between the embeddings of the positive sentence and the question, $\eta = \cos(E_p, E_q)$, which we use to identify candidate negative sentences from the context. These candidates are the sentences whose similarity to the question is less than η, i.e., $\cos(\mathcal{E}(S_j), E_q) < \eta$ for any sentence S_j in the context. In case the number of sentences with a similarity score less than or equal to η is less than $\beta\%$ of the total sentences, i.e. $|\{i \mid \cos(E_i, E_q) \leq \eta\}| < \beta \cdot K$, we exclude such ($C$, Q, P) tuples from our dataset.

Once we generate a set of negative candidates, we introduce a negative filtering method using a probability density function to further filter the negatives. Specifically, for a negative sentence S_{Neg}, we use a pre-trained LLM, ϕ, to obtain the probability density of the answer tokens A, given the context C without the negative sentence as:

$$P_{C \text{ w/o } Neg} = \phi\left(A \mid \{S_i\}_{i=1..K, i \neq neg}\right). \quad (1)$$

Similarly, we calculate the probability density of the answer tokens, given the whole context as:

$$P_C = \phi\left(A \mid \{S_i\}_{i=1..K}\right). \quad (2)$$

For a negative sentence with no information about the question, these two distributions should be close. If the KL-divergence between two distributions ($KL(P_C, P_{C \text{ w/o } Neg}$) is high ($> \lambda$), it indicates that the negative sentence has information regarding the question. Consequently, we filter out such negatives from our dataset. An overview of the data collection pipeline is illustrated in Figure 2.

3.3 Context-aware Sentence Encoding

Next, our pipeline relies on a context-aware sentence encoder, f_θ to capture the semantics of a sentence or question, given its context. We train f_θ on our *CQR dataset* in a contrastive learning setup. Specifically, f_θ is trained to learn the context-aware representation by minimizing distances between the question and positive samples (context sentences that contain some information regarding the question) while maximizing it from the negative samples (irrelevant context sentence). Let's denote a training batch of our dataset be, $\mathcal{B} = \{C_b, Q_b, P_b, \{N_b^{n=1..M}\}\}_{b=1..B}$, where C, Q, P, and N are the context, question, positive, and negatives. For a sample $1 \leq b \leq B$, we consider the exponent of the cosine similarity between the question's semantic embedding (ξ_{Q_b}) and the context-aware sentence embedding of the positive sample ($\xi_{(P_b,C_b)}$) as positives, i.e, $Sim_P = exp(cosine(\xi_{Q_b}, \xi_{(P_b,C_b)}))$, where $\xi_{P,C}$ is the context-aware representation of sentence P in the context of C, defined as:

$$\xi_{P,C} = norm(\frac{\sum_{t=i}^{j} Z_t}{|j - i + 1|}). \quad (3)$$

Here, Z_i is the token-level embedding of token x_i, calculated as:

$$\{Z_i\}^{i=1..L} = f_\theta(x_0, ..., x_L), \quad (4)$$

where i and j are the start and end indexes of the positive sentence P in context C. Similarly, the question embedding (without the context) is represented as ξ_Q. Similarly, we define the negative as $Sim_N = exp(cosine(\xi_{Q_b}, Neg_{(b,ext)})$, where $Neg_{(b,ext)}$ consists of the corresponding negative sentences of sample b, along with all the positives and negatives from the same batch $\mathcal{B}$ excluding b, i.e. $Neg_{(b,ext)} = \{\xi_{(P_i,C_i)}\}_{i=1..B|i \neq b} \cup \{\xi_{(N_i^n,C_i)}\}_{i=1..B|i \neq b}^{n=1..M} \cup \{\xi_{(N_b^n,C_b)}\}^{n=1..M}$. Consequently, we denote our contrastive training loss as:

$$\mathcal{L}_{SC} = -log\frac{exp(Sim_P)}{exp(Sim_P) + \sum exp(Sim_N)}, \quad (5)$$

