Overcoming Self-Imposed Limits: Five Words to Break an LLM's Context Compression Barrier

Anonymous ACL submission

Abstract

This paper focuses on efficient Large Language Model data compression. Considering the linear context growth of self-evaluating and divide-and-conquer LLM modeling methods, techniques are needed to manage the size of shared context. Existing approaches compress data through prompt tuning, using detailed instructions to guide the output. However, this method may be suboptimal: (i) defining principles may restrict an LLM's inherent ability to compress data; (ii) longer prompts increase the overhead needed to process data.

To address these issues, we built upon the framework proposed by LLMLingua2, which formulates data compression as a token classification problem, and trains knowledge distilled models on data generated using compression prompts. We observed their model's output, designed new prompts targeted at areas of improvement, and evaluated on downstream tasks, such as summarization, question answering and mathematics. We then test our best prompting method on the summarization task of MeetingBank, 3% the size of LLMLingua2's prompt, while achieving a 61% size reduction of distilled data and higher model evaluation result than LLMLingua2's prompting method on all eight different metrics, at a low resource level of 1000 training pairs.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across various NLP tasks, but their effectiveness often relies on utilizing long, detailed prompting techniques such as In-Context Learning (Dong et al., 2024) or routine correspondence in multi-agent systems (Wu et al., 2023). This increases computational cost and memory usage, making LLM deployment less efficient in real world applications.

One approach to improving efficiency is data compression: reducing context length while maintaining task performance. Existing methods such as prompt tuning (Lester et al., 2021) and Retrieval Augmented Generation (Lewis et al., 2020), often introduce information loss or require extensive fine-tuning. We explored prompt tuning for compression in Appendix C, evaluated on the similarity of context before and after compression, but did not have the resources to evaluate on downstream tasks. This shows the need for a more resource-friendly method of compression, without inferencing an LLM at each step of the process. However, we did reach the conclusion that allowing the LLM to compress data with less instructions could be more efficient.

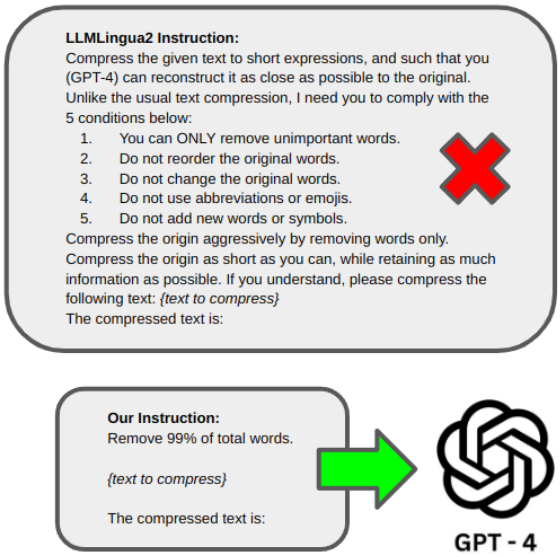

Figure 1: Shorter compression prompt compared to LLMLingua2.

To make our compression method reproducible, we build upon the framework proposed by LLMLingua2 (Pan et al., 2024) and train knowledge distilled compressors that preserves task critical information while significantly reducing the prompt size needed to generate training data. Compressed data

Issue	LLMLingua2 Output	ID
Conceptual Ambiguity	LEWIS. Yes. Morales Yes. Peterson Yes. Council President Gonzales favor unopposed.	36
Duplication Redundancy	Public Safety and Human Services Committee.? Public Safety Human Services Committee.	17
Duplication Redundancy	consent calendar motion on consent calendar? fox consent calendar	5
Syntactical Redundancy	vote June 3rd.? & i..vote no.	1

Table 1: Analyzed cases of custom trained LLMLingua2 model's compression output consolidated into three areas for improvement. Specifications are the same as in Table 4. ID taken from corresponding MeetingBank test cases.

from trained models are evaluated as context for downstream tasks such as summarization, question answering and mathematics, on the same datasets used by LLMLingua2, mainly MeetingBank (Hu et al., 2023) and GSM8K (Cobbe et al., 2021).

