Automatically Generating Custom Context-Driven SFT Data for LLMs with Multi-Granularity

Shanghaoran Quan

Beihang University
shrquan@buaa.edu.cn

Abstract

Constructing high-quality query-response pairs from custom corpora is crucial for supervised fine-tuning (SFT) large language models (LLMs) in many applications, like creating vertical-domain AI assistants or roleplaying agents. However, sourcing this data through human annotation is costly, and existing automated methods often fail to capture the diverse range of contextual granularity and tend to produce homogeneous data. To tackle these issues, we introduce a novel method named AUGCON, capable of **au**tomatically **g**enerating **con**text-driven SFT data across multiple levels of granularity with high diversity, quality and fidelity. AUGCON begins by generating queries using the Context-Split-Tree (CST), an innovative approach for recursively deriving queries and splitting context to cover full granularity. Then, we train a scorer through contrastive learning to collaborate with CST to rank and refine queries. Finally, a synergistic integration of self-alignment and self-improving is introduced to obtain high-fidelity responses. The results highlight the significant advantages of AUGCON in producing high diversity, quality, and fidelity SFT data against several state-of-the-art methods.

1 Introduction

With the rise of impressive capabilities of large language models (LLMs), a variety of vertical-domain LLM-based AI assistants have been introduced [15, 13, 80, 25, 41]. By incorporating specialized knowledge into LLMs, these custom models have been shown to outperform their general-purpose counterparts in their respective areas. Directly supervised fine-tuning on the raw, custom corpora, also known as domain-adaptive pre-training (DAPT) [24], has proven beneficial [9, 33] but revealed to be insufficient and may impair prompting ability on domain-specific tasks [48, 56].

To better leverage the privatized knowledge and customize the outputs of LLMs, supervised fine-tuning using custom query-response pairs has become common practice [65, 12, 10, 20, 83]. Since sourcing these pairs through human annotation is very costly and can't generate at scale, recent studies have explored automated methods for creating these pairs from custom corpora [15, 28, 75]. However, those existing methods using the same workflow repeatedly on the same context tend to produce redundant queries without adequately covering the entire context at various levels of granularity. To automatically construct synthetic custom SFT data incorporating a wide range of contextual **granularity** (queries range from detailed questions to macro topics) with high **diversity** (queries need to be diversified to cover as much as possible the provided corpus), **quality** (responses are correct and efficient in answering the queries), and **fidelity** (data needs to follow human values and conform to predetermined tone and formats) still remain challenges.

To address these challenges, we propose AUGCON, which automatically generates multi-granularity context-driven SFT data for LLMs at scale with high diversity, quality, and fidelity. AUGCON performs the following three essential steps:

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

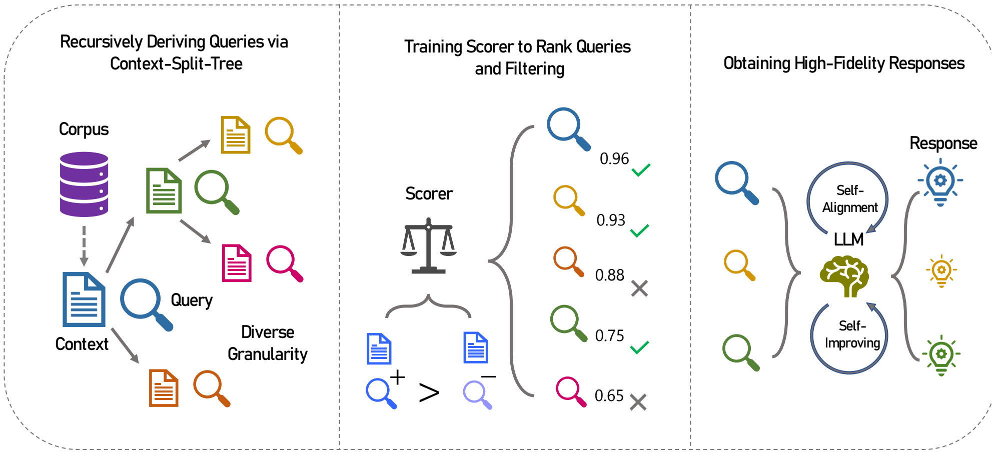


Figure 1: An overview of the proposed AUGCON.

1. **Recursively Deriving Queries via Context-Split-Tree:** Considering that it is challenging for predetermined prompts to generate diverse queries with broad granularity from the same context, we propose a novel method called Context-Split-Tree (CST). Each context will recursively continue to derive queries that matches context granularity and splits until it cannot be further divided.
2. **Contrastive-Learning-Based Scorer to Rank Queries and Filtering:** We use contrastive learning to train a scorer to sort the derived queries under the same context, and we only retain the queries that get high scores and the diversity reaching a specific threshold.
3. **Obtaining High-Fidelity Responses:** We employ a principle-driven self-alignment approach to guide the LLM in producing high-fidelity responses to filtered queries and apply random search to conduct the LLM to self-evaluate its responses and discover the best in-context learning prompt.

The entire process only requires a handful of few-shot CST examples, alignment principles, and query response examples. We can also achieve impressive results by just utilizing the open-source model, which will later be fine-tuned with synthetic data, eliminating the necessity of distilling more powerful LLMs like ChatGPT.

To assess the efficacy of our approach, we meticulously construct a test scenario and carefully assemble a dataset consisting of high-quality Chinese magazine articles centered around daily topics, along with corresponding test queries. Human evaluation demonstrates that our method excels in generating queries of superior quality and in enhancing the performance of fine-tuned models. Additionally, automatic evaluations conducted on four popularly used English benchmarks with relevant metrics further highlight the significant advantages our method holds in capturing contextual knowledge when compared to other state-of-the-art context-driven SFT data generation approaches.

2 Our Method: AUGCON

2.1 Recursively Deriving Queries via Context-Split-Tree

This step is to derive context-query pairs (C, q) from the given corpus $\mathcal{C}$. Previous approaches applied regex-based or predetermined prompts for query generation, which often led to queries that were relatively monotonous in structure and granularity. We believe that this type of approach did not fully exploit the context, leading to queries incapable of effectively provoking the model's capability to comprehend and differentiate between various levels of detail within the context, resulting in suboptimal outcomes.

To address this issue, we propose a very novel and effective method called Context-Split-Tree (CST), with the pseudocode shown in Algorithm 1. CST starts with an entire context C, with each attached with the instruct prompt I_{CST} and few-shot examples E_{CST} to call an LLM to generate a query q deriving from the entire context. At the same time, we ask the LLM to semantically divide the context into two child contexts C_1 and C_2, and the instruct prompt is designed with hints to let the LLM polish the two split contexts to make them as independent as possible and collectively encompass the entirety of the original context. Each child context will continue to recursively derive query and split until reaching a point where one of its split child context lengths is not less than itself or the length falls below a predetermined threshold λ. At this point, we consider it to have been split into the minimum granularity and cannot be further divided. Upon the completion of this recursive process, a binary tree structure is formed, with the initial context at the root, and each node representing a context along with its corresponding query tailored to its specific granularity. We collect data from all nodes as the outcome of this step. The detailed prompt templates and several case demonstrations are attached in Appendix B.1.2 and D, respectively.

The minimum length threshold λ and the initial context length l are like the lower bound and upper bound to control the granularity distribution of generated questions. One can easily adjust the overall average granularity of generated queries by adjusting the length threshold. Similarly, if we seek to address more global questions, we can do it by simply increasing the initial context length, as long as the model's context window permits.

2.2 Contrastive-Learning-Based Scorer to Rank Queries and Filtering

To further enhance the quality and diversity of the generated data, we introduce an effective ranking and filtering strategy collaborating with CST. Previous works have attempted to filter training data via heuristic algorithms, such as filtering out queries that are too long or too short [74]. Other works that are more relevant to us attempt to train scorers to judge the complexity and quality of question-response pairs [44], but they need to have a step of distillation on stronger LLM APIs like ChatGPT, and their training methods are less effective. For example, they put a series of responses and ask for direct ranking, suffering from the positional bias [42] in LLMs, or ask LLMs to directly assign a scalar score to a response, which is unstable. In this work, we apply contrastive learning to train a scorer to judge the degree of adherence to instruct prompt and few-shot examples, which is data-efficient and can achieve effective performance without the need for stronger LLMs.

The structure of our scorer is obtained by adding a linear head after the base model to map the last hidden state to a one-dimensional space. We take context-query pairs as inputs, applying scorer Sc to yield a scalar score $s = Sc(C, q)$. We use the context query pairs obtained from Step 1 as positive samples: $q^+ = \text{LLM}(I_{\text{CST}}, E_{\text{CST}}, C)$, and obtain negative samples by manipulating the instruct prompt (use suboptimal instructions): $q^- = \text{LLM}({I_{\text{CST}}}^-, E_{\text{CST}}, C)$, few-shot examples (reduce ICL examples count): $q^- = \text{LLM}(I_{\text{CST}}, {E_{\text{CST}}}^-, C)$ or both of them: $q^- = \text{LLM}({I_{\text{CST}}}^-, {E_{\text{CST}}}^-, C)$. The details of constructing positive-negative pairs are presented in Appendix B.2.1. Note that we do not generate all corresponding negative examples for positive data for training scorer, but rather randomly select a very small number of samples (*e.g.* only 500 pairs for each negative types in our implementation) to form the training set D_{Sc}. Then, the loss function of scorer can be represented as:

$$\mathcal{L} = -\mathbb{E}_{(C,q^+,q^-)\sim D_{Sc}}[\log(\sigma(Sc(C, q^+) - Sc(C, q^-))))] \tag{1}$$

We use the trained scorer applied on all the context query pairs obtained in Step 1 to get their scores. For each root context, we rank all queries from its CST in descending order of scores. Then, we start with an empty set and add one training query each time, only if the current query has a ROUGE-L precision score of less than 0.7 compared to any previously added queries. We will stop adding as the count reaches the limit. Each context will form such a set, and ultimately, we consolidate and retain the training data from all the sets. Through this approach, we can obtain diverse data and easily control the quantity, for it makes it possible to apply multi-times CST in the same context and filter the repeated one. The details of how the filtering pipeline cooperates with CST to improve the quality and diversity of queries are expatiated in Appendix B.2.2.