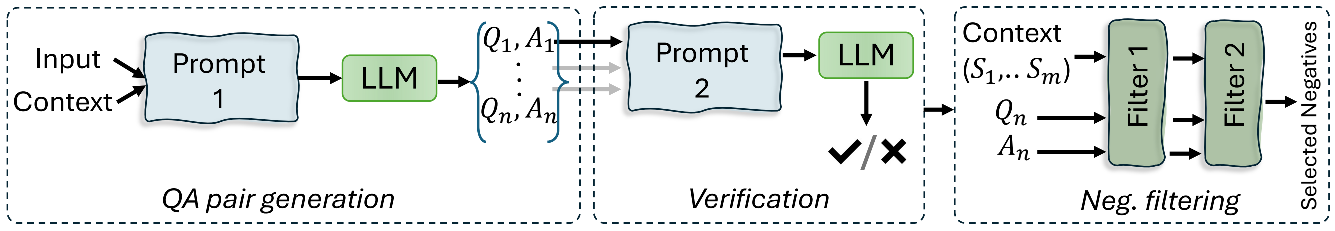


Figure 2: Data collection pipeline. The data collection process starts by asking the LLM with *Prompt 1* to generate a set of (Q,A) pairs. In the next step, we verify the pairs with *Prompt 2*. In the last step, we select a set of negatives for each (Q,A) pair.

In practice, we do not train f_θ from scratch. Rather, we adopt a pre-trained LLM and use the concept introduced in BehnamGhader et al. (2024) through finetuning to learn bidirectional attention over the whole context. More specifically, we mask $\delta\%$ of the tokens from the context, and train the model with a masked next token prediction loss:

$$\mathcal{L}_{MNTP} = \sum_{i}^{tokenlen(C)} -logP(x_i|X_{um}), \qquad (6)$$

where X_{um} is the sequence of all the unmasked tokens and $P(x_i)$ is the probability distribution over the vocabulary for the ith masked token. Finally, we train the model with the combination of the two losses: $\mathcal{L} = \mathcal{L}_{SC} + \mathcal{L}_{MNTP}$.

3.4 Inference

At inference, given context C and question Q, our aim is to generate a compressed context $C_{compressed}$ that preserves all the relevant information to answer the question Q. We start by splitting the context into sentences $\{S_i\}_{i=1}^{K}$ and generating the context-aware embedding for each sentence $\xi_{S_i,C}$ as explained above. Similarly, we generate the question embedding as ξ_Q. Next, we calculate the embedding similarly between ξ_Q and each $\xi_{S_i,C}$ to generate the relevance score for all the sentences: $S_i = cosine(\xi_{S_i,C}, \xi_Q)$. For a compression ratio of τ, we take the top T sentences with the highest relevance scores such that $\sum_{t=1..T} tokenlen(S_t) \leq \tau \cdot tokenlen(C)$. We then restore the order of selected sentences in the original context to form the compressed context $C_{compressed}$.

4 Experiments

4.1 Datasets

We train our proposed context-aware sentence encoder on our own collected dataset (CQR dataset) described earlier. For evaluation, we follow the experimental setup of previous works (Jiang et al. 2023c) and report the performance of our proposed solution on LongBench (Bai et al. 2023) and ZeroSCROLLS (Shaham et al. 2023). LongBench is a dataset with different sub-tasks, including single-document QA, multi-document QA, and more. Similarly, ZeroSCROLLS is another dataset with different sub-tasks, including summarization, QA, and few-shot learning. We also evaluate on domain-specific benchmarks such as medical paper summarization (PubMed (Cohan et al. 2018)), keyword extraction (Krapivin (Krapivin et al. 2009)), meeting transcript summarization (MeetingBank (Hu et al. 2023)) and TV show summarization (SummScreen (Chen et al. 2021)) datasets.

4.2 Implementation Details

For collecting our dataset, we use WikiText (Merity et al. 2017) as our seed corpus. Our method is flexible towards the choice of pre-trained LLM that can be used as ψ and ϕ in our dataset collection method. However, given that we need strong text generation capabilities for ψ and access to the probability density outputs for ϕ, we chose the OpenAI *gpt-3.5-turbo-instruct* and *microsoft/Phi-3-small-128k-instruct* respectively. We used *sentence-transformers/all-mpnet-base-v2* as the sentence encoder (Reimers and Gurevych 2019) for negative collection. Finally, we set $\beta = 30\%$, $\lambda = 4e-3$, $\delta = 80\%$, and $M = 2$.

For f_θ, we adopt the *Mistral-7B-Instruct-v0.2* (Jiang et al. 2023a) given that it is open-source and allows for long context lengths, and fine-tune this model with *LoRA* (Hu et al. 2022) of rank 16 to learn the context-aware embeddings. We train the model with 2048 samples with a batch size 32, and AdamW optimizer with a learning rate of $5e-5$. The training is conducted on a single A100 80G GPU.