As referenced in Figure 1, our method removes the self-imposed limitations set forth in LLMLingua2, while superseding the LLM's context compression barrier. Word-wise, our best performing prompt is 3% the length of LLMLingua2's prompt, and achieves a 86% reduction in length of output compared to 64% using LLMLingua2's prompt. Model trained on our distilled data maintains comparable task performance to LLMLingua2 on the summarization task of MeetingBank, verified through multiple metrics.

Our contributions are as follows:

- We identify key limitations in existing prompt compression techniques.
- We propose, and evaluate, different prompting methods for compression.
- We demonstrate significant prompt length reduction while generating more concise data for more efficient context compressor training.

2 Related Work

Large-scale language models (LLMs) such as GPT-3 and GPT-4 require carefully crafted prompts to perform well on downstream tasks. Early work in prompt engineering focused on manual prompt design (Brown et al., 2020) and prompt tuning (Lester et al., 2021), where task-specific prompts were either hand-crafted or automatically optimized. However, as the complexity of tasks grew beyond the token limit of LLMs, the need to efficiently compress context without losing critical information became evident.

Recent approaches have explored various compression strategies, from simple prompt compression (Li et al., 2023) to more sophisticated methods, such as learned token pruning (Kim et al., 2022), to reduce the context length while maintaining semantic integrity.

In natural language processing, distillation (Hinton, 2015) has been successfully applied to model compression (Sanh, 2019), enabling compact models to retain performance levels close to their larger counterparts. This paradigm has been extended to various tasks including machine translation (Kim and Rush, 2016), text summarization (Liu et al., 2021), and question answering (Jiao et al., 2020). Since proprietary LLMs are mostly inaccessible, black-box knowledge distillation (Wang, 2021) has emerged as an effective technique to transfer information from large, complex models to smaller, more efficient ones.

While context compression and knowledge distillation have individually proven beneficial, only a limited number of studies have explored their integration. Existing methods that focus on context compression often suffer from information loss, especially when the reduction is aggressive. On the other hand, knowledge distillation techniques have not been widely applied to compression tasks, where the goal is to create a condensed yet semantically rich representation of the original context. Recent work (Pan et al., 2024) proposes a teacher–student framework that guides context compression, but their method require detailed prompts to generate training data for knowledge distillation, increasing financial and computational overhead. Thus, we propose to train a knowledge distilled context compressor using shorter prompts, while ensuring comparable performance against model trained on data generated using LLMLingua2's prompting method for downstream tasks.

3 Method

In this section, we present our approach to generating high quality compression pairs for training a compression model using a teacher–student framework where GPT-4 serves as the teacher. Our goal is to compress lengthy prompts into concise representations that retain critical task-relevant in-

Comparison	LLMLingua2	ND	ND+DR	ND+SR	ND+DR+SR
Summary BLEU	22.74	**23.64**	20.59	21.70	22.63
Summary Rouge-Lsum	41.03	**41.39**	37.86	39.81	40.13
Summary BERTScore F1	**90.34**	90.25	89.60	90.02	90.17
QA Exact Match	41.33	**56.66**	54.66	54.33	51.33
GSM8K Exact Match	88	**95**	92	92	91
Training Data Size ↓	36.23	14.50	19.11	**12.51**	17.78

Table 2: All number represented as percentages. Evaluated on the first 100 cases of each dataset.

formation. The overall pipeline consists of data preparation, teacher generation, student training via knowledge distillation, and evaluation. Since we are utilizing the same model as LLMLingua2, our focus for this paper will be mainly on prompt modifications at the data preparation stage.

We first analyze LLMLingua2's compression result, and consolidate three areas for improvement. Examples of observed issues are shown in Table 1.

1. Conceptual Ambiguity: Concepts repeat due to insufficient compression.

2. Duplication Redundancy: Keywords repeat due to importance unit-wise, but not as a whole.

3. Syntactical Redundancy: Punctuations remain even when context has been dropped or modified.

For each of the above, we design principles to directly counter these issues, and test whether they might be an improvement over LLMLingua2's prompting method for knowledge distillation.