2.3 Obtaining High-Fidelity Responses

Inspired by the significant impact principles [69, 68] have on LLMs, this principle-driven self-alignment step begins by appending the context and a set of helpful, ethical, and reliable principles to the LLM. These principles are meticulously crafted to ensure the LLM's outputs are closely aligned with human preferences or mimic certain response tones. Before initiating the response generation, we deploy a self-improving pipeline that makes the LLM self-evaluate its response and sift through the entire set of human-annotated Q&A pairs E_R, where random search is applied to find the most fitting few-shot examples to help LLM generate high-fidelity responses under the predetermined principles, denoted as ${E_R}'$. The detailed implementation is shown in Appendix B.3.

Our innovative synergistic integration of the principle-driven self-alignment with self-improving methodology effectively improves the fidelity of generated responses. Following this, we execute LLM($I_R, {E_R}', C, q$) to elicit each response r, ensuring that each response is not only in high quality but also in good alignment with the established principles. Notably, due to the precise matching of each query with its context's granularity within the CST framework, the LLM can effortlessly provide accurate and pertinent responses to the queries.

After obtaining all generated data, we prune all context, instruction, and response principles and only retain synthetic query response pairs as SFT data. This approach allows the fine-tuned LLM to potentially learn the methods and nuances of responding to queries in a manner that naturally aligns with human expectations, enabling the LLM to directly generate responses that are well-aligned with reliable principles and optimal ICL exemplars across a wide range of queries. It's important to note that the fine-tuned LLM can generate high-quality responses without the need to explicitly reference the principles set and ICL exemplars.

3 Evaluations

To demonstrate the advantages of our method, we meticulously collect the following relevant baselines: (1)Chat Model [7, 71], (2)DAPT [24], (3)AdaptLLM [15], (4)ETRC [28], (5)Context-Instruct [75]. The set of contexts, base language models, and quantity of retained query-response pairs are maintained the same (if applicable) on both the baselines and our method to ensure a fair comparison.

We also notice that there are several alternative methodologies such as RAG and long context LLMs, but we don't compare them as we have a huge difference both in training and inference [53, 39]. We encourage interested readers to refer to Section A for more relevant information.

For datasets featuring short-form responses (applied to the SQuAD1.1 [60], TriviaQA [30], and DROP [19] datasets), we measure the model's performance using exact matching (EM) accuracy. For datasets with long-form responses (applied to the WebGLM-QA [45] dataset), we employ BERTScore (BS) [88] (we use Roberta-Large [47] for calculation) to evaluate the semantic similarity between the generated outputs and the reference responses.

Method	Short-form (EM) SQuAD1.1	TriviaQA	DROP	Long-form (BS) WebGLM-QA
Llama3-c$_{70B}$	$0.212_{\pm 0.004}$	$0.723_{\pm 0.003}$	$0.220_{\pm 0.004}$	$0.837_{\pm 0.002}$
DAPT	$0.258_{\pm 0.004}$	$0.767_{\pm 0.003}$	$0.266_{\pm 0.004}$	$0.851_{\pm 0.002}$
AdaptLLM	$0.273_{\pm 0.003}$	$0.791_{\pm 0.004}$	$0.284_{\pm 0.003}$	$0.842_{\pm 0.001}$
ETRC	$0.301_{\pm 0.004}$	$0.812_{\pm 0.003}$	$0.326_{\pm 0.004}$	$0.903_{\pm 0.001}$
Context-Instruct	$0.314_{\pm 0.003}$	$0.825_{\pm 0.003}$	$0.334_{\pm 0.003}$	$0.885_{\pm 0.001}$
AUGCON*(Ours)*	$\mathbf{0.336}_{\pm 0.004}$	$\mathbf{0.849}_{\pm 0.003}$	$\mathbf{0.350}_{\pm 0.003}$	$\mathbf{0.924}_{\pm 0.002}$

Table 1: The results of automatic evaluation on four benchmarks. We run 10 times for each test and report the mean value and standard deviation, with the best results shown in bold.

We use Llama3-70B-Instruct [4] as the base model for calling and conducting fine-tuning for automatic evaluations for all our baselines and the proposed AUGCON. The detailed results are shown in Table 1. The results illustrate that our proposed method consistently outperforms the established baselines across all four datasets. Specifically, when analyzing short-form datasets, it becomes evident that the data generated by AUGCON surpasses the comparative methods in extracting pivotal information and knowledge from the corpus, thus improving the question-answering accuracy of fine-tuned models.

3.1 Abalation Study

We develop four distinct variations: (1) $\textbf{AUGCON}^{w/o}_{\textbf{CST1}}$ drops the CST part and replaces it by directly iteratively deriving queries from the extracted context until the desired number of queries is obtained. (2) $\textbf{AUGCON}^{w/o}_{\textbf{CST2}}$ removes the use of LLM to split in the CST process by directly splitting the contexts in the middle (we will set the whole sentence in the middle all belongs to the first sub-context to maintain semantic integrity). (3) **AugCon**$^{w/o}_{\textbf{filter}}$ eliminates the contrastive-learning-based score training and filtering process and randomly samples a sufficient number of queries. (4) **AugCon**$^{w/o}_{\textbf{fidelity}}$ obtains the answers to the queries without adhering to self-alignment and self-improving but utilizes fixed few-shot examples along with a straightforward prompt design devoid of guiding principles.

We implement the four variants on TriviaQA (short-form) and WebGLM-QA (long-form) datasets and conduct a comparison with our AugCon. The results are shown in Table 2.

We find that all variants yield suboptimal outcomes, underscoring the fact that the three essential steps are all crucial and collectively contribute to achieving superior performance. A more detailed results analysis and explanation of the four variants can be found in the appendix.

Variant	TriviaQA (EM)	WebGLM-QA (BS)
AugCon$^{w/o}_{\text{CST1}}$	$0.793_{\pm 0.003}$	$0.912_{\pm 0.001}$
AugCon$^{w/o}_{\text{CST2}}$	$0.826_{\pm 0.003}$	$0.910_{\pm 0.001}$
AugCon$^{w/o}_{\text{filter}}$	$0.828_{\pm 0.003}$	$0.915_{\pm 0.001}$
AugCon$^{w/o}_{\text{fidelity}}$	$0.833_{\pm 0.004}$	$0.907_{\pm 0.002}$
AugCon	$\mathbf{0.849}_{\pm 0.003}$	$\mathbf{0.924}_{\pm 0.002}$

Table 2: The results of ablation study.

3.2 Training Phase & Granularity Comparison

We present the loss curve training on the generated *DailyM-SFT* in Figure 2. An interesting observation is that the training loss appears to plateau within epochs from Epoch 2 onwards, yet we observe sudden drops in loss at the boundaries between two consecutive epochs. This pattern strongly signals that our training dataset is characterized by extremely low similarity and exceptionally high diversity, meaning that training on one segment of data does not have an impact on the loss associated with another segment. We also conduct a human evaluation at each checkpoint during model training, on *DailyM* test set and a widely-used general alignment benchmark AlignBench [46]. The overall satisfaction scores increase steadily in both the *DailyM* test set and AlignBench, showcasing that our methods can increase the specific conversation ability without sacrificing general performance.

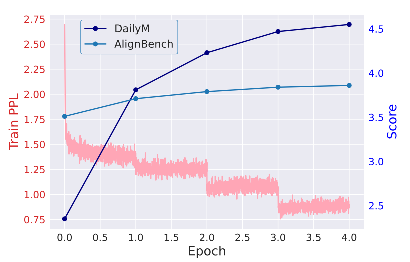


Figure 2: The training loss and human evaluation results during training phase.

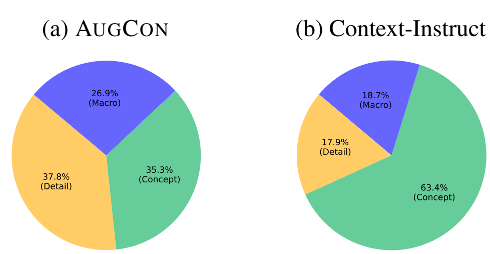


Figure 3: The proportion of three levels of granularity questions generated by AugCon and Context-Instruct.

A key innovative advantage of our approach is its ability to generate queries of varying granularity. To assess the performance of this feature in comparison to baseline methods, we categorize questions into three distinct types based on their scope and depth: detail questions, concept questions, and macro questions (with the detailed guidance for categorizing shown in the appendix), and compare our method with Context-Instruct. The proportions of the three types of questions are illustrated in Figure 3. Our approach achieves a more balanced distribution of question granularities, demonstrating its advantage in covering a diverse range of user inquiries and providing an intuitive explanation for our superior performance.

4 Conclusion

In this work, we propose AUGCON, a highly innovative and effective method to build vertical-domain AI assistants from custom corpora by deriving SFT query-response pairs with diverse granularity. AUGCON starts with query generation through the Context-Split-Tree (CST). We then employ contrastive learning to develop a scorer to rank and refine queries. Finally, we introduce a synergistic integration of self-alignment and self-improving to obtain high-fidelity responses. The automatic evaluation on four benchmarks demonstrate advantages of our method in producing high diversity, quality, and fidelity context-driven SFT data and improving the performance of fine-tuned models.

References

[1] Nazmiye Ceren Abay, Yan Zhou, Murat Kantarcioglu, Bhavani Thuraisingham, and Latanya Sweeney. Privacy preserving synthetic data release using deep learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, pages 510–526. Springer, 2019.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Angus Addlesee, Weronika Sieińska, Nancie Gunson, Daniel Hernández Garcia, Christian Dondrup, and Oliver Lemon. Multi-party goal tracking with llms: Comparing pre-training, fine-tuning, and prompt engineering. *arXiv preprint arXiv:2308.15231*, 2023.

