CurLL: Curriculum Learning of Language Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce a comprehensive continual learning dataset and benchmark (CURLL) grounded in human developmental trajectories from ages 5–10, enabling fine-grained assessment of models' ability to acquire new skills. CURLL spans five developmental stages (0–4) covering ages 5–10, supported by a skill graph that breaks down broad skills into smaller abilities, concrete goals, and measurable indicators, while also capturing which abilities build on others. We generate a 23.4B-token synthetic dataset with controlled skill progression, vocabulary complexity, and format diversity, comprising paragraphs, comprehension-based QA (CQA), skill-testing QA (CSQA), and instruction–response (IR) pairs. Stage-wise token counts range from 2.12B to 6.78B tokens, supporting precise analysis of forgetting, forward and backward transfer. Using a 135M-parameter transformer trained under independent, joint, and sequential setups, we show trade-offs in skill retention and transfer efficiency.

1 Introduction

The capacity for lifelong learning in humans is not just a practical advantage but a fundamental aspect of intelligence itself [Kudithipudi et al., 2022, Yan et al., 2024, Schmidgall et al., 2023]. The continual learning (CL) problem thus is one of the grand challenges for achieving human-like artificial intelligence. It addresses the core problem of how computational systems can progressively acquire, integrate, and refine knowledge over extended periods without compromising earlier capabilities. For language models (LMs), this challenge is particularly interesting: despite their impressive performance across various tasks, these models face a fundamental limitation in that their skill-set and knowledge of the world become static after training, frozen at the point of deployment [Shi et al., 2024, Wu et al., 2024, Bell et al., 2025]. Despite the importance of the CL problem for LMs, current evaluation methodologies suffer from significant limitations: 1) Poor skill control: Existing benchmarks often lack precise control over the specific skills being tested, making it difficult to isolate the effects of learning new capabilities [Liu et al., 2025, Rivera et al., 2022]. 2) Unclear knowledge dependencies: The relationships between skills are rarely explicitly modeled, missing out on important transfer effects [Zheng et al., 2025, Nekoei et al., 2021]. 3) Inadequate forgetting metrics: Many evaluations fail to properly measure catastrophic forgetting across sequential learning tasks [Chen et al., 2023a, Huang et al., 2023].

To address these gaps, we introduce a dataset (CURLL) to train and evaluate continual learning algorithms for language models. Coming up with a set of skills with a rich structure and dependencies is a challenge in the construction of such a dataset. We find such a source of skills in human education. CURLL is grounded in the curriculum for human education from ages 5–10, divided into five developmental stages (0–4). Each of these stages represent one human-year. Our framework incorporates 1,300+ fine-grained skills with dependencies codified in a skill graph having skills as nodes with the edges capturing a prerequisite relationship. The edges are weighted on a scale of (1–5) to capture dependency strength. Starting from this set of skills, we generate a synthetic

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

dataset of 23.4B tokens, with controlled vocabulary complexity (stage-specific word sampling from Age-of-Acquisition data as seed) and multiple formats (paragraphs, comprehension QA, skill-testing QA, instruction–response). Each stage's dataset ranges from 2.12B to 6.78B tokens, enabling fine-grained evaluation at indicator, skill, and stage levels. Our code, dataset (stages 0–4), and skill graph will be publicly released. Our contributions include: a) The idea of grounding skills in human education curriculum in the context of CL. b) A synthetic data generation pipeline spanning 5 developmental stages with stage-specific vocabulary and explicit skill dependencies. This pipeline gives us a benchmark with fine-grained control over measuring skill transfer, forgetting and sample efficiency c) A skill graph-based dependency model that explicitly captures prerequisite relationships between learning objectives, enabling nuanced analysis of skill transfer and forgetting.