4.3 Evaluation Protocols

We perform various experiments on several downstream tasks based on commonly used protocols. We evaluate the effectiveness of our method on summarization problems by calculating the Rouge metric (Lin 2004) between the ground truth responses versus the model's generated outputs given compressed prompts. For document QA tasks, we measure the F1 score between the response generated by the model and the ground truth response. For code completion tasks, we use the textual similarity metric based on the Levenshtein edit distance (Yujian and Bo 2007). For the different subsets of LongBench and ZeroSCROLLS, we follow the same evaluation protocol adopted by existing literature (Jiang et al. 2023c). For the keyword extraction, we evaluate the recall of the keywords that LLM extracts from a compressed context in relation to ground truth keywords.

4.4 Results

Main results. The main results of our proposed solution on the LongBench and ZeroSCROLLS datasets are presented in Table 2. Following prior works, we report the re-

Methods	LongBench									ZeroSCROLLS		
	SingleDoc	MultiDoc	Summ.	FewShot	Synth.	Code	AVG	Tokens	$1/\tau$	AVG	Tokens	$1/\tau$
Original Prompt	39.7	38.7	26.5	67.0	37.8	54.2	44.0	10,295	-	32.5	9,788	-
Zero-shot	15.6	31.3	15.6	40.7	1.6	36.2	23.5	214	48×	10.8	32	306×
3,000 tokens constraint												
Retrieval-based Methods												
BM25	32.3	34.3	25.3	57.9	45.1	48.9	40.6	3,417	3×	19.8	3,379	3×
SBERT	35.3	37.4	26.7	63.4	51.0	34.5	41.4	3,399	3×	24.0	3,340	3×
OpenAI	34.5	38.6	26.8	63.4	49.6	37.6	41.7	3,421	3×	22.4	3,362	3×
LongLLMLingua	37.6	42.9	26.9	68.2	49.9	53.4	46.5	3,424	3×	29.3	3,350	3×
Compression-based Methods												
Selective-Context	23.3	39.2	25.0	23.8	27.5	53.1	32.0	3,328	3×	20.7	3,460	3×
LLMLingua	31.8	37.5	26.2	67.2	8.3	53.2	37.4	3,421	3×	30.7	3,366	3×
LLMLingua-2	35.5	38.7	26.3	69.6	21.4	62.8	42.2	3,392	3×	33.5	3,206	3×
LongLLMLingua	40.7	46.2	**27.2**	**70.6**	**53.0**	55.2	48.8	3,283	3×	32.8	3,412	3×
CPC (Ours)	**42.7**	**47.9**	23.8	69.3	52.8	**63.5**	**50.0**	3,327	3×	**34.9**	3,470	3×
2,000 tokens constraint												
Retrieval-based Methods												
BM25	30.1	29.4	21.2	19.5	12.4	29.1	23.6	1,985	5×	20.1	1,799	5×
SBERT	33.8	35.9	25.9	23.5	18.0	17.8	25.8	1,947	5×	20.5	1,773	6×
OpenAI	34.3	36.3	24.7	32.4	26.3	24.8	29.8	1,991	5×	20.6	1,784	5×
LongLLMLingua	37.8	41.7	26.9	66.3	53.0	52.4	46.3	1,960	5×	24.9	1,771	6×
Compression-based Methods												
Selective-Context	16.2	34.8	24.4	15.7	8.4	49.2	24.8	1,925	5×	19.4	1,865	5×
LLMLingua	22.4	32.1	24.5	61.2	10.4	56.8	34.6	1,950	5×	27.2	1,862	5×
LLMLingua-2	29.8	33.1	25.3	66.4	21.3	58.9	39.1	1,954	5×	33.4	1,898	5×
LongLLMLingua	39.0	42.2	**27.4**	69.3	**53.8**	56.6	48.0	1,809	6×	32.5	1,753	6×
CPC (Ours)	**42.6**	**48.6**	23.7	**69.4**	52.8	**60.00**	**49.5**	1,844	5×	**33.8**	1,821	5×

Table 2: Performance of different methods under different compression ratios on LongBench and ZeroSCROLLS.

Method	Krapivin-2009	PubMed	MeetingBank	SummScreen
LLMLingua-2	36.7	26.24	16.73	21.57
LongLLMLingua	17.1	24.31	14.39	17.48
CPC	**43.5**	**27.59**	**17.66**	**21.83**

Table 3: Comparison to prior works on different domain-specific evaluation tasks.

