CSLM: A Framework for Question Answering Dataset Generation through Collaborative Small Language Models

Yiming Wang, Yang Liu, Lingchen Wang, An Xiao*
Huawei Noah's Ark Lab
{wangyiming22, liuyang633, wanglingchen, an.xiao}@huawei.com

Abstract

Collecting high-quality question-answer (QA) pairs is vital for the training of large language models (LLMs), yet this process is traditionally laborious and time-intensive. With the rapid evolution of LLMs, the potential for leveraging these models to autonomously generate QA pairs has become apparent, particularly through the use of large-scale models like GPT-4. However, the computational demands and associated costs often render such approaches prohibitive for the average researcher. Addressing this gap, we introduce the **Collaborative Small Language Model Framework (CSLM)**, an innovative solution that combines a group of small-scaled, open-source LLMs to collaboratively produce QA pairs. Experiments on datasets of various domains show that CSLM unleashes the full potential of diverse small models to generate high-quality QA pairs, making it accessible to a broader range of researchers.

1 Introduction

The generation of high-quality question-answering (QA) pairs is crucial for enhancing the capabilities of language models across various applications. Despite the availability of general domain QA datasets, a significant gap exists in the availability of domain-specific datasets, such as law, medicine, and finance, etc. Manual annotation of such datasets is not only laborious and time-consuming but also entails substantial costs due to the specialized expertise required (Xie et al., 2023). Moreover, manual datasets such as SQuAD (Rajpurkar et al., 2016) often suffer from a lack of diversity, with answers being directly extracted from the source text without the nuance of deeper understanding or context, which restricts the potential of downstream models.

To address this, recent research has explored the use of large language models (LLMs) to synthesize QA pairs from documents or raw corpora (Wang et al., 2023; Lee et al., 2023; Wan et al., 2024). However, to generate high-quality QA pairs, large-scale models like Llama-70B(Touvron et al., 2023) or closed-source models like GPT-4 (OpenAI et al., 2024) are needed, while reliance on such models is not always feasible due to the substantial computational resource requirements. Furthermore, using external APIs, like GPT-4, to generate QA pairs introduces privacy and confidentiality concerns, especially when dealing with sensitive data in fields that demand stringent data protection measures.

In response to these challenges, we introduce the **Collaborative Small Language Model Framework (CSLM)** for generating QA pairs. By leveraging a group of smaller, open-source language models connected by a minimal number of extra trainable parameters, CSLM harnesses the unique strengths of each model to generate QA pairs that closely match the performance of larger models but with significantly reduced computational requirements. Furthermore, CSLM allows researchers to maintain control over their data within secure, internal environments, ensuring that sensitive information is protected.

We demonstrate the effectiveness of CSLM through extensive experiments on various domain-specific texts, showcasing its ability to generate accurate and diverse QA pairs that are not only of high quality but also respectful of privacy and confidentiality constraints, thereby lowering the barrier for organizations and researchers to generate their own datasets.

2 Collaborative Small Language Model Framework

2.1 Preliminary

In the context of domain-specific data, a QA pair typically consists of three components: a domain-relevant text passage $\mathcal{T}$, a question Q and an an-

*Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 11816–11825
November 12-16, 2024