2 Related Work

Many datasets and benchmarks exist for continual learning of LMs [Jang et al., 2021, Li et al., 2025]. TRACE [Wang et al., 2023] highlights that existing benchmarks are too simple or are already included in instruction-tuning sets. MMLM-CL [Zhao et al., 2025] notes the limited real world applicability in benchmarks. OCKL [Wu et al., 2023] proposes new metrics for measuring knowledge acquisition rate and knowledge gap but concentrates on knowledge-intensive tasks as compared to procedural tasks. TemporalWiki [Jang et al., 2022] is for updating factual information in LMs based on temporal data. SuperNI contains a variety of traditional NLP tasks and serves as a practical benchmark for continual learning of large language models [He et al., 2024]. Despite these developments, these benchmarks are often considered unsuitable for evaluating state-of-the-art LMs [Wang et al., 2023, Razdaibiedina et al., 2023, Scialom et al., 2022, Zhang et al., 2015]. These benchmarks often emphasize artificial task boundaries He et al. [2024], lack temporal and distributional complexity. Moreover, these datasets do not offer precise control over skills or information to validate the effectiveness of existing solutions for continual learning. Skill-it [Chen et al., 2023b] emphasizes the problem but only introduces a data sampling algorithm for continual pretraining by arranging the skills in a increasing order of complexity. Other existing works [Khetarpal et al., 2020, Greco et al., 2019, Xu et al., 2024] discuss the importance of skill distinction and its effect on evaluating continual learning. In contrast, our work is grounded in human developmental curricula and enables fine-grained evaluation of transfer, forgetting, and sample efficiency beyond what existing benchmarks support.

3 Dataset Setup

Our framework is grounded in human learning curriculum, with the dataset designed to mimic the developmental stages from age 5-10. We use two established educational frameworks to develop our skill taxonomy: the Early learning Outcomes framework (ELOF) for children aged 5[1] and the Cambridge curriculum for children aged 5-10[2]. These frameworks help us define fine grained notion of skills as specified by a skill-tuple that consists of four components: 1) Skills[3]: High-level domains or subjects (e.g. Mathematics, Science). 2) Sub-skills: Specific components within a skill (e.g., Counting and Cardinality). 3) Goals: Broad statement of learning expectations within a sub-skill. 4) Indicators: Specific, observable behaviors that demonstrate mastery of a goal.

The ELOF framework has five broad areas: Approaches to Learning, Social and Emotional Development, Language and Literacy, Cognition, and Perceptual, Motor, and Physical Development. Cambridge Primary Curriculum covers subjects including English, Mathematics, Science, Computing, and Global Perspectives. The curriculum structure flows from subjects (renamed as skills in our framework) to domains/strands (renamed as subskills), then to substrands (goals), each with specific learning objectives (indicators). We also adopt the notion of stages from the Cambridge curriculum in our framework, where each stage corresponds to one year starting from age 5. Therefore, we have 5 stages in our framework, where stage 0 denotes ages up to 5, stage 1 denotes age 5-6 and so on. The number of skill-tuples in our framework is the same as the number of indicators present in stages 0-4, statistics of which are mentioned in Table 1. We construct a skill graph, which is a directed graph that has indicators as nodes, with edges representing prerequisite relationships weighted from 1-5 to indicate dependency strength. These edges model how skills are built on each other in developmental

[1] U.S. Department of Health & Human Services, Administration for Children & Families [2024]

[2] Cambridge Assessment International Education [2025]

[3] "Skill" here has a specific meaning, which is different from the general notion of skill used before

Table 1: Dataset statistics across developmental stages (0–4), including total tokens

Stage	Skills & Goals				Instances			# Tokens in Bn
	# Skills	# Sub-skills	# Goals	# Indicators	# CQA	# CSQA	# IR Pairs	
0	7	24	59	182	1.0M	3.01M	3.30M	2.12
1	7	29	86	292	20.2M	4.04M	4.10M	3.47
2	6	26	67	249	23.5M	4.70M	4.78M	4.56
3	6	26	68	271	31.2M	6.24M	6.29M	6.47
4	6	23	70	349	27.4M	5.49M	5.52M	6.78

stages[4]. While the skill graph isn't directly used for skill data generation, it provides insights for analyzing continual learning patterns and interpreting evaluation results (see Appendix A).