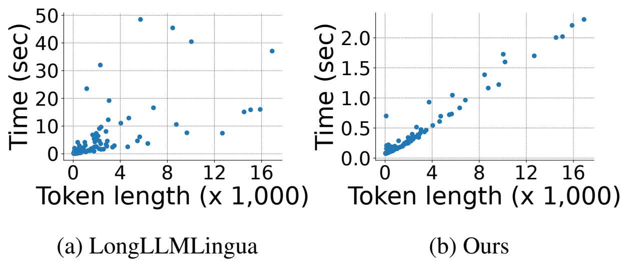
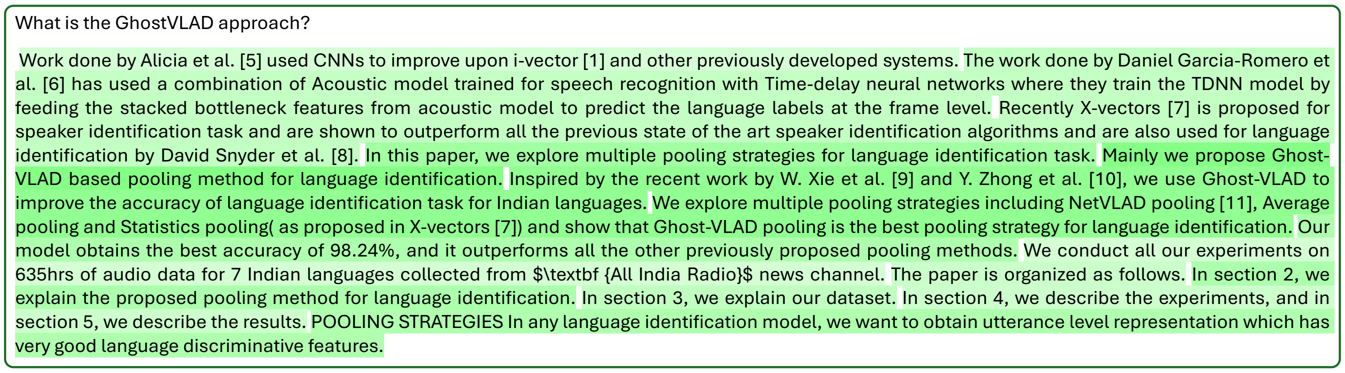


Figure 3: Comparison of latency between LongLLMLingua and our proposed method.

sults on LongBench on the sub-tasks of a single document, multi-document, summarization, few-shot, synthetic, code, and the average over all the tasks. Following Jiang et al. (2023c), the results are reported on 3,000 and 2,000 tokens, which is about 3× and 5× compression of the total token length. For ZeroSCROLLS, we present the performance with the same 3,000 and 2,000 tokens. As we observe in this table, our method outperforms the prior works by a considerable margin on LongBench with both 2,000 and 3,000 token constraints. More specifically, on 2,000 tokens, CPC outperforms others by 1.2 points on average and up to 8.3 points on individual tasks. On evaluation with 2,000 tokens, CPC shows a 1.5 points improvement on average. On Zero-

Embed.	Sin. Doc	Mul. Doc	Summ.	FewShot	Synth.	Code	AVG
LLMLingua-2	33.39	52.07	24.88	46.94	27.0	59.84	40.69
LongLLMLingua	26.70	39.25	22.66	58.45	13.75	27.73	31.42
CPC (Ours)	42.43	62.86	25.74	69.25	53.75	63.44	52.91

Table 4: Comparison to prior works with larger pre-trained model.



What is the GhostVLAD approach?

Work done by Alicia et al. [5] used CNNs to improve upon i-vector [1] and other previously developed systems. The work done by Daniel Garcia-Romero et al. [6] has used a combination of Acoustic model trained for speech recognition with Time-delay neural networks where they train the TDNN model by feeding the stacked bottleneck features from acoustic model to predict the language labels at the frame level. Recently X-vectors [7] is proposed for speaker identification task and are shown to outperform all the previous state of the art speaker identification algorithms and are also used for language identification by David Snyder et al. [8]. In this paper, we explore multiple pooling strategies for language identification task. Mainly we propose Ghost-VLAD based pooling method for language identification. Inspired by the recent work by W. Xie et al. [9] and Y. Zhong et al. [10], we use Ghost-VLAD to improve the accuracy of language identification task for Indian languages. We explore multiple pooling strategies including NetVLAD pooling [11], Average pooling and Statistics pooling(as proposed in X-vectors [7]) and show that Ghost-VLAD pooling is the best pooling strategy for language identification. Our model obtains the best accuracy of 98.24%, and it outperforms all the other previously proposed pooling methods. We conduct all our experiments on 635hrs of audio data for 7 Indian languages collected from $\textbf {All India Radio}$ news channel. The paper is organized as follows. In section 2, we explain the proposed pooling method for language identification. In section 3, we explain our dataset. In section 4, we describe the experiments, and in section 5, we describe the results. POOLING STRATEGIES In any language identification model, we want to obtain utterance level representation which has very good language discriminative features.