[4] AI@Meta. Llama 3 model card. 2024.

[5] Chenxin An, Fei Huang, Jun Zhang, Shansan Gong, Xipeng Qiu, Chang Zhou, and Lingpeng Kong. Training-free long-context scaling of large language models. *arXiv preprint arXiv:2402.17463*, 2024.

[6] Rohit Babbar and Bernhard Schölkopf. Data scarcity, robustness and extreme multi-label classification. *Machine Learning*, 108(8):1329–1351, 2019.

[7] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[8] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[9] Markus Bayer, Philipp Kuehn, Ramin Shanehsaz, and Christian Reuter. Cysecbert: A domain-adapted language model for the cybersecurity domain. *ACM Transactions on Privacy and Security*, 27(2):1–20, 2024.

[10] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

[11] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR, 2022.

[12] Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *arXiv preprint arXiv:2308.08469*, 2023.

[13] Wei Chen, Qiushi Wang, Zefei Long, Xianyin Zhang, Zhongtian Lu, Bingxuan Li, Siyuan Wang, Jiarong Xu, Xiang Bai, Xuanjing Huang, et al. Disc-finllm: A chinese financial large language model based on multiple experts fine-tuning. *arXiv preprint arXiv:2310.15205*, 2023.

[14] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

[15] Daixuan Cheng, Shaohan Huang, and Furu Wei. Adapting large language models via reading comprehension. *arXiv preprint arXiv:2309.09530*, 2023.

[16] Cheng Deng, Tianhang Zhang, Zhongmou He, Qiyuan Chen, Yuanyuan Shi, Yi Xu, Luoyi Fu, Weinan Zhang, Xinbing Wang, Chenghu Zhou, et al. K2: A foundation language model for geoscience knowledge understanding and utilization. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 161–170, 2024.

[17] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

[18] Kevin Du, Vésteinn Snæbjarnarson, Niklas Stoehr, Jennifer C White, Aaron Schein, and Ryan Cotterell. Context versus prior knowledge in language models. *arXiv preprint arXiv:2404.04633*, 2024.

[19] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference on NAACL*, 2019.

[20] Zane Durante, Bidipta Sarkar, Ran Gong, Rohan Taori, Yusuke Noda, Paul Tang, Ehsan Adeli, Shrinidhi Kowshika Lakshmikanth, Kevin Schulman, Arnold Milstein, et al. An interactive agent foundation model. *arXiv preprint arXiv:2402.05929*, 2024.

[21] Nathan C Frey, Ryan Soklaski, Simon Axelrod, Siddharth Samsi, Rafael Gomez-Bombarelli, Connor W Coley, and Vijay Gadepally. Neural scaling of deep chemical models. *Nature Machine Intelligence*, 5(11):1297–1305, 2023.

[22] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023.

[23] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

[24] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

[25] Tianyu Han, Lisa C Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K Bressem. Medalpaca–an open-source collection of medical conversational ai models and training data. *arXiv preprint arXiv:2304.08247*, 2023.

[26] Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. Structured prompting: Scaling in-context learning to 1,000 examples. *arXiv preprint arXiv:2212.06713*, 2022.

[27] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023.

[28] Ting Jiang, Shaohan Huang, Shengyue Luo, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, et al. Improving domain adaptation through extended-text reading comprehension. *arXiv preprint arXiv:2401.07284*, 2024.

[29] Erik Jones, Hamid Palangi, Clarisse Simões, Varun Chandrasekaran, Subhabrata Mukherjee, Arindam Mitra, Ahmed Awadallah, and Ece Kamar. Teaching language models to hallucinate less with synthetic tasks. *arXiv preprint arXiv:2310.06827*, 2023.

[30] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[31] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.

[32] Sungdong Kim, Sanghwan Bae, Jamin Shin, Soyoung Kang, Donghyun Kwak, Kang Min Yoo, and Minjoon Seo. Aligning large language models through synthetic feedback. *arXiv preprint arXiv:2305.13735*, 2023.

[33] Jan-David Krieger, Timo Spinde, Terry Ruas, Juhi Kulshrestha, and Bela Gipp. A domain-adaptive pre-training approach for language bias detection in news. In *Proceedings of the 22nd ACM/IEEE joint conference on digital libraries*, pages 1–7, 2022.

[34] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626, 2023.

[35] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, et al. Datasets: A community library for natural language processing. *arXiv preprint arXiv:2109.02846*, 2021.

[36] Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. Common 7b language models already possess strong math capabilities. *arXiv preprint arXiv:2403.04706*, 2024.

[37] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36, 2024.

[38] Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023.

[39] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

[40] Haoxiong Liu and Andrew Chi-Chih Yao. Augmenting math word problems via iterative question composing. *arXiv preprint arXiv:2401.09003*, 2024.

[41] Mingjie Liu, Teodor-Dumitru Ene, Robert Kirby, Chris Cheng, Nathaniel Pinckney, Rongjian Liang, Jonah Alben, Himyanshu Anand, Sanmitra Banerjee, Ismet Bayraktaroglu, et al. Chipnemo: Domain-adapted llms for chip design. *arXiv preprint arXiv:2311.00176*, 2023.

[42] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

[43] Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, et al. Best practices and lessons learned on synthetic data for language models. *arXiv preprint arXiv:2404.07503*, 2024.

[44] Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*, 2023.

[45] Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. Webglm: Towards an efficient web-enhanced question answering system with human preferences. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4549–4560, 2023.

[46] Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Zhuoer Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, et al. Alignbench: Benchmarking chinese alignment of large language models. *arXiv preprint arXiv:2311.18743*, 2023.

[47] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[48] Zhengliang Liu, Aoxiao Zhong, Yiwei Li, Longtao Yang, Chao Ju, Zihao Wu, Chong Ma, Peng Shu, Cheng Chen, Sekeun Kim, et al. Tailoring large language models to radiology: A preliminary approach to llm adaptation for a highly specialized domain. In *International Workshop on Machine Learning in Medical Imaging*, pages 464–473. Springer, 2023.

[49] Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Mohammad Shoeybi, and Bryan Catanzaro. Chatqa: Building gpt-4 level conversational qa models. *arXiv preprint arXiv:2401.10225*, 2024.

[50] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[51] Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, and Chang Zhou. # instag: Instruction tagging for diversity and complexity analysis. *arXiv preprint arXiv:2308.07074*, 2023.

[52] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

[53] Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. *arXiv preprint arXiv:2305.16938*, 2023.

[54] Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[55] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[56] Soumen Pal, Manojit Bhattacharya, Sang-Soo Lee, and Chiranjib Chakraborty. A domain-specific next-generation large language model (llm) or chatgpt is required for biomedical engineering and research. *Annals of Biomedical Engineering*, 52(3):451–454, 2024.

[57] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*, 2023.

[58] Shanghaoran Quan. Dmoerm: Recipes of mixture-of-experts for effective reward modeling. *arXiv preprint arXiv:2403.01197*, 2024.

[59] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020.

[60] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[61] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023.

[62] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506, 2020.

[63] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. *arXiv preprint arXiv:2401.18059*, 2024.

[64] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2024.

[65] Omar Shaikh, Valentino Emil Chai, Michele Gelfand, Diyi Yang, and Michael S Bernstein. Rehearsal: Simulating conflict to teach conflict resolution. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2024.

[66] Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. Character-llm: A trainable agent for role-playing. *arXiv preprint arXiv:2310.10158*, 2023.

[67] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[68] Zhiqing Sun, Yikang Shen, Hongxin Zhang, Qinhong Zhou, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Salmon: Self-alignment with principle-following reward models. *arXiv preprint arXiv:2310.05910*, 2023.

[69] Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36, 2024.

[70] Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Fine-tuning language models for factuality. *arXiv preprint arXiv:2311.08401*, 2023.

[71] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[72] Jiahao Wang, Bolin Zhang, Qianlong Du, Jiajun Zhang, and Dianhui Chu. A survey on data selection for llm instruction tuning. *arXiv preprint arXiv:2402.05123*, 2024.

[73] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[74] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

[75] Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Man Zhang, et al. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. *arXiv preprint arXiv:2310.00746*, 2023.

[76] Zige Wang, Wanjun Zhong, Yufei Wang, Qi Zhu, Fei Mi, Baojun Wang, Lifeng Shang, Xin Jiang, and Qun Liu. Data management for large language models: A survey. *arXiv preprint arXiv:2312.01700*, 2023.

[77] Jerry Wei, Le Hou, Andrew Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, et al. Symbol tuning improves in-context learning in language models. *arXiv preprint arXiv:2305.08298*, 2023.

[78] Jerry Wei, Da Huang, Yifeng Lu, Denny Zhou, and Quoc V Le. Simple synthetic data reduces sycophancy in large language models. *arXiv preprint arXiv:2308.03958*, 2023.

[79] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[80] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.

[81] Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

[82] Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. Effective long-context scaling of foundation models. *arXiv preprint arXiv:2309.16039*, 2023.

[83] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.

[84] Zhaojian Yu, Xin Zhang, Ning Shang, Yangyu Huang, Can Xu, Yishujie Zhao, Wenxiang Hu, and Qiufeng Yin. Wavecoder: Widespread and versatile enhanced instruction tuning with refined data generation. *arXiv preprint arXiv:2312.14187*, 2023.

[85] Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

[86] Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Wei Lin, et al. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325*, 2023.

[87] Qi Zhang, Yiming Zhang, Haobo Wang, and Junbo Zhao. Recost: External knowledge guided data-efficient instruction tuning. *arXiv preprint arXiv:2402.17355*, 2024.

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

[89] Xuan Zhang, Navid Rajabi, Kevin Duh, and Philipp Koehn. Machine translation with large language models: Prompting, few-shot learning, and fine-tuning with qlora. In *Proceedings of the Eighth Conference on Machine Translation*, pages 468–481, 2023.