3.1 Synthetic Data Generation

Our synthetic data consists of instances, each mimicking a situation a child might encounter. Instances are of three types: (1) IR: an instruction-response pair, where the instruction is about some general world knowledge, (2) CQA: context-based question-answers for testing comprehension, (3) CSQA: context-based question-answers for testing skills. A context is a short piece of text which forms the basis of the corresponding question-answer pairs in the instance. Contexts can be of multiple types as specified by a template: e.g., a simple narrative, or a dialogue. IR-pairs can also have different types specified by templates, e.g., mimic action or follow simple direction (Appendix A for examples).

Instances are generated by prompting an LLM with a *seed*. A seed consists of a skill-tuple, vocabulary seed, instance type, template. This choice is crucial for ensuring diversity and coverage of our data. This tuple is also our way to ground the generations in the skill graph. To generate one instance of the data, we first construct a seed: each skill-tuple is combined with a vocabulary seed for that stage, an instance type and a template for that instance type. If the instance type is CQA or CSQA, then we first generate the context, and then using the context, we generate the corresponding question-answers. If the instance type is IR then we directly generate the instruction-response pairs. The prompts for all the generations and details of the generation process are presented in Appendix A. In our dataset, each instance includes the seed used to generate it as part of its metadata. We generated data for stages 0-4, containing a total of 23.4B tokens (Table 1).

We measure diversity of generated data using: 1) Diversity as reciprocal of compression ratio using gzip Gailly and Adler [1992]. 2) The intra- and inter-text deduplication rate as calculated by semantic deduplication. Cross-stage analysis shows higher diversity and lower deduplication rate (<5%) between stages compared to intra-stage results, confirming that content evolves meaningfully across developmental progression while maintaining stage-specific uniqueness. See Appendix B for more details. We also measure progression in the difficulty of the skills as the stage number increases. We sample 500K instances from each stage for each data type and run statistical readability tests[5]. Means across multiple readability metrics are reported in Appendix B.3. The readability tests show that as stages progress, the texts also become increasingly challenging. At least 50 random instances from each dataset per stage were manually analysed, revealing that CQA data for all stages was found to be accurate. IR and CSQA data had certain patterns like excessive use of discourse markers for early stages and verbose response to instructions. We choose 25 instances per indicator for test set resulting in 5k-7k samples per stage. Since the data is synthetically generated at scale, we reserve the highest quality samples for the test set. 100 instances per indicator instance type are sampled randomly and rated by LLM on a scale 1-5. Top 25 instances are selected for test, next 25 for validation set.

4 Experiments and Results

We conduct preliminary experiments to validate that the dataset exposes meaningful challenges: whether models can retain earlier-learned skills, how sequential training affects generalization, and to what extent transfer across related skills occurs. By analyzing these at the granularity of skills, we demonstrate that CURLL enables insights that are not visible in existing benchmarks. Unlike

[4] an LLM (Gemma3-27B-IT is used for all LLM inferences throughout this work) is used to predict the edges

[5] Uses pre-defined words to predict the grade of a text (https://github.com/cdimascio/py-readability-metrics)

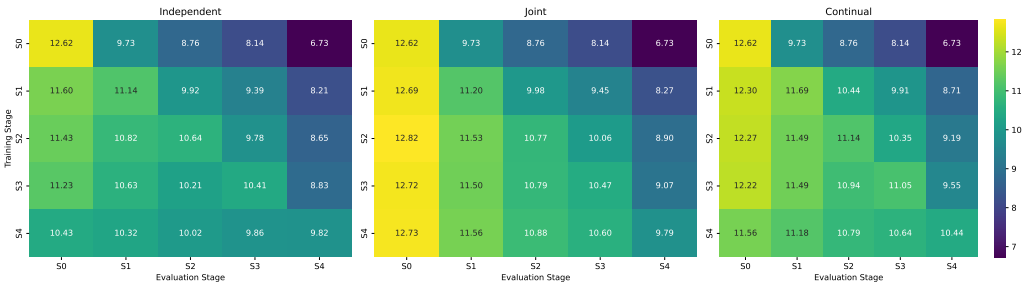

Figure 1: Stage-wise results for all training setups. Independent corresponds to models trained on a single stage, joint to models trained on mixtures of data up to a stage, and continual to sequential upto a stage. Heatmaps report summed correctness scores across all test formats (IR, CQA, CSQA)