Figure 4: An example illustrating the question and sentence relevance scores from our context-aware sentence encoder. Here, the more relevant sentences are coloured with higher intensity.

Method	Avg.	Med.	Avg. Rel.	Med. Rel.
LLMLingua	4.73	1.47	16.89×	9.8×
LLMLingua-2	0.23	0.1	0.82×	0.67×
LongLLMLingua	7.70	1.64	27.5×	10.93×
CPC	0.28	0.15	1×	1×

Table 5: Inference latency for different methods. Here, Avg., Med., and Rel., stand for average, median and relative time.

SCROLLS, our proposed method also outperforms the previous state-of-the-art by 1.4 points on 3,000 tokens and 0.4 points on 2,000 tokens.

Domain generalization. Since CPC effectively removes sentences that do not contain any useful information regarding the question or the prompt, it holds the potential for a wide range of tasks/domains. To this end, we investigate the performance of CPC on new domain-specific benchmarks described earlier in Section 4.1. The results of these experiments are presented in Table 3 with a 2,000-token constraint. As we observe from this table, CPC achieves significant improvements over prior methods across all domain-specific tasks. Specifically, CPC outperforms LLMLingua-2 by 6.8, 1.35, 0.93, and 0.26 points on Krapivin-2009, PubMed, MeetingBank, and SummScreen, respectively.

Larger backbone. To evaluate the scalability of our method with a larger model, we compare the performance of CPC against prior works with a larger LLM, GPT-4o, as the backbone. The results of this study are presented in Table 4. As we find from this table, CPC consistently outperforms existing methods across all subtasks of the LongBench. On average, CPC outperforms LLMLingua-2 and LongLLMLingua by 12.22 and 21.49 points, respectively.

Latency evaluation. Here, we discuss the latency of our proposed solution and compare it to prior works on prompt compression. We summarize the findings in Table 5. All the evaluations are conducted on the same A100 GPU used for training. The reported results are the average processing time of a sample of the LongBench. We report the results for our method in comparison to three prior SOTAs: LLMLingua (Jiang et al. 2023b), LLMLingua-2 (Pan et al. 2024), and LongLLMLingua (Jiang et al. 2023c). As we find from this study, our method is significantly faster than LLMLingua and LongLLMLingua, showing up to 27.5× and 10.93× average and median speedup. While LLMLingua-2 is slightly faster than our solution during inference, its performance is considerably worse than ours in all benchmarks and even worse than LongLLMLingua in most tasks. We also analyze the effect of the length of the original context on the latency of each of the above approaches. We depict these results in Figure 3. As we find from the Figure, the time complexity of our approach scales linearly with the growing length of the input context, while having relatively low latency compared to LongLLMLingua. On the other hand, LongLLMLingua has high latency for some of the samples with shorter context due to the dependence on the text structure, while our method is agnostic to the text structure.

Ablation studies. Here we present ablation studies on different components of the proposed method. All the ablation studies are conducted on the LongBench subset Single-

Ablation	Accuracy
$\mathcal{L}_{SC} + \mathcal{L}_{MNTP}$	**42.38**
$\mathcal{L}_{MNTP}$	35.46
$\mathcal{L}_{SC}$	41.17

(a) Ablation on loss function.

Setup	Accuracy
Ours	**42.38**
w/o question verification	41.51
w/o sim-based neg. filter	41.34
w/o KL-based neg. filter	40.71

(b) Dataset collection.

Setup	Accuracy
2 negatives	**42.38**
4 negatives	41.52
8 negatives	41.06

(c) Context-aware emb.

Table 6: Ablation and analysis of different components of CPC, conducted on the SingleDoc subset of LongBench.