[90] Jinfeng Zhou, Zhuang Chen, Dazhen Wan, Bosi Wen, Yi Song, Jifan Yu, Yongkang Huang, Libiao Peng, Jiaming Yang, Xiyao Xiao, et al. Characterglm: Customizing chinese conversational ai characters with large language models. *arXiv preprint arXiv:2311.16832*, 2023.

Appendix

A Related work

Synthetic Data for Language Models Due to the challenges of data scarcity [6], privacy concerns [1], and the sheer cost of data collection and annotation [22], synthetic data has emerged as a promising solution to build large, diverse, and high-quality datasets at scale [43]. One benefit of synthetic data is it can be tailored to specific requirements [15, 28, 45], with practical applications having been employed in various domains. WizardMath [52] leverages a series of operations to increase the complexity of questions and answers using GPT-3.5, while Reflexion [67] employs external or internally simulated linguistic feedback to improve the code reasoning capabilities of language models. Similarly, Toolformer [64] learns to decide which APIs to call and what arguments to pass by training on template-generated data. In addition, synthesized data has been proven effective in mitigating hallucination [77, 78, 29, 70] and aligning with shared human preferences and values [8, 69, 68, 58, 32]. While the generation of context-driven synthetic data has proven to be a powerful substitute for manual annotation, the challenge of ensuring high-quality synthetic data, which encompasses the complexity of queries [44, 36, 40], the diversity of semantics [17, 76, 72, 51, 49], and the scale of the synthetic datasets [85, 23, 38], has been a consistent pursuit.

Context-Driven Synthetic Data Numerous studies have developed techniques for creating synthetic data informed by contextual cues. UltraChat [17] leverages user-specified topics and supplements these with existing textual material to craft instructional conversations aimed at enhancing chatbot performance. SPIN [14], on the other hand, autonomously generates training data from its previous iterations, employing this approach to progressively refine its capabilities. Additionally, various methods have been devised to extract character profiles and personas from collected books or scripts for the purpose of producing roleplaying dialogues [66, 90, 75, 37], and several initiatives focus on mining domain-specific data from specialized corpora to construct domain-specific language models [15, 28, 21, 16, 86]. While alternative approaches employ retrieval augmented generation (RAG) [61, 11] or integrate auxiliary knowledge in vast context windows [82, 5], issues like entity susceptibility [18], high inference computational demand [39, 26], and alignment difficulties with formats and preferences [57, 53, 3] highlight the crucial role of context-driven SFT in effectively incorporating corpus knowledge internally.

B Method Details

B.1 Recursively Deriving Queries via Context-Split-Tree

Algorithm 1 Context Split Tree

Input: A corpus $\mathcal{C}$, CST prompt instruction I_{CST}, CST few-shot examples E_{CST}
Output: Query dataset $Data$ comprises of split context and derived query pairs

function ContextSplitTree($C, Data$)
  if $len(C) < \lambda$ **then**
    return ▷ Below the minimum granularity
  end if
  Call LLM to get $C_1, C_2, q \leftarrow$ LLM($I_{\text{CST}}, E_{\text{CST}}, C$)
  Append (C, q) to $Data$
  if $len(C_1) \geq len(C)$ **or** $len(C_2) \geq len(C)$ **or** ROUGE-L[P] < 0.7 **then**
    return ▷ The signs of hallucinations
  end if
  ContextSplitTree($C_1, Data$) ▷ Recursive calling
  ContextSplitTree($C_2, Data$)
end function

Initialize $Data \leftarrow$ empty list
for each extracted context $C \in \mathcal{C}$ **do** ▷ Extraction method detailed in Appx B.1
  ContextSplitTree($C, Data$)
end for
return $Data$

The Context-Split-Tree construction process prepares by dividing the corpus into short, consecutive text contexts, each with a limit of 500 words. Gain inspiration from retrieval augmentation methods [63], if a sentence surpasses the 500-word threshold, we move the entire sentence to the next context, rather than cutting it mid-sentence. This approach maintains the contextual integrity and semantic consistency of the text in each context. After obtaining the extracted contexts, for each context, we construct a correspondence CST in the manner described previously in Section 2.1. The specific templates and few-shot examples used are detailed in Appendix B.1.2.

B.1.1 Proof of Linear Relationship

A commendable property is that with the initial text length l provided, we can achieve multi-granularity effects through a linear quantity of generated questions. To be more precise, the number of questions generated will have a linear relationship with the number of minimum sentence units contained in the initial context. This allows us to generate different distributions of query granularity by simply adjusting the minimum sentence length λ and the initial context length without worrying about significant fluctuations in the computation consumption or the overall number of queries obtained. We will prove this property in the following.

We represent context as a collection of sentences $C = \{S_1, S_2, \dots, S_n\}$. We assume that during the CST process, these sentences are the smallest units and will not be split internally or increased in quantity (they may be polished but do not change the essential semantics). Formally, we have the following assumption.

Assumption: Given a context $C = \{S_1, S_2, \dots, S_n\}(n > 1)$, using LLM for a split will definitely yield one question q and two child context C_1 and C_2 that satisfy that there exists an integer $1 \leq i < n$ such that $C_1 = \{S_1, \dots, S_i\}$ and $C_2 = \{S_{i+1}, \dots, S_n\}$. Specifically, when the context degrades to a single sentence, calling the LLM will generate a question and terminate.

Then, based on the preceding assumptions, we have the following proposition.

Proposition: For any context $C = \{S_1, S_2, \dots, S_n\}$ containing n sentences, where n is an arbitrary positive integer, applying CST to it will ultimately generate $2n - 1$ questions.

Proof: We will prove the proposition using the *Second Principle of Mathematical Induction*.

1. First, for $n = 1$, calling the LLM will generate a question and then terminate, which is consistent with the proposition.
2. Secondly, assume the conclusion holds for all $n \leq k$ ($k \geq 1$). When $n = k + 1$, according to the assumption, calling the LLM with $C = \{S_1, \dots, S_{k+1}\}$ will generate a question q, along with two sub-contexts C_1 and C_2, where there exists $1 \leq i < k + 1$ such that $C_1 = \{S_1, \dots, S_i\}$, and $C_2 = \{S_{i+1}, \dots, S_{k+1}\}$. The numbers of sentences in C_1 and C_2 are i and $k + 1 - i$ and both strictly less than $k + 1$. By assumption, C_1 will eventually generate $2i - 1$ questions, and C_2 will generate $2(k + 1 - i) - 1 = 2(k - i) + 1$ questions. Therefore, in total, C will generate $1 + (2i - 1) + (2(k - i) + 1) = 2k + 1 = 2n - 1$ questions by the end. Thus, the proposition also holds for $n = k + 1$.
3. Finally, combining 1 and 2 along with the *Second Principle of Mathematical Induction*, it can be concluded that the proposition holds.

Therefore, a context containing n sentences will ultimately generate $2n - 1$ questions, which establishes a linear relationship between the number of sentences in the context and the number of questions generated. □

B.1.2 Prompt Template & Few-Shot Examples

Our method is applicable across various languages, and we provide the utilized prompt templates in English here.

Prompt Template for Context-Split-Tree with Instruction in English

Given an entire context as the Context, generate a Question about the entire context that users might be interested in, which answer should be able to be derived directly from the Context. Then, divide the entire context into two sub-contexts Context 1 and Context 2 based on their semantic content, making necessary adjustments within each sub-contexts to ensure they are independently coherent.

—

Provide in the following form:
Context: The entire context
Question: Regarding the entire context
Context 1: Sub-context 1
Context 2: Sub-context 2

—

Context: Trying to rebound from their divisional home loss to the Buccaneers, the Panthers flew to the Louisiana Superdome for a Week 5 divisional duel with the winless New Orleans Saints. With QB Jake Delhomme out and done for the year with a right elbow injury, QB David Carr was given the start. In the first quarter, Carolina took the early lead with kicker John Kasay getting a 23-yard field goal. The Saints responded with kicker Olindo Mare getting a 25-yard field goal. In the second quarter, the Panthers went back into the lead with Kasay nailing a 35-yard field goal. New Orleans would respond with Mare kicking a 28-yard field goal. In the third quarter, Carolina trailed as Saints FB Mike Karney got a 2-yard TD run for the only score of the period. In the fourth quarter, the Panthers tied the game with Carr completing a 17-yard TD pass to WR Steve Smith. Afterwards, Carolina sealed the win in the final seconds with Kasay nailing a 52-yard field goal as time ran out.
Question: How did the Carolina Panthers secure their victory against the New Orleans Saints in their Week 5 divisional duel?
Context 1: Trying to rebound from their divisional home loss to the Buccaneers, the Panthers flew to the Louisiana Superdome for a Week 5 divisional duel with the winless New Orleans Saints. With QB Jake Delhomme out for the year with a right elbow injury, QB David Carr was given the start. In the first quarter, Carolina took the early lead with kicker John Kasay getting a 23-yard field goal. The Saints responded with kicker Olindo Mare getting a 25-yard field goal. In the second quarter, the Panthers went back into the lead with Kasay nailing a 35-yard field goal, followed by New Orleans' response with Mare kicking a 28-yard field goal.
Context 2: As the game progressed into the third quarter, the Panthers found themselves trailing after Saints FB Mike Karney got a 2-yard TD run, marking the only score of the period. However, in the fourth quarter, the Panthers managed to tie the game thanks to QB David Carr completing a 17-yard TD pass to WR Steve Smith. The climax of the match came in the final seconds with John Kasay nailing a 52-yard field goal as time ran out, securing a dramatic victory for Carolina against the New Orleans Saints.