1. Numerical Disambiguity (ND): Specify a near impossible 99% size reduction to push the limits of compression.

2. Duplication Removal (DR): Keep only the first instance of each word.

3. Syntactical Removal (SR): Remove all punctuations.

We test all combinations on mathematics from GSM8K (Cobbe et al., 2021), summarization and question-answering from MeetingBank (Hu et al., 2023), using the first 100 test cases from each to better display evaluation result in percentages, and determine the overall superior method. All models were trained on data distilled using the first 20 instance of MeetingBank, 120 training pairs after chunking. Since GPT-4 seems to ignore prompt commands for syntactical removal, it is enforced by manually modifying GPT-4's output during training data generation, by making a string translation table and removing all instances of punctuation with the translate method in Python 3.11.

Once we find the best combination of the principles above, we can allocate an order of magnitude more resources to train our model using the first 100 instances of MeetingBank, 1000 training pairs after chunking, and compare them with a model trained using LLMLingua2's prompting method at each epoch for 10 epochs, the same amount LLMLingua2 was trained on. We compare both performance during training and evaluation using the first 100 test cases of MeetingBank summarization task, to see if prompts with higher compress rate translate to higher performance on either one. Higher performance on evaluation would signify successful reduction of long context on downstream task, which is the main objective of this study.

Our approach is implemented using Huggingface's Transformers and PyTorch 2.5.1 with CUDA-12.1 on a single NVIDIA 4070 Ti GPU. We use xlm-roberta-large (Conneau et al., 2020) as our model, the same as LLMLingua2, with only minor changes to the code as to allow communication with the newer GPT model, to also test the performance of this knowledge distillation framework with more advanced tools. Under a API budget of $300 USD, all experiments use GPT-4-0613 as the LLM instance, for both dataset generation and downstream task, with LLMLingua2's default settings and parameters for reproducibility.

4 Experiment

We take Numerical Disambiguity as our compression base, and test combinations with Duplication Removal and Syntactical Removal. For each combination, we inference GPT-4 using the training set of MeetingBank to generate 120 training pairs, and train a model for one epoch. In Table 2, we

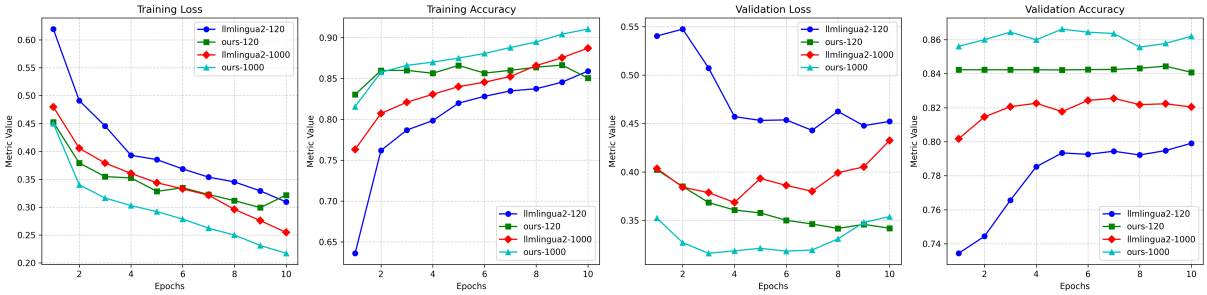

Figure 2: Training result of models using LLMLingua2 versus our best prompting method on 120 and 1000 training pairs. All metrics represented in percentages.

Comparison	LLMLingua2	ND
BLEU	28.68	**29.18**
Rouge-1	59.76	**59.98**
Rouge-2	33.28	**33.98**
Rouge-L	44.46	**45.50**
Rouge-Lsum	44.73	**45.88**
BERTScore Precision	91.33	**91.64**
BERTScore Recall	**91.21**	91.04
BERTScore F1	91.26	**91.33**

Table 3: Comparison of models trained on 120 training pairs. All number represented as percentages. Evaluated on the first 100 cases of MeetingBank summarization task after 10 epochs.

Comparison	LLMLingua2	ND
BLEU	30.13	**32.84**
Rouge-1	60.42	**61.58**
Rouge-2	34.99	**37.68**
Rouge-L	45.48	**47.82**
Rouge-Lsum	45.91	**48.17**
BERTScore Precision	91.24	**91.87**
BERTScore Recall	91.47	**91.63**
BERTScore F1	91.35	**91.74**

Table 4: Comparison of models trained on 1000 training pairs. All number represented as percentages. Evaluated on the first 100 cases of MeetingBank summarization task after 10 epochs.

find that simply Numerical Disambiguity outperforms all other combinations in MeetingBank Summary, MeetingBank QA and GSM8K, with only marginally higher compression size than removing all punctuation, which is to be expected. We chose to display BLEU (Papineni et al., 2002), Rouge-Lsum (Lin, 2004) and BertScore-F1 (Zhang et al., 2019) for a varied evaluation through precision, recall and F1. All four prompt combinations are provided in Appendix A, along with a comprehensive list of all metrics used to evaluate the performance of MeetingBank summarization task.