Embed.	Sin. Doc	Mul. Doc	Summ.	FewShot	Synth.	Code	AVG
MpNet-v2	38.4	47.5	23.5	65.1	40.5	55.5	47.2
LLM2Vec	38.3	45.7	**24.7**	67.5	27.0	54.3	42.9
Ours	**42.6**	**48.6**	23.7	**69.4**	**52.8**	**60.0**	**49.5**

Table 7: Performance with different sentence encoders.

Doc consisting of NarrativeQA, Qasper and MultiFieldQa-en datasets. We perform these ablations on the higher compressed ratio with 2,000 tokens. First, we present an ablation study on the loss function used for training the context-aware sentence encoder. As discussed earlier, we train the encoder with the combination of two losses, $\mathcal{L}_{SC}$ and $\mathcal{L}_{MNTP}$. In Table 6a, we present the results for removing each of these components. As we find from this study, removing the $\mathcal{L}_{MNTP}$ shows a large drop in performance since $\mathcal{L}_{MNTP}$ helps the model to learn the bidirectional attention that captures the context of the whole sequence. Similarly, removing $\mathcal{L}_{SC}$ also results in a large drop in the performance as the ablated model does not learn the context-aware embeddings that are required to understand the relevance of each sentence to the given context and question.

Next, we ablate the design choices of different components of the dataset collection procedure. We note that the dataset collection has several steps of verification and filtering: question and answer verification, similarity-based negative filtering, and KL-based negative filtering. In Table 6b, we present an ablation study by removing each of these steps from our dataset collection pipeline. As we find from this table, removing question verification results in a 0.87 point drop in performance. This is caused by the fact that without question verification, the sentence and positive pairs are generated such that the question is directly fully answerable without the need to consider the context. Next, we find that removing the similarity-based negative filtering and KL-based filtering results in a 1.04 and 1.67 points drop in performance, respectively, showing the importance of these steps in ensuring that the negative does not contain any information regarding the question.

Next, we study the context-aware sentence embedding learning with different numbers of negatives. More specifically, we study 2, 4, and 8 negatives per positive samples in Table 6c. The results indicate that the best context-aware sentence embedding is learned with 2 negatives per positive sample. Adding more negatives results in an increase in the computational cost during training while also showing reduced performance compared to 2 negatives.

Finally, we study the impact of our learned context-aware encoder versus pre-trained off-the-shelf semantic encoders. The results from this study are presented in Table 7, comparing our encoder to MpNet-v2 and LLM2Vec. As we find from this table, semantic embeddings are considerably less effective in finding the relevance of context sentences to a question to perform context compression. The average scores for MpNet-v2 and LLM2Vec are 2.3 and 6.6 points lower than our proposed context-aware sentence encoder.

Qualitative analysis. We present a qualitative analysis of our prompt compression method in Figure 4. Here, the first line shows a question, and the rest of the paragraph is the input context. The intensity of the colour indicates the relevance score of each sentence to the question predicted by our method. As we observe in this example, the sentences that are relevant to the question ('What is the GhostVLAD approach?') have higher scores than sentences that have less relevance.

5 Conclusion

In this work, we address the problem of prompt compression by developing a novel sentence-level context-aware compression method. To train our method, we first generated a new dataset of questions, positive, and negative pairs using a pre-trained LLM, followed by several steps of verification and filtering. We then use this dataset to contrastively train our novel context-aware sentence encoder. Detailed experiments demonstrate that our method is considerably faster than the state-of-the-art while showing considerable improvements on benchmark question-answering datasets. Additionally, our method shows strong domain generalization and robust performance with larger encoders. We believe our work will foster research in this promising direction of reducing the inference cost of LLMs.

References

Bai, Y.; Lv, X.; Zhang, J.; Lyu, H.; Tang, J.; Huang, Z.; Du, Z.; Liu, X.; Zeng, A.; Hou, L.; et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

BehnamGhader, P.; Adlakha, V.; Mosbach, M.; Bahdanau, D.; Chapados, N.; and Reddy, S. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Bulatov, A.; Kuratov, Y.; and Burtsev, M. 2022. Recurrent memory transformer. *Advances in Neural Information Processing Systems*, 11079–11091.

Chen, H.; Pasunuru, R.; Weston, J. E.; and Celikyilmaz, A. 2023. Walking Down the Memory Maze: Beyond Context Limit through Interactive Reading. *arXiv preprint arXiv:2310.05029*.