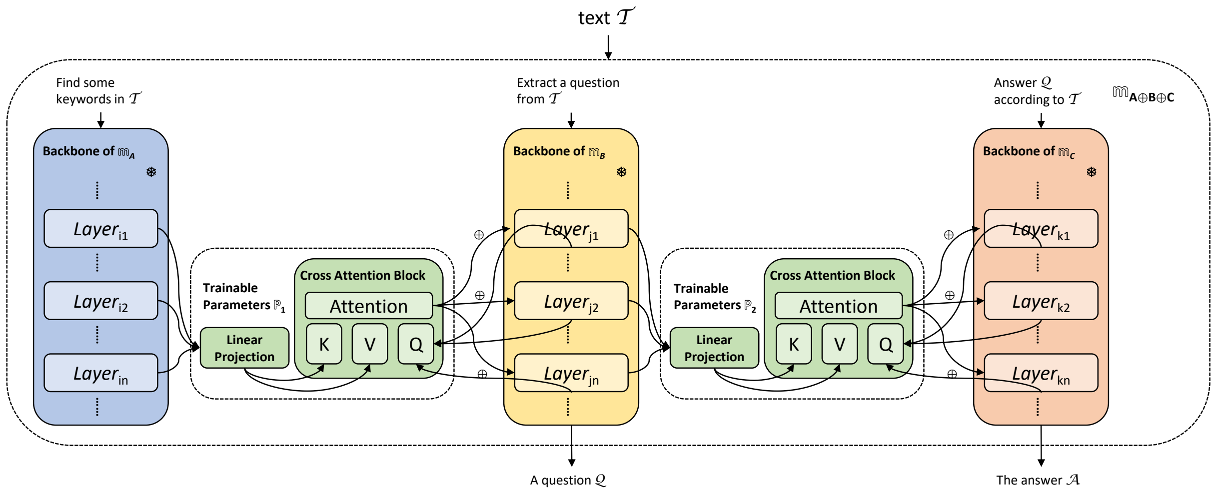

Figure 1: CSLM Framework: Illustrating the integration of multiple small language models through intermediate blocks for QA pair generation.

swer $\mathcal{A}$ that corresponds to $\mathcal{Q}$. The objective of generating QA pairs is to identify $\mathcal{Q}$ from $\mathcal{T}$ and subsequently derive $\mathcal{A}$ based on the content of $\mathcal{T}$, thereby ensuring the coherence and relevance of the generated pair.

2.2 Components of CSLM

To generate high-quality QA pairs with limited computational resource, we utilize the collective capabilities of multiple small language models. Taking cues from the CALM (Bansal et al., 2024), we select one model $\mathbb{m}_A$ as the primary augmenting model, whose role is to enhance the anchor model $\mathbb{m}_B$, in the task of question $\mathcal{Q}$ extraction. Concurrently, $\mathbb{m}_B$ plays the secondary role of an augmenting model, supporting $\mathbb{m}_C$ in formulating the answer $\mathcal{A}$. This dual-augmentation strategy is designed to amplify the individual strengths of each model, thereby enhancing the overall QA pairs generation process.

Figure 1 illustrates the integration of the intermediate blocks between the model pairs $\mathbb{m}_A$ and $\mathbb{m}_B$, as well as $\mathbb{m}_B$ and $\mathbb{m}_C$, which facilitates iterative interactions during inference to refine the models' collaborative output. The interactions occur at some selected layers as indicated in 2.2.1 and the intermediate blocks consist of two major components: (i) Linear projection block. (ii) Cross-attention block.

2.2.1 Interaction Layer Selection

We carefully select a subset of layers from each model, ensuring a uniform distribution across the models to maintain consistency in the interaction process. Assume that $\mathbb{m}_A$, $\mathbb{m}_B$ and $\mathbb{m}_C$ has N_A,N_B and N_C hidden layers respectively. We first select a subset of n layers $\mathbb{L}_A = \{i_1, i_2, ..., i_n\}$, $\mathbb{L}_B = \{j_1, j_2, ..., j_n\}$ and $\mathbb{L}_C = \{k_1, k_2, ..., k_n\}$ from each model. The intervals between consecutive selected layers also remain consistent, which means $i_n - i_{n-1} = j_n - j_{n-1} = k_n - k_{n-1} = min(N_A, N_B, N_C)//n$, facilitating a balanced and structured integration of model layers.