—

Context: As a cell grows, its volume increases more quickly than its surface area. If a cell was to get very large, the small surface area would not allow enough nutrients to enter the cell quickly enough for the cell's needs. However, large cells have a way of dealing with some size challenges. Big cells, such as some white blood cells, often grow more nuclei so that they can supply enough proteins and RNA for the cell's requirements. Large, metabolically active cells often have lots of cell protrusions, resulting in many folds throughout the membrane. These folds increase the surface area available for transport of materials into or out of the cell. Such cell types are found lining your small intestine, where they absorb nutrients from your food through protrusions called microvilli.
Question: How do large cells adapt to the challenge of having a volume that increases more quickly than their surface area to meet their metabolic needs?
Context 1: As a cell grows, its volume increases more quickly than its surface area. If a cell was to get very large, the small surface area would not allow enough nutrients to enter the cell quickly enough for the cell's needs.
Context 2: Large cells have a way of dealing with their size challenges. Big cells, such as some white blood cells, often grow more nuclei so that they can supply enough proteins

and RNA for the cell's requirements. Large, metabolically active cells often have lots of cell protrusions, resulting in many folds throughout the membrane. These folds increase the surface area available for transport of materials into or out of the cell. Such cell types are found lining your small intestine, where they absorb nutrients from your food through protrusions called microvilli.

—

Context: Philip Arnold Heseltine is best known as a composer of songs and other vocal music; he also achieved notoriety in his lifetime through his unconventional and often scandalous lifestyle.
Question: Why is Philip Arnold Heseltine's reputation mixed?
Context 1: Philip Arnold Heseltine is best known as a composer of songs and other vocal music.
Context 2: Philip Arnold Heseltine also achieved notoriety in his lifetime through his unconventional and often scandalous lifestyle.

—

Context: {Context}
Question:

In the prompt template, the {Context} in blue color will be replaced with the given context to derive query and split during usage. After the LLM provides a response, we employ regular expressions to parse out the fields for Question, Context 1, and Context 2. Should the parsing fail, we will attempt the process up to three more times; otherwise, the context will be discarded. However, in our experiment, we find that almost all contexts can be successfully parsed in the first response. To observe this template in a more specific way, we have provided a detailed case demonstration in Appendix D.

Note that the three few-shot examples in the prompt templates are not entirely fixed but can be adapted to different corpora by selecting suitable examples to stimulate the model to generate a question distribution that is close to the distribution of real user questions about that domain corpus, which can improve the quality of the synthetic data and the effectiveness of the final fine-tuned model.

B.2 Training Scorer to Rank Queries and Filtering

B.2.1 Training Scorer via Contrastive Learning

In our approach to enhancing data quality and diversity, we focus on the innovative construction of positive and negative samples for training our scorer. By employing contrastive learning, we train the scorer that efficiently evaluates the degree of adherence to instruct prompts and few-shot examples, surpassing previous methods that rely on heuristic algorithms or direct scoring, which are prone to positional bias and instability.

Positive samples are straightforwardly generated by employing the LLM to create context-query pairs that adhere to well-designed instruct prompts and few-shot examples shown in Appendix B.1.2, which are also the queries generated through Step 1. These serve as exemplars reflecting the desired output. On the other hand, the creation of negative samples involves intentional manipulation of instruct prompts or few-shot examples, or both. More specifically, we manipulate the instruction by simplifying the instruct prompt to *"Given a context, generate a question and split context into two sub-contexts"*. To manipulate few-shot examples, we degrading it to one-shot by retaining only one example. We also combine both approaches to simultaneously manipulate the instruction and few-shot examples to generate more negative samples. These manipulations aim to deviate from the optimal query generation, thus producing examples that diverge from the model's training objective. For each type, we randomly select 500 positive samples generated from Step 1 to construct negative samples, forming $1,500$ positive-negative sample pairs for scorer training in total.

We construct the scorer's structure as outlined in Section 2.2, initializing it with parameters from our base LLM to serve as the training warm-up. The scorer's parameter set is a duplicated version, ensuring that modifications do not impact the original base LLM. For efficient training, we employ QLoRA with 4-bit quantization and the ranks of 32, significantly reducing GPU memory requirements and speeding up the training process. A more detailed breakdown of the hyperparameter configuration is provided in Table 6.

B.2.2 Collaborating Scorer with CST to Filter Queries

Firstly, we show the pseudocode of filtering process in Algorithm 2.

Algorithm 2 Scorer collaborates with CST to filter queries

Input: A Context C, Required number of maintained queries N

Output: Query dataset $Data$ comprises exactly N queries with high quality and diversity

```
function FILTER(Q_All)
    Initialize Q_Cand ← empty list
    Sort Q_All by score descending
    for each (q, s) ∈ Q_All do
        if All ROUGE-L[F1] with Q_Cand < 0.7 then
            Append q to Q_Cand                          ▷ Append if diversity reach the threshold
            if len(Q_Cand) = N then return Q_Cand       ▷ Enough quantity
            end if
        end if
    end for
    return Q_Cand
end function

Initialize Data ← empty list
Initialize Q_All ← empty list                           ▷ Store all query-score pairs
while len(Data) < N do                                  ▷ Iterate until enough queries obtained
    Initialize Q_New ← empty list
    CONTEXTSPLITTREE(C, Q_New)                          ▷ Call CST to get new queries
    for each q ∈ Q_New do
        Apeend (q, Sc(C, q)) to Q_All                   ▷ Score each new query
    end for
    Set Data ← FILTER(Q_All)
end while
return Data
```

The scorer is designed to work in cooperation with CST to enhance the quality and diversity of the generated questions while also meeting the quantitative requirements. For a given context, we first use CST to generate a series of potential questions. Then, each question is scored by the scorer, with the scores used to rank them from highest to lowest. We sequentially add questions to a candidate set, but only if the current question's ROUGE-L F1 similarity to any question already in the set is less than 0.7. This process continues until the number of questions in the candidate set reaches the desired quantity N.

If the initial round of CST and scoring does not yield the required number of questions, we initiate another round of CST to expand on the initial questions, followed by repeating the scoring and selection process until the target quantity is achieved. The scorer's role in scoring and filtering is like to effectively condense the output from CST, ensuring that even with a smaller set of questions, both high quality and diversity are maintained.

B.3 Obtaining High-Fidelity Responses

Our design is motivated by the significant influence principles have on guiding LLMs, aiming to achieve high-fidelity responses through a principle-driven self-alignment step. These principles are anticipated to enhance the LLM's ability to produce high-fidelity, realistic, and helpful answers, and the specific principles vary depending on the task and remain exploratory. They may also include rules for directing the LLM to generate responses in a particular tone. This could be particularly valuable when creating SFT data for a custom LLM assistant or for role-playing. Furthermore, the existence of principles serves as a method for aligning with human preferences, offering a viable alternative to the cumbersome process of reinforcement learning from human feedback (RLHF) [55].

Different from previous approaches, we innovate to integrate a self-improving pipeline to further increase fidelity. Instead of manually selecting a few-shot examples from annotated examples, we divide the annotated examples into training and testing sets. We then conduct a random search that iteratively selects a subset from the training set and allows the LLM to self-evaluate the output scores in the test set. This process is iterated 16 times by default, and the subset that achieves the highest scores in the test set is used as the few-shot ICL examples. We implement this pipeline through the DSPy [31] framework, significantly reducing coding effort. This self-improving process works well with principle-driven self-alignment, as it aids in identifying the optimal ICL examples that guide the LLM to generate helpful, realistic, and reliable answers in line with alignment principles, markedly enhancing the quality and fidelity of the responses and, consequently, the responses by the fine-tuned models.

Finally, we prune all contexts, principles, and ICL examples to retain only the query-response pairs for supervised fine-tuning of the LLM. While several studies [87, 84, 81] try to further execute filtering on generated answers, we leave it as a future work as it is not such crucial for our method. Actually, simply generating additional iterations on the same query and retaining self-consistent [73] responses may further improve some degrees of reasoning accuracy for short-form responses. However, this might not be a good deal when also taking the computing costs into consideration since letting LLMs improve and correct their own responses is not an easy thing [27]. We believe that since each question we obtain in CST precisely matches the granularity of its context, it will be easy for LLM to provide accurate and pertinent answers to the questions.

C Implementation Details

All experiments are implemented on a single node with eight Nvidia A100 80G GPUs and 160 Intel Xeon Gold 6248 CPUs. To speed up the generation of LLM calls, we use the vLLM [34] inference engine for acceleration, and make concurrent requests with a concurrency of 8 threads. In order to reduce GPU memory usage and accelerate training speed, we use the DeepSpeed [62] distributed training framework accelerating with ZeRO-2 [59], where the AdamW [50] optimizer is applied for gradient descent. We employ QLoRA with 4-bit quantization and the ranks of 32, which has been demonstrated to be able to achieve satisfactory results in previous works [28, 89]. We train for 4 epochs in total, for it has been shown as the maximum number of iterations that negligible affect the training loss [54]. On the *DailyM* dataset, our AugCon generates about 120K pieces of data in 184 A100 GPU hours and required another 272 A100 GPU hours for supervised fine-tuning. On four benchmarks, the generation throughput is about 340 pairs per A100 GPU hour and the total running times vary from 120 to 448 A100 GPU hours depending on corpus size. More detailed settings on hyperparameters can be found in Appendix C.3.

C.1 Baselines

(1) **Chat Model** [7, 71] applies instruction tuning and alignment tuning after pre-training. We utilize it both as the basic baseline and as the base model for calling and fine-tuning across all other baselines and our methods for fair comparison.

(2) **DAPT** [24] continuously pre-trains directly on the raw custom corpus to adapt and grasp domain-specific knowledge.

(3) **AdaptLLM** [15] builds SFT samples by converting the raw corpora into reading comprehension tasks via regex-based mining patterns. Tasks they design include summarization, word-to-text, natural language inference, commonsense reasoning, and paragraph detection.

(4) **ETRC** [28] derives question-answer pairs from extracted contexts with an LLM and augments data by ensembling contexts and their corresponding question-answer pairs with a length-based clustering algorithm. their corresponding question-answer pairs with a length-based clustering algorithm.