Chen, M.; Chu, Z.; Wiseman, S.; and Gimpel, K. 2021. Summscreen: A dataset for abstractive screenplay summarization. *arXiv preprint arXiv:2104.07091*.

Chevalier, A.; Wettig, A.; Ajith, A.; and Chen, D. 2023. Adapting Language Models to Compress Contexts. In *Conference on Empirical Methods in Natural Language Processing*, 3829–3846.

Church, K. W. 2017. Word2Vec. *Natural Language Engineering*, 155–162.

Cohan, A.; Dernoncourt, F.; Kim, D. S.; Bui, T.; Kim, S.; Chang, W.; and Goharian, N. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*.

Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Wu, Z.; Chang, B.; Sun, X.; Xu, J.; and Sui, Z. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Ge, T.; Jing, H.; Wang, L.; Wang, X.; Chen, S.-Q.; and Wei, F. 2024. In-context Autoencoder for Context Compression in a Large Language Model. In *International Conference on Learning Representations*.

Ghalandari, D. G.; Hokamp, C.; and Ifrim, G. 2022. Efficient Unsupervised Sentence Compression by Fine-tuning Transformers with Reinforcement Learning. *arXiv preprint arXiv:2205.08221*.

Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Hu, Y.; Ganter, T.; Deilamsalehy, H.; Dernoncourt, F.; Foroosh, H.; and Liu, F. 2023. MeetingBank: A benchmark dataset for meeting summarization. *arXiv preprint arXiv:2305.17529*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023a. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

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

Jiang, H.; Wu, Q.; Luo, X.; Li, D.; Lin, C.-Y.; Yang, Y.; and Qiu, L. 2023c. LongLLMLingua: Accelerating and Enhancing LLMs in Long Context Scenarios via Prompt Compression. In *Annual Meeting of the Association for Computational Linguistics*.

Jung, H.; and Kim, K.-J. 2024. Discrete prompt compression with reinforcement learning. *IEEE Access*.

Kim, S.; Shen, S.; Thorsley, D.; Gholami, A.; Kwon, W.; Hassoun, J.; and Keutzer, K. 2022. Learned token pruning for transformers. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 784–794.

Krapivin, M.; Autaeu, A.; Marchese, M.; et al. 2009. Large dataset for keyphrases extraction. *Technical report, University of Trento*.

Laban, P.; Schnabel, T.; Bennett, P.; and Hearst, M. A. 2021. Keep It Simple: Unsupervised Simplification of Multi-Paragraph Text. In *International Joint Conference on Natural Language Processing*, 6365–6378.

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

Li, Y.; Dong, B.; Guerin, F.; and Lin, C. 2023a. Compressing Context to Enhance Inference Efficiency of Large Language Models. In *Conference on Empirical Methods in Natural Language Processing*, 6342–6353.

Li, Z.; Zhang, X.; Zhang, Y.; Long, D.; Xie, P.; and Zhang, M. 2023b. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 74–81.

Merity, S.; Xiong, C.; Bradbury, J.; and Socher, R. 2017. Pointer Sentinel Mixture Models. In *International Conference on Learning Representations*.

Pan, Z.; Wu, Q.; Jiang, H.; Xia, M.; Luo, X.; Zhang, J.; Lin, Q.; Rühle, V.; Yang, Y.; Lin, C.-Y.; et al. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. *arXiv preprint arXiv:2403.12968*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, 1532–1543.

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

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *International Joint Conference on Natural Language Processing*, 3982–3992.

Shaham, U.; Ivgi, M.; Efrat, A.; Berant, J.; and Levy, O. 2023. ZeroSCROLLS: A Zero-Shot Benchmark for Long

Text Understanding. In *Conference on Empirical Methods in Natural Language Processing*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Wang, C.; Yang, Y.; Li, R.; Sun, D.; Cai, R.; Zhang, Y.; Fu, C.; and Floyd, L. 2024. Adapting LLMs for efficient context processing through soft prompt compression. *arXiv preprint arXiv:2404.04997*.

Wang, L.; Yang, N.; Huang, X.; Jiao, B.; Yang, L.; Jiang, D.; Majumder, R.; and Wei, F. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Wang, Y.; and Xiao, Z. 2024. LoMA: Lossless Compressed Memory Attention. *arXiv preprint arXiv:2401.09486*.

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

Xu, F.; Shi, W.; and Choi, E. 2024. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *International Conference on Learning Representations*.

Yujian, L.; and Bo, L. 2007. A normalized Levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1091–1095.