2.2.2 Linear Projection Block

This block aligns the hidden states of different models by mapping them to a common representation space. Let $R \in \mathbb{R}^{B*H*D}$ represent the hidden states within a model, where B, H, D correspond to the batch size, the number of attention heads, and the dimensionality of each hidden state. Between each pair of models, we introduce a linear projection function:

$$f_{proj}(R_f) = R_{mid},$$
$$R_f \in \mathbb{R}^{B\times H\times D_f},\ R_{mid} \in \mathbb{R}^{B\times H\times D_l}$$

which maps the former model's hidden states to the representation dimensionality of the latter model. This block facilitates cross-attention between models that possess hidden states of varying sizes, aligning their representations to ensure compatibility without re-training the original models.

2.2.3 Cross Attention Block

We introduce cross-attention module between each pair of models, which is calculated using the mid-representation R_{mid} as key and $value$ vectors, with the layer representation R^l from the latter model as the $query$ vectors:

$$K,\ V = R_{mid}W^K,\ R_{mid}W^V$$

$$Q = R_L W^Q$$

$$f_{cross} = Attn(Q, K, V)W^O$$

W^Q, W^K, W^V and $W^O \in \mathbb{R}^{D_l \times D_l}$ are trainable weights. The resulting attention-weighted outputs f_{cross} derived from the i_{th} layer are then integrated into the subsequent layers of the latter model, thereby enhancing the models' mutual understanding and integration of information.

2.3 Unified Model for QA Pair Generation

CSLM integrates these three models through a unified function $\mathbb{m}_{A \bigoplus B \bigoplus C} = f(\mathbb{m}_A, \mathbb{m}_B, \mathbb{m}_C, \mathbb{P})$, where $\mathbb{P}$ represents a small set of trainable parameters in the intermediate blocks. To elevate the ability of the collaborative models in QA pairs generation, we fine-tune the connecting parameters $\mathbb{P}$ using a small amount of data, with the weights of $\mathbb{m}_A$, $\mathbb{m}_B$ and $\mathbb{m}_C$ frozen. During QA pairs generation, Model $\mathbb{m}_A$ identifies keywords in the original text and $\mathbb{m}_B$ to extract a question based on the important parts. Then, the question, along with the focused attention from $\mathbb{m}_A$, is channeled through the interaction blocks to $\mathbb{m}_C$, which leverages this enriched context to formulate a precise and relevant answer. The prompts used in each model are listed in Appendix A.

[t]

3 Experiments

3.1 Experiments Setup

We select five distinct corpora representing general, medical, financial and nuclear domains. These include MS_MARCO(Nguyen et al., 2016) and SQuAD1.1 for general domain, Asclepius-Synthetic-Clinical-Notes (ASCN)(Kweon et al., 2023) for medical domain, Financial-Articles (Lettria, 2024) for finance domain and Nuclear-Patent (Arcee-AI, 2023) for science domain. The details of these datasets are shown in Appendix B, with which we can ensure a comprehensive evaluation of CSLM's capability.

3.2 Implementation Details

In CSLM framework, We select three smaller-scale, open-source language models: TinyLLama-1.1B (Zhang et al., 2024), QWen1.5-1.8B (Bai et al., 2023) and InternLM2-1.8B (Cai et al., 2024), denoted as $\mathbb{m}_A$, $\mathbb{m}_B$ and $\mathbb{m}_C$, respectively. To connect these models, we introduce intermediate blocks at the $5_{th}, 10_{th}, 15_{th}, 20_{th}$ layers. In total we introduce 40 million additional trainable parameters to our collaborative models, facilitating efficient training compared to the overall 4.6 billion original parameters that are kept frozen. The few number of new parameters allows for training with a modest dataset over a few epochs. For QA generation in all domains, we only use 500 general text QA pairs and adopt a five epoch training for the intermediate blocks to enable the collaborative model instill basic linguistic capabilities.

3.3 Evaluation Metrics

Evaluating the quality of QA pairs generated by language models encompasses a multifaceted assessment, extending beyond traditional metrics like ROUGE(Lin, 2004), due to the content of QA pairs that are not merely text extractions.