(5) **Context-Instruct** [75] is a context-driven instruction generation method that contains three parts: 1) partition text into manageable segments, 2) use an LLM to generate question, response, and confidence score triplets based on the segments, and 3) apply confidence-score-based filtering and deduplication to ensure data quality and diversity.

C.2 Assets Use

All the pre-trained LLM and open-source datasets we use in experiments can be respectively found on Huggingface transformers [79] and datasets [35], and we have checked that they are available for research purposes and have been properly cited and correctly adhered to open-source licenses. We list the public links of the used LLMs in Table 3 and the used datasets in Table 4. We will also make our *DailyM* dataset open-sourced at `https://anonymous.4open.science/r/AugCon` to boost the academy.

Table 3: Public links to the used LLMs.

LLM	Link
Qwen1.5-c_{32B} [7]	`https://huggingface.co/Qwen/Qwen1.5-32B-Chat`
Llama3-c_{70B} [4]	`https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct`

In our experiments, we conducted an automatic evaluation on four widely-used benchmarks, with the detailed description of these benchmarks listed below:

(1) **SQuAD1.1** [60] is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage. SQuAD1.1 contains $100,000+$ question-answer pairs on $500+$ articles.

(2) **TriviaQA** [30] includes 95K question-answer pairs authored by trivia enthusiasts and independently gathered evidence documents, six per question on average, that provide high quality distant supervision for answering the questions.

(3) **DROP** [19] is a crowdsourced, adversarially-created, 96K-question benchmark, in which a system must resolve references in a question, perhaps to multiple input positions, and perform discrete operations over them.

(4) **WebGLM-QA** [45] is the data used to train the WebGLM generator module and an LLM bootstrapped quoted and long-formed QA dataset via in-context learning and corresponding strategies to clean and refine, with 45K high-quality filtered and 83K unfiltered samples.

Table 4: Public links to the used datasets.

Dataset	Link
SQuAD1.1 [60]	`https://huggingface.co/datasets/rajpurkar/squad`
TriviaQA [30]	`https://huggingface.co/datasets/mandarjoshi/trivia_qa`
DROP [19]	`https://huggingface.co/datasets/ucinlp/drop`
WebGLM-QA [45]	`https://huggingface.co/datasets/THUDM/webglm-qa`

C.3 Hyperparameters

We present the generation hyperparameters in Table 5 and the fine-tuning configurations in Table 6.

D Case Demonstration

In this section, we select a specific context as a case demonstration to inspect the entire process around it, from generating questions by constructing Context-Split-Tree, to scoring queries and filtering them, and finally to generating answers by attaching self-alignment principles and optimal few-shot examples. This demonstration is designed to provide a clearer and more intuitive understanding of our algorithm's workflow.

Initially, we input the starting context as the root node of the whole Context-Split-Tree. With the assistance of the LLM, we derive an overall question from this context and simultaneously split it into two child contexts. We have omitted the specific instruction in the text boxes below, as it has already

Table 5: Generation Hyperparameters

Parameter	Value
Max instruction length	4096
Max new tokens	4096
Top-k	50
Top-p	1.0
Temperature (for query)	0.85
Temperature (for response)	0.2

Table 6: Training Configurations

Parameter	Value
Epoch	4
Learning rate	5e-5
Mini batch size	4
Warmup steps	50
Weight decay	0.01
Compute dtype	bfloat16
Quantization dtype	nf4
Lora rank	32
Lora alpha	32
Lora dropout	0.05
Lora bias	none

been presented in detail in Appendix B.1.2. The Context in boxes is the input while the Question, Context 1, and Context 2 are parsed outputs:

Node 1 $\rightarrow$ Node 2, Node 3

Context: The profits of the contemporary global value chains (GVC) form a V-shape, also known as the "smile curve". At one end of this curve are research and development (R&D) and design, and at the other end are services and marketing, with processing and production situated in the middle. Typically, the profit margin for industries at both ends ranges from 20% to 25%, whereas the profit margin for the production processes in the middle stands at merely 5%.
Question: Why do entrepreneurs worldwide strive to move up the value chain?
Context 1: The profits of the contemporary global value chains (GVC) form a V-shape, also known as the "smile curve". At one end of this curve are research and development (R&D) and design, and at the other end are services and marketing, with processing and production situated in the middle.
Context 2: Typically, the profit margin for industries at both ends of the global value chains ranges from 20% to 25%, whereas the profit margin for the production processes in the middle stands at merely 5%.

The child context undergoes recursive splitting and derives questions with corresponding granularity:

Node 2 $\rightarrow$ Node 4, Node 5

Context: The profits of the contemporary global value chains (GVC) form a V-shape, also known as the "smile curve". At one end of this curve are research and development (R&D) and design, and at the other end are services and marketing, with processing and production situated in the middle.
Question: What are the key components of the contemporary global value chains?
Context 1: The profits of the contemporary global value chains (GVC) form a V-shape, also known as the "smile curve".
Context 2: At one end of the smile curve are research and development (R&D) and design, and at the other end are services and marketing, with processing and production situated in the middle.

We use depth-first-search (DFS) so this is the time for Node 4. Actually, using breadth-first-search (BFS) will ultimately achieve the same effect. Node 4 derives a question but terminates afterward due to its unsuccessful attempt to split into two viable child contexts:

Node 4 Ends

Context: The profits of the contemporary global value chains (GVC) form a V-shape, also known as the "smile curve".
Question: What does the global value curve look like?
Context 1: The profits of the contemporary global value chains (GVC) form a V-shape, also known as the "smile curve".
Context 2:

The child Context 1 split from Node 5 will not further derive questions or split because it falls below the minimum length threshold. However, one can adjust this threshold to a lower value to derive more detailed questions:

Node 5 $\rightarrow$ Node 6

Context: At one end of the smile curve are research and development (R&D) and design, and at the other end are services and marketing, with processing and production situated in the middle.
Question: What is the structure of the smile curve?
Context 1: At one end of the smile curve are research and development (R&D) and design.
Context 2: The other end of the smile curve are services and marketing, with processing and production situated in the middle.

Node 6 derives a question and terminates because both of its child contexts are too short:

Node 6 Ends

Context: The other end of the smile curve are services and marketing, with processing and production situated in the middle.
Question: What lies in the middle of the smile curve?
Context 1: The other end of the smile curve are services and marketing.
Context 2: The processing and production are situated in the middle.

After Node 2 has completed its recursion, it is now Node 3's turn to proceed:

Node 3 $\rightarrow$ Node 7, Node 8

Context: Typically, the profit margin for industries at both ends of the global value chains ranges from 20% to 25%, whereas the profit margin for the production processes in the middle stands at merely 5%.
Question: Which type of industry has the lowest profit margin?
Context 1: Typically, the profit margin for industries at both ends of the global value chains ranges from 20% to 25%.
Context 2: Whereas the profit margin for the production processes in the middle stands at merely 5%.

Node 7 and Node 8 terminate after deriving one detailed question each, as they have reached the minimum granularity and cannot split properly:

Node 7 Ends

Context: Typically, the profit margin for industries at both ends of the global value chains ranges from 20% to 25%.
Question: How high can the profit margin go for industries at two ends of the global value chains?
Context 1: Typically, the profit margin for industries at both ends of the global value chains ranges from 20% to 25%.
Context 2:

Node 8 Ends

Context: Whereas the profit margin for the production processes in the middle stands at merely 5%.
Question: What is the profit margin for the production processes?
Context 1: Whereas the profit margin for the production processes in the middle stands at merely 5%.
Context 2:

Then, the recursion comes to an end, and this entire process ultimately results in the formation of the Context-Split-Tree depicted in Figure 4. Each node within this tree contains a context and a question that align with the corresponding granularity.

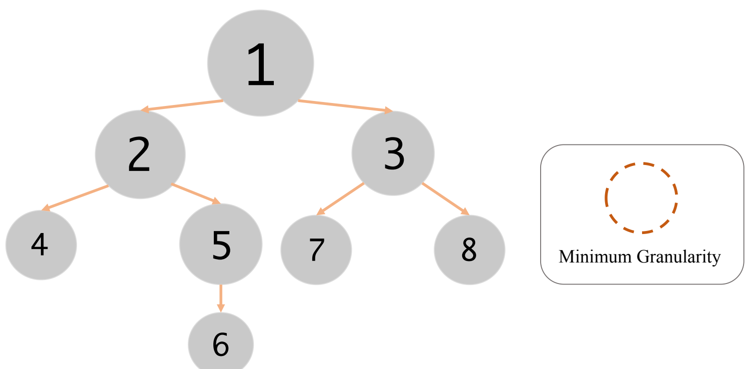


Figure 4: The schematic of the constructed CST in this case. Each node contains a context and a corresponding question, with the node size indicating different levels of granularity.

We collect context question pairs from all nodes into a list and employ the trained scorer to evaluate each item. The items are then sorted based on their scores, from highest to lowest. Next, we sequentially examine each item and choose those queries whose ROUGE-L scores with all previously selected queries are below 0.7. Due to the low ROUGE-L scores among query pairs in that case, the resulting selected set shown in Table 7 primarily comprises the preceding few items. However, if we aim to derive a greater number of questions from the context, such as 10 questions, an additional round of CST becomes necessary. Following this, all queries generated across both CST sessions undergo a collective ranking and filtering process. During this step, the ROUGE-L metric proves useful for eliminating queries that have lower scores and are similar to previously selected ones.