We train our model with knowledge distilled data generated using our best performing prompt (ND), along with the default prompt of LLMLingua2 on both 120 training pairs and 1000 training pairs, for a total of four models. An example of inference results from distillation is included in Appendix D. From Figure 2, we find that our validation accuracy were consistently higher, even winning against LLMLingua2's prompting method on 1000 training pairs, while only using 120 training pairs. This signifies that data generated using our prompting method is faster and much easier to learn.

We evaluate models on the summarization task of MeetingBank, and see comparable performance in Table 3 and Table 4, with evaluation result from every epoch in Appendix B. We can extrapolate that with more data, our shorter prompting method is likely to have higher performance, as we can see from the difference between 120 and 1000 training data pairs. We have also established that shorter prompts converges faster in training, however it does not translate to the performance of evaluation, at least not with 120 to 1000 training pairs.

5 Conclusion

In this work, we introduced the idea that shorter prompts might bring higher performance when tailored to the problem. By leveraging LLMLingua2's knowledge distillation pipeline for a context compressor, we tried and succeeded at finding a 97% shorter prompt for generating compressed data with a 61% size reduction compared to LLMLingua2, while maintaining similar performance on summarization task, providing a interesting look into the world of less being more.

Limitations

Extractive compression is never completely lossless, and does come with potential risk of dropping critical contextual data if the model deems it unimportant. Users still need to validate the output using other methods, to make sure the result does not stray too far from the intended task.

Furthermore, due to the amount of LLM inferencing needed to generate compression pairs for model training, we were unable to train a LLM with training data on par with LLMLingua2. Similarly, we were only funded to evaluate Meetingbank, under the scope of summarization. Further study is needed to understand the generalization ability of this prompting method and its full potential.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. *Preprint*, arXiv:2301.00234.

Geoffrey Hinton. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Yebowen Hu, Tim Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. 2023. Meetingbank: A benchmark dataset for meeting summarization. *Preprint*, arXiv:2305.17529.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. *Preprint*, arXiv:1909.10351.

Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. 2022. Learned token pruning for transformers. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 784–794.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. *Preprint*, arXiv:1606.07947.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353, Singapore. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu, Sheng Shen, and Mirella Lapata. 2021. Noisy self-knowledge distillation for text summarization. *Preprint*, arXiv:2009.07032.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. *Preprint*, arXiv:2403.12968.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

V Sanh. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Zi Wang. 2021. Zero-shot knowledge distillation from a decision-based black-box model. In *International conference on machine learning*, pages 10675–10685. PMLR.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

A Prompt Combinations

Syntactical data is removed manually from the result of these two prompts to enforce Syntactical Removal for ND+SR and ND+DR+SR respectively.

- Numerical Disambiguity (ND):
  Remove 99% of total words.

  {text to compress}

  The compressed text is:

- Duplication Removal (ND+DR):
  1. Remove 99% of total words.
  2. Skip words that have already been used.
  3. Drop all punctuations.

  {text to compress}

  The compressed text is:

The comprehensive result of all metrics used for evaluating MeetingBank summarization task is displayed in Table 5. The prompt of LLMLingua2 is included below for ease of reference.
You are an excellent linguist and very good at compressing passages into short expressions by removing unimportant words, while retaining as much information as possible. Compress some text to short expressions, and such that you (GPT-4) can reconstruct it as close as possible to the original. Unlike the usual text compression, I need you to comply with the 5 conditions below: 1. You can ONLY remove unimportant words. 2. Do not change the order of words. 3. Do not change the original words, e.g. 'asking'->'ask' is NOT OK, 'current'->'now' is NOT OK. 4. Do not use abbreviations or emojis, e.g. 'without'->'w/o' is NOT OK, 'as soon as possible'->'ASAP' is NOT OK. 5. Do not add new words or symbols, this is very important. For example, 'dedicate 3 hours to each chapter'->'3 hours/chapter' is NOT OK because you add new token '/', just compress it into '3 hours each chapter'. '30 eggs plus 20 eggs equals 50 eggs'->'30+20=50' is also NOT OK becuase you add new symbols + and =, just compress it into '30 plus 20 equals 50'. Compress the origin aggressively by removing words only. Compress the origin as short as you can, while retaining as much information as possible.
If you understand, please compress the following text:
{text to compress}
The compressed text is:

B Visualized Results

Each model trained for 10 epochs, and was evaluated at each epoch on the MeetingBank summarization task. Evaluation results are visualized in Figure 3.

C Additional Research

This is the study that motivated the research into shorter prompts for better results. We previously found there was no drastic change in evaluation result for compressing and decompressing context using LLMs via different prompting methods. The results are displayed in Table 6, where baseline similarity is compared between 2 paragraphs generated by GPT-4o on the same topic prompt at default temperature, while evaluation similarity takes one paragraph, compress and decompress it using the given prompts below, and compare the paragraph before and after. Each paragraph is processed with text-embedding-ada-002 and compared using cosine similarity.

- MI+SE:
  Forgo formality and legibility and optimize the following for the least amount of tokens understandable by another instance:
  {text to compress}

- MI+FE:
  Forgo formality and legibility and optimize the following for the least amount of tokens understandable by another instance ("This argument lack the evidence needed to make a supporting statement">"argu -evi"):
  {text to compress}

- HI+FE:
  Optimize the following for the least amount of tokens understandable by humans ("This argument lack the evidence needed to make a supporting statement">"argu -evi"):
  {text to compress}

Comparison	LLMLingua2	ND	ND+DR	ND+SR	ND+DR+SR
BLEU	22.74	**23.64**	20.59	21.70	22.63
Rouge-1	**56.34**	54.88	53.30	54.44	55.67
Rouge-2	**28.87**	28.49	25.11	26.61	28.16
Rouge-L	40.92	**41.29**	37.59	39.49	39.93
Rouge-Lsum	41.03	**41.39**	37.86	39.81	40.13
BERTScore Precision	**90.80**	90.49	89.76	90.26	90.54
BERTScore Recall	89.90	**90.03**	89.46	89.81	89.83
BERTScore F1	**90.34**	90.25	89.60	90.02	90.17

Table 5: All metrics used to evaluate the first 100 cases of MeetingBank summarization task. All number represented as percentages.