Recently, using LLMs for automatic evaluation of generated data has gradually matured and been widely applied, such as overall scoring(Fu et al., 2023), evaluation paradigms (Lin and Chen, 2023) and COT(Liu et al., 2023). Among these, the RACAR metric, a five-dimensional metric crafted to evaluate the quality of generated QA pairs, introduced by SciQAG (Wan et al., 2024), stands out for its comprehensiveness. Therefore, we adopt a similar automatic evaluation approach using a leading large language model, including four various aspects to assess the QA pair quality, which correlates more closely with human judgement.

Relevance. This dimension evaluates how closely the generated QA pairs align with the original text, ensuring that the content is contextually appropriate.

Comprehensiveness. This dimension measures how well the generated answer encompasses all necessary details from the question and the source text, thereby ensuring thoroughness.

Correctness. This dimension assesses the fidelity of the generated answer to the information presented in the source text, highlighting the importance of factual accuracy.

Coherence. This dimension evaluates whether the generated QA pair is free from contradictions and follows a clear, reasonable structure.

Each of these dimensions is scored on a scale from 1 to 3, with the higher scores indicating better performance in generating QA pairs that are not only accurate but also contextually rich and logically sound. The prompts for evaluation are presented in Appendix C.

Metric	Model / Dataset	InternLM2-1.8B	QWen1.5-1.8B	QWen1.5-4B	InternLM2-7B	QWen1.5-7B	LLaMA3-8B	Ushio et al., 2023	CSLM
Relevance	MS_MARCO	2.41	1.90	2.09	2.28	1.91			**2.61**
	SQuAD	2.46	1.06	2.25	2.29	2.17	2.43	1.87	**2.68**
	ASCN	1.32	1.10	2.18	2.20	2.24			**2.46**
	Nuclear	2.32	2.09	2.09	2.40	2.00			**2.58**
	Financial	2.27	1.98	2.02	2.11	1.88			**2.54**
Comprehensiveness	MS_MARCO	2.75	2.47	2.66	2.75	2.67			**2.89**
	SQuAD	2.70	1.10	2.51	2.80	2.54	2.84	1.35	**2.92**
	ASCN	1.08	1.15	2.79	2.38	2.64			**2.67**
	Nuclear	2.50	2.33	2.53	2.62	2.61			**2.88**
	Financial	2.75	2.64	2.79	2.85	2.64			**2.88**
Correctness	MS_MARCO	2.75	2.47	2.66	2.75	2.58			**2.88**
	SQuAD	2.73	1.11	2.60	2.78	2.51	2.85	1.39	**2.92**
	ASCN	1.10	1.15	**2.79**	2.39	2.64			2.73
	Nuclear	2.55	2.31	2.54	2.64	2.63			**2.87**
	Financial	2/73	2.53	2.76	2.80	2.57			**2.86**
Coherence	MS_MARCO	2.45	2.17	2.47	2.48	2.50			**2.70**
	SQuAD	2.44	1.25	2.42	2.45	2.15	**2.62**	1.51	2.60
	ASCN	1.10	1.21	2.64	2.38	2.40			**2.68**
	Nuclear	2.44	1.93	2.37	2.37	2.58			**2.66**
	Financial	2.47	2.13	2.46	2.49	2.52			**2.65**

Table 1: LLM evaluation of generated QA pairs: Performance metrics across 5 different domains highlighting Relevance, Comprehensiveness, Correctness, and Coherence.

3.4 Experimental Results

Table 1 offers a comprehensive evaluation of the QA pairs generated across five distinct domains using the CSLM framework. Notably, the CSLM model, with approximately 4.6B parameters, is compared to several other LLMs with less than 7B parameters, including InternLM2-1.8B(Cai et al., 2024) , QWen-1.5-1.8B(Bai et al., 2023), and their counterparts at 4B and 7B parameter sizes. The comparison result between CSLM and an established question generation method mentioned in Ushio et al., 2023 as well as a stronger model, LLaMA3-8B (Dubey et al., 2024) on SQuAD datasets is also shown in Table 1 which proves CSLM surpasses the traditional model and some stronger language model in the domain of QA pairs generation.