At this point, in addition to our results, we also provide the generated query list using the Context-Instruct method for a direct comparison. The Context-Instruct method produces question-confidence-answer triplets, where the confidence level can be either high or low and is used for filtering purposes. The results are presented in Table 8, with the responses omitted to conserve space. Putting Table 7 and Table 8 together, we can perceive that the queries generated by Context-Instruct exhibit a noticeable lack of diversity and granularity compared to our queries. This discrepancy provides an intuitive explanation for our method's superior performance in generating multi-granularity queries and ultimately producing better results than other methods.

query	score	select
Why do entrepreneurs worldwide strive to move up the value chain?	0.95	✓
What are the key components of the contemporary global value chains?	0.91	✓
How high can the profit margin go for industries at two ends of the global value chains?	0.88	✓
What does the global value curve look like?	0.83	✓
Which type of industry has the lowest profit margin?	0.74	✗
What is the structure of the smile curve?	0.67	✗
What is the profit margin for the production processes?	0.64	✗
What lies in the middle of the smile curve?	0.59	✗

Table 7: The results of ranking and filtering the queries.

query	confidence	select
What is the shape of the profits of global value chains?	High	✓
What is the "smile curve" also known as?	High	✓
What is the primary function of the industries at one end of the "smile curve"?	High	✓
Which activities have a profit margin ranging from 20% to 25%?	High	✓
What is the typical profit margin for industries in the middle of the "smile curve"?	High	✗
What is situated at both ends of the "smile curve"?	Low	✗
Do companies typically earn higher profits from production processes than from R&D?	Low	✗
Is the "smile curve" a linear representation of profits in global value chains?	Low	✗

Table 8: The generated questions using Context-Instruct.

Returning to our workflow, after filtering the queries, we leverage the alignment principles and a select few-shot examples (detailed in Appendix B.3) to assist our LLM in providing high-fidelity answers to these questions. Thanks to our Context-Split-Tree process providing the specific window of context required to answer the questions, obtaining these answers is relatively easy and results in more accurate and context-relevant responses. Once we have obtained high-fidelity answers, we remove the alignment principles, few-shot examples, and context, retaining only the query-response pairs to supervised fine-tune our LLM. The final obtained SFT data for this case are presented in the following text tables.

SFT Data 1

Question: Why do entrepreneurs worldwide strive to move up the value chain?
Answer: Entrepreneurs worldwide strive to move up the value chain because the profit margins are significantly higher at the ends of the curve, ranging from 20% to 25%, compared to the middle, which has a profit margin of only 5%. By moving up the value chain, entrepreneurs can increase their profit margins and gain a competitive advantage in the market.

SFT Data 2

Question: What are the key components of the contemporary global value chains?
Answer: The key components of the contemporary global value chains are:
1. Research and Development (R&D) and Design (at one end of the curve)
2. Processing and Production (at the other end of the curve)
3. Services and Marketing (in the middle of the curve)

SFT Data 3

Question: How high can the profit margin go for industries at two ends of the global value chains?
Answer: The profit margin for industries at both ends of the global value chains can go up to 25%.

SFT Data 4

Question: What does the global value curve look like?
Answer: It looks like a V-shape, also known as the "smile curve".

E Additional Experiments Analysis

E.1 Ablation Study

In this section, we detail the ablation experiments conducted to assess the indispensability and impact of the three essential steps in our proposed method. These steps are integral to our approach, designed to recursively derive queries, rank and filter them for quality and diversity, and finally, generate high-fidelity responses. Through these experiments, we aim to delineate the contribution of each step towards the overall effectiveness of our method. In this study, we develop the following four distinct variations of our method, with each one specifically tailored to concentrate on a fundamental step:

1. **AugCon$^{w/o}_{\textbf{CST1}}$** omits the use of the Context-Split-Tree for iteratively splitting and generating queries for given contexts. Instead, AugCon$^{w/o}_{\text{CST}}$ employs a technique where few-shot examples are used to iteratively derive queries from the extracted context until the desired number of queries is obtained (we set the desired number to be the same with all generated queries of AugCon without filtering). The purpose of this modification is to assess the efficacy of CST in deriving multi-granularity queries. Additionally, this variant facilitates an examination of how the exclusion of CST impacts the diversity of the generated queries and the overall performance of the final fine-tuned model.
2. **AugCon$^{w/o}_{\textbf{CST2}}$** also omits the use of the Context-Split-Tree for iteratively splitting and generating queries for given contexts. Different from AugCon$^{w/o}_{\text{CST1}}$, AugCon$^{w/o}_{\text{CST2}}$ splits the contexts in a heuristic way that each time splits it in the middle (we will let the whole sentence in the middle in the first sub context to maintain semantic integrity) until reaching the minimum granularity. And then use all split contexts to iteratively derive queries until the quantity is enough. This variant is designed to further assess the efficacy of CST in deriving multi-granularity queries with the comparison with a heuristic context segmentation method.
3. **AugCon$^{w/o}_{\textbf{filter}}$** eliminates the scoring and filtering process to evaluate its effects on the overall quality and diversity of the generated queries. If the number of queries generated in Step 1 exceeds the predetermined limit, we just proceed by randomly selecting a sufficient number of queries to meet the quota. This variant enables us to assess the effects of bypassing our established quality and diversity control mechanisms.
4. **AugCon$^{w/o}_{\textbf{fidelity}}$** obtains the answers to the queries without adhering to self-alignment or employing the self-improving. Instead, AugCon$^{w/o}_{\text{fidelity}}$ utilizes fixed predetermined few-shot examples along with a straightforward prompt design devoid of guiding principles. This variant allows us to evaluate the efficacy of our response generation methodology in enhancing the overall quality and relevance of responses.

We implement the four variants on TriviaQA (short-form) and WebGLM-QA (long-form) datasets and conduct a comparison with our AugCon. The results are shown in Table 2.

Our analysis has led to three key insights. Firstly, when compared to our AugCon, all variants yield suboptimal outcomes. This highlights the critical nature of each step within our methodology, underscoring the fact that they are all crucial and collectively contribute to achieving superior performance.

Secondly, within the context of short-form datasets, it was observed that the variants that undergo modifications in the CST process perform the poorest. This finding suggests that the CST process plays a vital role in encompassing a comprehensive scope of granularity, thereby enabling the extraction of a broader spectrum of knowledge.

Thirdly, with regard to long-form datasets, the variant AUGCON$^{w/o}_{\text{fidelity}}$ demonstrates the lowest level of performance. This outcome underlines the significance of self-alignment and self-enhancement mechanisms in generating responses of high quality and fidelity.

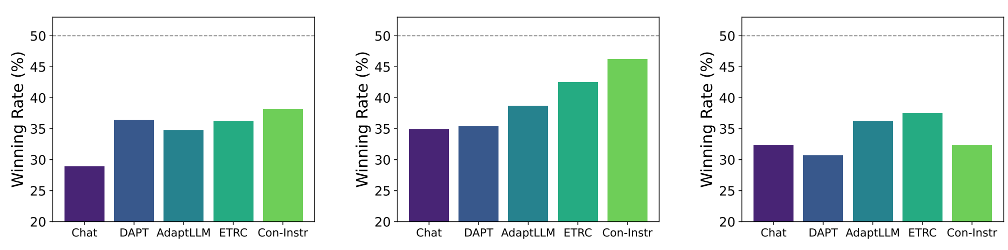
E.2 GPT-4 Judge

To conduct a more comprehensive evaluation, we utilized GPT-4 [2] (we use gpt-4-0125-preview) to judge as the referee on the baseline method and our approach. By using GPT-4 to compare the outputs of the model fine-tuned on our generated data with the baseline methods, we can better understand the advantages of our model and avoid biases stemming from manual preferences. We provide the query in the *DailyM* test set as inputs and get the outputs of the model fine-tuned using our method and baseline method on the *DailyM* dataset. To facilitate a nuanced evaluation, we categorized the queries into three levels of granularity: detail question (*e.g.*, *how deep does the abortion needle penetrate?*), concept question (*e.g.*, *what is the difference between emergencies and crises?*), and macro question (*e.g.*, *what impact does faith have on us?*). We present the detailed guidance for categorizing in Table 11.

For the reliability of the results, we extract relevant references to the questions in the dataset corpus to assist GPT-4 in making decisions. Then, we ask GPT-4 to compare the outputs generated by the two models, with the template shown in below text box. To mitigate the potential impact of position bias of LLMs, we implement a robust evaluation strategy that for each pair of outputs, we swap their positions and queried GPT-4 twice. In cases where the two responses were not consistent, we continued to inquire until we obtained a unanimous answer. The comparison results are shown in Figure 5.

GPT-4 Judge Template

Given a question and relevant reference, decide which one is better between answer 1 and answer 2.
Question: {Question}
Reference: {Reference}
Answer 1: {Answer1}
Answer 2: {Answer2}
Your decision:



(a) Detail question (b) Concept question (c) Macro question

Figure 5: The results of GPT-4 judge on three levels of questions.

The performance comparisons are depicted in Figure 5. Note that since we employ a decisive, non-tie judgment, for a method to be considered superior, it must surpass the 50% winning threshold. Through this rigorous comparison, it becomes apparent that our proposed method significantly outperforms the baseline methods across all three categories of question granularity. Notably, the winning rates for the baseline methods generally fall below 40% in most instances, underscoring the effectiveness of our approach.

Among the baseline methods evaluated, ETRC and Context-Instruct exhibit relatively better performance in the domain of concept questions, with winning rates slightly exceeding 40%. This performance is commendable when compared to other baseline methods. However, it's important to acknowledge that even these higher-performing baselines do not meet the 50% threshold and exhibit weaknesses across the other two question granularities. This observation highlights a critical advantage of our method: its ability to maintain a balanced focus across different types of questions. Consequently, our approach not only excels in one particular area but also enhances performance across all three types of queries simultaneously.

The results from this comparison clearly illustrate the superior capability of our method in handling a diverse range of question complexities and types. This balanced focus is pivotal for developing systems that can adapt to varied informational needs, thereby improving the overall query response performance. Our findings suggest that our method could significantly contribute to advancements in improving the performance of fine-tuned models by offering a more versatile and effective approach to extracting multi-granularity query-response pairs from context.