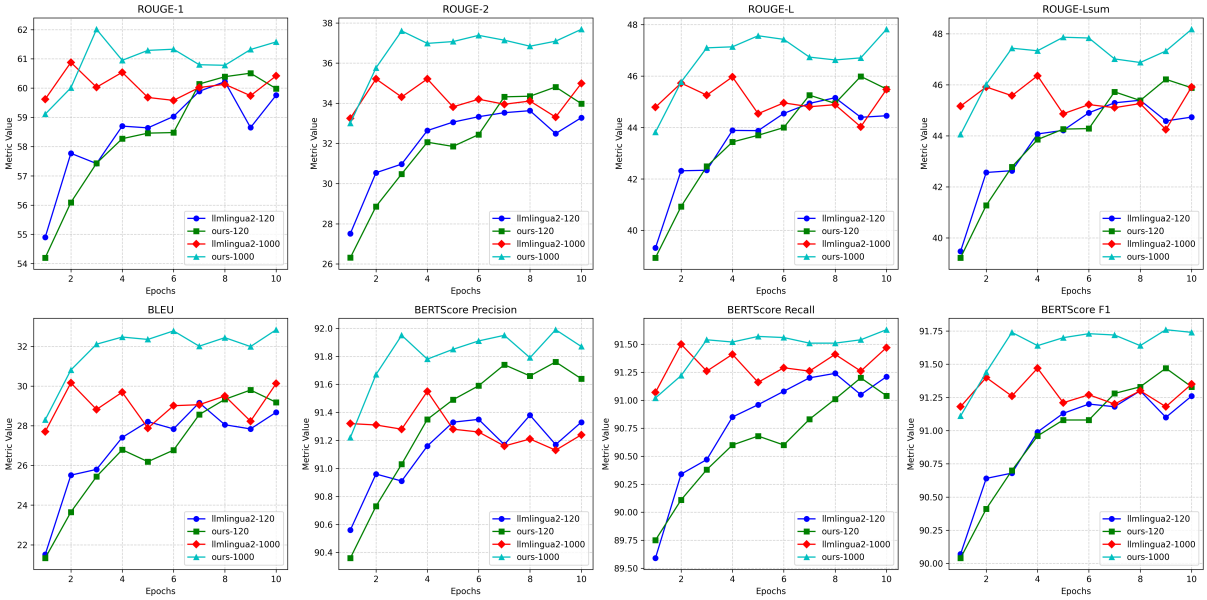

Figure 3: Evaluation results of our model and LLMLingua2 on 120 and 1000 training pairs. All metrics represented in percentages.

Method	Base Similarity	Eval Similarity
MI + SE	97.58%	97.72%
MI + FE	97.44%	97.46%
HI + FE	97.42%	97.64%
HI + SE	97.41%	97.87%

Table 6: Baseline similarity and evaluation similarity on the combination of Machine or Human Interpretable (MI/HI) and Self or Fixed Exploration (FE/SE).

Method	Cost Reduction	Time Reduction
MI + SE	34%	55%
MI + FE	31%	51%
HI + FE	32%	47%
HI + SE	22%	41%

Table 7: Percentage of cost reduced and percentage of time reduced on combinations of Machine or Human Interpretable (MI/HI) with Self or Fixed Exploration (FE/SE).

- HI+SE:
  Optimize the following for the least amount of tokens understandable by humans:
  {text to compress}

- Decompression prompt:
  Expand this paragraph to be approximately {word count of original paragraph} words long, while maintaining the key ideas from the abstracted paragraph: {text to compress}

From Table 7, we observe that the cost of these different prompting methods varied quite a bit, with the optimal choice being the shortest prompt. It was focused on letting the LLM discover the best way to complete the compression task so another instance could decompress it later, hence Machine-

Interpretable (MI) and Self-Exploration (SE). On the flip side was Human-Interpretable (HI) and Fixed-Exploration (FE), where we detailed instructions in the prompt for the LLMs to follow, and focus on the compressed result still being readable.

However, the worst performing combination was Human-Interpretable and Self-Exploration, perhaps due to the fact that instructing an LLM to behave like humans, without telling it how to, was the hardest task to complete out of these four.

D Example of Distillation Output

Input
Council is reconvene. We have one final vote to take on a previously previously amended council bill 161. Councilwoman Ortega was correct. Thank you, Councilwoman Ortega. We need a motion from you to order published council Bill 161 as amended. Thank you, Mr. President. I move that council bill 161 be ordered, published as amended. It has been moved and signed it. I will not let any comments or questions be made on this. Madam Secretary, Roll Call Clerk Espinosa. I Flynn. Are Gilmore. Herndon Cashman. I can eat. Can each name Lopez? I knew Ortega. Sussman? No. Black. Mr. President. I. I. Okay. Council Bill. 11 I's. Thank you. 11 I's two days. Council Bill has been ordered publish as amended. Final consideration of Council Bill 161. Public hearing will be Monday May days May 1st. So want to let the public know for the new council bill as amended, 161 will be May 1st.
LLMLingua2 Output
Council reconvene. Final vote on amended council bill 161. Councilwoman Ortega correct. Need motion to publish council Bill 161 as amended. Mr. President, move council bill 161 be ordered, published as amended. Moved and signed. No comments or questions. Secretary, Roll Call Clerk Espinosa. Flynn. Gilmore. Herndon Cashman. Lopez? Ortega. Sussman? No. Black. Mr. President. I. Council Bill. 11 I's. 11 I's two days. Council Bill ordered publish as amended. Final consideration of Council Bill 161. Public hearing Monday May 1st. Public know new council bill as amended, 161 May 1st.
ND Output
Council reconvenes, final vote on bill 161. Councilwoman Ortega moves to publish amended bill. No comments allowed. Roll Call, majority votes yes. Bill 161 ordered published, public hearing May 1st.

Table 8: Case study of inferencing GPT-4 with different compression prompts on training case 6.