Table 2 shows examples of QA pair generated by CSLM and other methods to illustrate the alignment between human evaluations and RACAR metric. It is obvious that other baseline models make errors in logic and fact, while CSLM successfully synthesizes the QA pair. Meanwhile, we can find that the answer generated by Ushio et al., 2023 can only be extracted directly from the text which makes it pretty rigid. Besides, Ushio et al., 2023 can not fully use the information in the text and sometimes will even generate a wrong answer when the generated question should be answered by summarizing the text. Thus CSLM is a better method to generate QA pairs. And the QA pair generated by CSLM achieves the highest score on the automatic evaluation, aligning with human judgment.

We also conduct pairwise comparison on SQuAD datasets, which asks the judging model to rank the QA pairs generated by different models. Table 3 shows the average ranking on the four dimensions.

These results reveal a compelling advantage of the CSLM framework. Across all four dimensions of the evaluation metric, our collaborative models consistently outperform the individual LLMs, even surpassing models with larger parameters, like 7B. This indicates that the CSLM framework sufficiently utilizes the collective strengths of its constituent models, thereby achieving a better QA pairs generation that not only matches but exceeds the capabilities of larger-scale, single LLMs.

We also compare the inference speed of CSLM with other models. Based on generating 1000 QA pairs on a single GPU, CSLM takes 38 minutes while InternLM-7B costs 100 minutes and QWen1.5-4B costs 52 minutes. This demonstrates that CSLM surpasses the components of it and larger models not only in terms of generation quality but also in generation speed.

3.5 Ablation Study

Scalability. We test the flexibility and scalability to incorporate larger-scale models of CSLM framework. We integrate Yi-1.5-6B(AI et al., 2024), InternLM2-7B, and QWen1.5-7B, against InternLM2-20B, and conduct QA pairs generation on MS_MARCO dataset. The results in Table 4 confirm that CSLM maintains its capability when scaled up, showcasing its versatility in collaboration with LLMs of varying sizes.

Example 1
Original Text: Bacon that has been defrosted in the fridge can be kept for an additional 1 to 2 days in the refrigerator before cooking; bacon that was thawed in the microwave or in cold water should be cooked immediately.
CSLM **Q:** How long can you keep defrosted bacon in the fridge before cooking? **A:** Defrosted bacon can be kept for an additional 1 to 2 days in the fridge before cooking.
InternLM2-7B **Q:** How long can bacon be stored in the refrigerator after defrosting in the microwave or cold water? **A:** Bacon that has been defrosted in the fridge can be kept for an additional 1 to 2 days in the refrigerator before cooking; bacon that was thawed in the microwave or in cold water should be cooked immediately.
QWen1.5-7B **Q:** How long can bacon be stored in the refrigerator after being frozen and thawed, and what should be done with it before cooking? **A:** Bacon that has been frozen and thawed can be stored in the refrigerator for an additional 1 to 2 days before cooking, and it should be cooked immediately after thawing.
Example 2
Original Text: A GMC truck with a standard cab but a long bed is more than 18 1/2 feet in total length. A truck with a double cab and a regular bed is slightly more than 19 feet long. If a pickup has a larger crew cab with a standard box, the vehicle is about 20 feet long. However, if the truck has a crew cab with a short box, the entire length is a little more than 19 feet.
CSLM: **Q:** What is the difference between a GMC truck with a standard cab and a GMC truck with a double cab? **A:** The difference between a GMC truck with a standard cab and a GMC truck with a double cab is that the standard cab has a longer length of 18 1/2 feet, while the double cab has a slightly longer length of 19 feet.
Ushio et al., 2023: **Q:** What is the total length of a GMC truck with a standard cab and a long bed? **A:** 18 1/2 feet

Table 2: Examples of QA pair generated by CSLM and other methods

	InternLM2-1B	InternLM2-7B	QWen1.5-1B	QWen1.5-4B	QWen1.5-7B	CSLM
Relevance	4.70	4.46	3.68	3.24	3.63	**1.28**
Comprehensiveness	4.52	3.82	3.37	2.97	**2.84**	3.49
Correctness	3.24	2.49	5.52	4.61	3.89	**1.25**
Coherence	3.49	2.97	4.69	4.33	3.84	**1.69**

Table 3: Pairwise comparison assessment of CSLM framework with other models. A lower score indicates a higher ranking.