E.3 Computation Experiment

In previous experiments, since we find that all methods spend much more time on final fine-tuning compared to the previous generation, we maintained that each method produced an equivalent number of query response pairs for the given contexts to balance computing resources. Nonetheless, when examining methods like ETRC, Context-Instruct, and our AUGCON that need LLM calls, we observed variations in the computational time required to generate each query response data pair. In this evaluation, we control the LLM computation to be the same, all gave 80 A100 GPU hours for running the workflow to generate data from the *DailyM* dataset and then compare the fine-tuned model using GPT-4 judge on *DailyM* test set. The results are shown in Table 9.

	Winning Rate		
AUGCON	64.5%	35.5%	ETRC
AUGCON	60.3%	39.7%	Con-Instr

Table 9: The winning rates in computation experiment.

In our comparative analysis, we discovered that our AUGCON consistently achieved higher winning rates when set against two other LLM-based, context-driven query extraction methodologies. This finding is particularly significant as it validates the efficacy of our method in yielding positive outcomes even with a reduced quantity of query-response pairs. The cornerstone of this enhanced performance lies in the superior quality and diversity of the data pairs generated by our method. Unlike conventional approaches that may produce voluminous but redundant data, our method focuses on creating data pairs that are both essential and varied. This strategic approach to data generation ensures that AUGCON operates efficiently, making the most of every data pair to contribute meaningfully to the fine-tuning of LLMs. As a result, AUGCON stands out as a highly data-efficient method, capable of fine-tuning LLMs with less generated data to achieve better performance. This advantage is especially crucial in scenarios where access to large amounts of the corpus or the computation resource is constrained. By eliminating redundancy and emphasizing the importance of quality and diversity, AUGCON paves the way for more effective and efficient context-based SFT data generation processes, ultimately leading to enhanced model performance even with limited sources.

E.4 Human Evaluation

In human evaluation on the test scenario, we meticulously curate a corpus dataset, referred to as the *DailyM* dataset, which consists of $1,000$ articles carefully selected from a variety of high-quality Chinese magazines closely related to daily life. These articles extensively cover issues of widespread public concern such as basic livelihood, politics, economics, and law, with each article containing approximately $4,000$ Chinese characters. Then, we test how well our method and baselines build an AI chat assistant specialized in this daily concern corpus. We apply our method on *DailyM* to generated SFT data called *DailyM-SFT* and use these data to fine-tune Qwen1.5-32B-Chat [7] to get fine-tuned model Qwen-DailyM-32B. To further test our method, we conduct annotators to write a total of $1,000$ queries they are interested in related to these articles, forming the *DailyM* test set.

E.4.1 Metrics

In our comprehensive evaluation framework, we assess both the generated queries and the outputs under the *DailyM* test set of the fine-tuned models to ensure a holistic understanding of the method's performance. Specifically, we evaluate the realism and diversity of generated queries and the relevance, accuracy, and satisfaction of fine-tuned models' outputs.

For both generated queries and model outputs, evaluators are provided with detailed scoring rubrics and examples to promote consistency in evaluation. The queries and outputs will be reviewed by multiple independent evaluators to ensure a balanced and objective assessment, with average scores calculated for each metric to determine the overall performance. Detailed annotation instructions are shown in Appendix F given space limitation.

E.4.2 Results

For all our baselines and the proposed AugCon, we employ Qwen1.5-32B-Chat [7] as the base model for calling and conducting fine-tuning for later evaluations. For methods such as AdaptLLM, ETRC, Context-Instruct, and our AugCon which generate query-response pairs based on context, we adhere to a standard where every 35 Chinese characters derive one query-response pair to ensure a fair comparison. We limit the number of generated entries to the same in the comparison because we find that all methods spend much more time on final fine-tuning process compared to the previous generation.

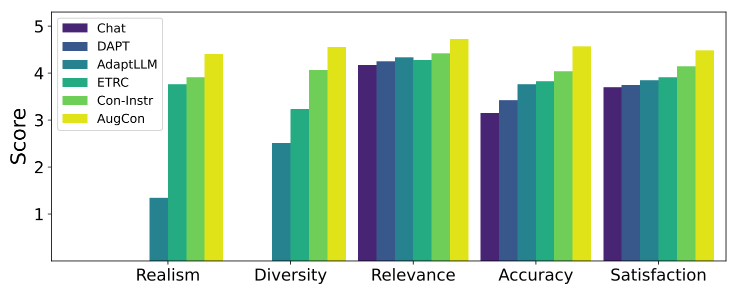


Figure 6: The results of human evaluation on *DailyM*. Query metrics are not applicable for the base chat model and DAPT so we don't show them.

Figure 6 presents the results of the human evaluation on the *DailyM* test set. The results demonstrate that AugCon consistently surpasses the baseline methods across all evaluation metrics. Specifically, the superior performance in terms of query realism and diversity underscores our method's ability to produce human-like and high-diversity queries. Since our CST and filtering process effectively gain multi-granularity queries that are more effective in covering all granularity levels of context, the derived data will extract more useful knowledge from the corpus. Furthermore, the impressive performance in judging relevance, accuracy, and satisfaction in responses from fine-tuned models further validates that our method's high-quality and diverse queries, coupled with high-fidelity responses, can indeed enhance the performance of subsequently fine-tuned models and achieve higher satisfaction scores from humans. This suggests that AugCon is particularly adept at constructing high-quality supervised fine-tuning data for LLMs from a given corpus.

F Human Evaluation Guidance

To establish a robust assessment framework for both generated queries and model outputs, we have devised an extensive human evaluation guideline shown in Table 10. Each score will also be accompanied by several corresponding examples, ensuring a consistent and objective evaluation process. This guideline emphasizes key metrics, including realism, diversity, relevance, accuracy, and satisfaction. By following this guide, evaluators can thoroughly assess the effectiveness of our method, guaranteeing the generation of high-quality, multi-granularity queries and responses. Our approach strives for comprehensive evaluations, aided by detailed scoring rubrics and examples to enable balanced decision-making.

Table 10: The guidance for human evaluation.

Score	Realism
5	The query is indistinguishable from those a human might ask. It is natural, authentic, and precisely the type of question a curious user would pose.
4	The query closely resembles real user inquiries, with minor differences. It maintains a high level of realism and naturalness.
3	The query shows moderate realism, differing somewhat from typical user questions. It still appears natural and understanding.
2	The query has noticeable deviations from real user questions, affecting its realism. It shows signs of artificiality but remains understandable.
1	The query is clearly artificial, lacking realism and naturalness. It differs significantly from how a real user would ask.
Score	**Diversity**
5	The queries exhibit exceptional diversity, covering a wide range of topics and varying greatly in their nature and specificity.
4	The queries show good diversity, exploring multiple topics and presenting different types of questions. They maintain a solid variety, even if not exhaustive.
3	The queries present moderate diversity, touching upon several topics but with some repetitiveness or predictability in their nature.
2	The queries show limited diversity, often sticking to a narrow range of topics or lacking variety in their structure and content.
1	The queries lack diversity, being highly repetitive, monotonous, and showing minimal to no variation in topics or approach.
Score	**Relevance**
5	The response is highly relevant, precisely addressing the query's intent and providing contextually appropriate information.
4	The response is mostly relevant, with minor deviations that do not significantly affect its overall alignment with the query.
3	The response shows moderate relevance, partially addressing the query but with some noticeable gaps or misalignments.
2	The response has limited relevance, straying significantly from the core of the query or providing only partially related information.
1	The response is irrelevant, failing to address the query's intent or providing information that is completely off-topic.
Score	**Accuracy**
5	The response is completely accurate, with no factual errors or hallucinations. All information provided is verifiable and aligns with external sources.
4	The response contains minor inaccuracies or minor hallucinations, but the overall information conveyed is mostly correct and reliable.
3	The response shows moderate accuracy, with some noticeable factual errors or hallucinations that don't significantly alter the main message.
2	The response has significant inaccuracies or hallucinations, affecting the overall reliability and correctness of the information provided.
1	The response is highly inaccurate, containing multiple factual errors or severe hallucinations that render the information untrustworthy.
Score	**Satisfaction**
5	The evaluator is highly satisfied with the responses. They fully meet expectations, leaving no room for improvement.
4	The evaluator is mostly satisfied. The responses are generally good, with only minor shortcomings or areas for improvement.
3	The evaluator feels moderately satisfied. The responses have notable strengths but also some weaknesses that need addressing.
2	The evaluator is somewhat dissatisfied. The responses show significant room for improvement and may not fully meet expectations.
1	The evaluator is highly dissatisfied. The responses fail to meet expectations on multiple levels, requiring substantial improvement.

Table 11: The guidance to categorize each granularity of questions.

Category	Description and Examples
Detail Questions	**Description:** Detail questions ask for specific information or facts about a particular aspect of a broader topic. These questions are often precise and seek exact answers. **Tips for Identification**: • Look for questions asking for “how,” “what,” “where,” “when,” or “who” in a specific context. • These questions usually focus on a narrow aspect rather than the whole topic. **Examples:** • How deep does the abortion needle penetrate? • What is the boiling point of mercury? • Who was the first person to climb Mount Everest?
Concept Questions	**Description:** Concept questions explore the understanding, differences, or definitions of ideas, theories, or terminologies. They are more about the “why” or “what is” than about specific details. **Tips for Identification**: • These questions often ask for explanations, comparisons, or definitions. • They seek to clarify concepts or understand the distinctions between them. **Examples:** • What is the difference between emergencies and crises? • How do you differentiate between classical and operant conditioning? • What does the term ’biodiversity’ encompass?
Macro Questions	**Description:** Macro questions address broad themes, trends, or impacts. They are expansive and consider the bigger picture, often relating to societal, global, or philosophical inquiries. **Tips for Identification**: • Look for questions that ask about “impact,” “influence,” “role,” or “importance.” • These questions are overarching and not limited to specific instances. **Examples:** • What impact does faith have on us? • How does climate change affect global agriculture? • What is the role of technology in shaping modern education?