Metric \ Model	Yi-1.5-6B	InternLM2-7B	QWen1.5-7B	InternLM2-20B	CSLM
Relevance	2.01	2.66	2.27	2.57	**2.84**
Comprehensiveness	2.02	2.85	2.80	2.88	**2.96**
Correctness	2.02	2.82	2.48	2.90	**2.96**
Coherence	2.11	2.41	2.54	2.62	**2.81**

Table 4: Scalability assessment of CSLM framework with larger-scale models

Components. We conduct an ablation study on the MS_MARCO dataset (Table 5) to isolate the impact of each trainable intermediate block within CSLM. We drop the trainable intermediate parameters while keep the LLMs. The findings underscore the importance of these blocks, as their removal leads to a significant decrease in performance across most dimensions.

Metric \ Model	without $\mathbb{P}_1$	without $\mathbb{P}_2$	without $\mathbb{P}_1$ and $\mathbb{P}_2$	CSLM
Relevance	2.42	2.89	2.10	**2.93**
Comprehensiveness	2.68	2.73	2.11	**2.93**
Correctness	2.67	2.73	2.12	**2.89**
Coherence	2.43	2.43	2.29	**2.82**

Table 5: Ablation study on CSLM's trainable intermediate blocks

4 Conclusion

This paper introduces the Collaborative Small Language Model Framework (CSLM), an innovative approach that uses multiple smaller, open-source language models to achieve a performance comparable to larger models, yet with less computational cost. Our extensive experiments across various domains have demonstrated the robustness and efficacy of CSLM in QA pairs generation, offering a viable alternative to large-scale language models. In general, this study not only offers a novel perspective on the data generation field but also presents a viable solution for researchers applying LLMs with limited computational resource.

References

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

Arcee-AI. 2023. nuclear_patents. `https://huggingface.co/datasets/arcee-ai/nuclear_patents/discussions`.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Rachit Bansal, Bidisha Samanta, Siddharth Dalmia, Nitish Gupta, Sriram Ganapathy, Abhishek Bapna, Prateek Jain, and Partha Talukdar. 2024. LLM augmented LLMs: Expanding capabilities through composition. In *The Twelfth International Conference on Learning Representations*.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. 2024. Internlm2 technical report. *Preprint*, arXiv:2403.17297.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor

Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *Preprint*, arXiv:2302.04166.

Sunjun Kweon, Junu Kim, Jiyoun Kim, Sujeong Im, Eunbyeol Cho, Seongsu Bae, Jungwoo Oh, Gyubok Lee, Jong Hak Moon, Seng Chan You, Seungjin Baek, Chang Hoon Han, Yoon Bin Jung, Yohan Jo, and Edward Choi. 2023. Publicly shareable clinical large language model built on synthetic clinical notes. *Preprint*, arXiv:2309.00237.

Seongyun Lee, Hyunjae Kim, and Jaewoo Kang. 2023. Liquid: A framework for list question answering dataset generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):13014–13024.

Lettria. 2024. Financial-articles. `https://huggingface.co/datasets/Lettria/financial-articles`.

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

Yen-Ting Lin and Yun-Nung Chen. 2023. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. *Preprint*, arXiv:2305.13711.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *Preprint*, arXiv:2303.16634.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Asahi Ushio, Fernando Alva-Manchego, and Jose Camacho-Collados. 2023. An empirical comparison of lm-based question and answer generation methods. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14262–14272.

Yuwei Wan, Aswathy Ajith, Yixuan Liu, Ke Lu, Clara Grazian, Bram Hoex, Wenjie Zhang, Chunyu Kit, Tong Xie, and Ian Foster. 2024. Sciqag: A framework for auto-generated scientific question answering dataset with fine-grained evaluation. *Preprint*, arXiv:2405.09939.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Tong Xie, Yuwei Wan, Wei Huang, Zhenyu Yin, Yixuan Liu, Shaozhou Wang, Qingyuan Linghu, Chunyu Kit, Clara Grazian, Wenjie Zhang, et al. 2023. Darwin series: Domain specific large language models for natural science. *arXiv preprint arXiv:2308.13565*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *Preprint*, arXiv:2401.02385.

5 Limitation

(1) Owing to constraints in computational resource, the validation of the generated QA pairs in downstream tasks was not conducted, nor was the potential of integrating larger-scale models explored. Future studies should consider the application of CSLM in targeted downstream applications and investigate the performance of model ensembles of varying scales. (2) The sequence and synergy among the models within the CSLM framework remain insufficiently studied. The optimization of model collaboration requires further exploration.

A Appendix of Input Prompts to CSLM

(i) Prompt inputs to $\mathbb{m}_A$: Please find some keywords from the following text.

Text -Start

Text -End

Limitation: Please only reply keywords extracted form the text without any other information.

(ii) Prompt inputs to $\mathbb{m}_B$: Please generate a question about the provided paragraph.

Limitation: please only reply a question without any other additional information and do not answer the question. Here is the paragraph:

Paragraph -Start

Paragraph -End

(iii) Prompt inputs to $\mathbb{m}_C$: Please answer the question according to the following paragraph.

Limitation: 1. Please answer the question in one sentence.

2. Please only use the information in the paragraph to answer the question.

Paragraph -Start

Paragraph -End

Question -Start

Question -End

B Appendix of Experimental Datasets Details

Dataset	Domain	Origin passages amount	Chosen passages and generated QA pairs amount
MS_MARCO	General	over 1000000	10000
SQuAD 1.1	General	18895	10000
ASCN	Medical	158000	10000
Nuclear	Science	33500	5000
Financial	Finance	18400	5000

Table 6: Details of experimental datasets

C Appendix of Evaluation Prompts

(i) Prompt for **Relevance** evaluation: Given a paragraph of text and questions generated from it, evaluate the relevance of the question to the text and return a score ranging from 1–3 and give reasons as to why this score was assigned. The output must be a list of dictionaries corresponding to each question, with the fields 'score' and 'reasons'. If the question does not pertain to the text, assign a score of 1.

(ii) Prompt for **Comprehensiveness** evaluation: Given a paragraph of text and question answer pairs generated from it, evaluate the completeness of the answer for each question and return a score ranging from 1–3 indicating the extent to which the answer fully addresses the question using the information in the paper, including all subquestions. Also give reasons for assigning the score. The output must be a list of dictionaries for each question answer pair, with the fields 'score' and 'reasons'.

(iii) Prompt for **Correctness** evaluation: Given a paragraph of text and question answer pairs generated from the text, evaluate the accuracy of the answer for each question and return a score ranging from 1–3 indicating whether the answer is accurately extracted from the text and give reasons as to why this score was assigned. This involves checking the accuracy of any claims or statements made in the text, and verifying that they are supported by evidence. The output must be a list of dictionaries for each question answer pair, with the fields 'core' and 'reasons'.

(iv) Prompt for **Coherence** evaluation: Given a paragraph of text and statements, evaluate the reasonableness of the statements with respect to the text and return a score ranging from 1–3 indicating how logically consistent the content is, with no obvious contradictions and provide reasons for assigning the score. The output must be a list of dictionaries for each statement, with the fields 'score' and 'reasons.' Assign a score of 1 if the statement has logical error like contradicts.