traditional language model training that includes two stages: pretraining and then finetuning, we do a single phase training. All instance types i.e. CQA, CSQA, and IR are included in the same phase. Since all of them are question-answers, with and without context paragraphs, we use a standard chat template to train the language models from scratch. Smollm2-135M parameter model is used as the base architecture. All training runs are performed on one full epoch of the data. Learning rate of 5e-3 and effective batch size of 1536 instances remain unchanged across experiments. We use a context length of 1024. Other training and inference related hyper-parameters are mentioned in the Appendix C. We perform three types of training: 1) Independent (M_i): The model is trained from scratch on data of each stage independently 2) Joint (M_{ij}): Jointly trained on a mixture of stages. The data from different stages is combined and shuffled randomly. 3) Continual (M_{i-j}): The model is first trained on stage i, then stage j, then stage k and so on. To evaluate the trained models, the instances from test set are passed through the chat template and the model is asked to complete the generation post instruction. These inferences along with the prompt is passed to an LLM to rate on a scale of 1-5. This is followed for all three types of test sets. Each model is evaluated on test sets of all stages. The main objective of the rating is to evaluate the correctness of the model inference with some weightage to the stage on which the model is being evaluated. The summation of scores across test set types (IR, CQA, CSQA) is presented in Figure 1. The individual scores, prompts and rubrics for evaluation are available in the Appendix E.

Joint models (M_{ij}) generalize better to later stages and maintain strong performance on trained stages compared to independent models (M_i). Continual models (M_{i-j}), however, achieve the best performance on later stages but suffer degradation on earlier ones. Sequential (continual) ordering improves generalization but also induces forgetting of earlier skills, which is counter-intuitive since later skills depend on foundational ones. The skill graph helps explain this. The largest performance gaps between joint and continual training occur for "Perceptual, Motor, and Physical Development" and "Digital Literacy". Both have very few outgoing edges in the skill graph (Appendix D), meaning their indicators rarely serve as prerequisites for later skills.

5 Conclusion

We introduced (CURLL), a novel continual learning evaluation framework for language models grounded in human developmental curricula. (CURLL) combines a directed, weighted skill graph of over 1,300 fine-grained skills with a 23.4B-token synthetic dataset that controls stage-wise vocabulary, difficulty, and format. The skill graph serves as a diagnostic tool: its metadata enables fine-grained control over the number of instances and skills seen during training, supports evaluation of sample efficiency, and allows targeted testing of transfer effects (e.g., whether learning Skill A improves Skill B). Forgetting, forward transfer, backward transfer, and data efficiency can all be measured at the levels of skills, sub-skills, and indicators. This enables richer analysis than stage- or task-level metrics in existing benchmarks, which typically report only overall accuracy on entire tasks (e.g., classification or QA) without revealing which underlying abilities are gained or lost. Our experiments with independent, joint, and sequential training demonstrate that simply changing the order of data presentation affects both generalization and forgetting. Finally, the scalable data generation pipeline enables exploring continual pretraining in a controlled yet realistic setting.

References

Jack Bell, Luigi Quarantiello, Eric Nuertey Coleman, Lanpei Li, Malio Li, Mauro Madeddu, Elia Piccoli, and Vincenzo Lomonaco. The future of continual learning in the era of foundation models: Three key directions. *ArXiv*, abs/2506.03320, 2025. URL https://api.semanticscholar.org/CorpusId:279155222.

Cambridge Assessment International Education. International curriculum. https://www.cambridgeinternational.org/why-choose-us/benefits-of-a-cambridge-education/international-curriculum/, 2025. URL https://www.cambridgeinternational.org/why-choose-us/benefits-of-a-cambridge-education/international-curriculum/. Accessed: 2025-08-20.

Ernie Chang, Matteo Paltenghi, Yang Li, Pin-Jie Lin, Changsheng Zhao, Patrick Huber, Zechun Liu, Rastislav Rabatin, Yangyang Shi, and Vikas Chandra. Scaling parameter-constrained language models with quality data. *ArXiv*, abs/2410.03083, 2024. URL https://api.semanticscholar.org/CorpusID:273162494.

Jiefeng Chen, Timothy Nguyen, Dilan Gorur, and Arslan Chaudhry. Is forgetting less a good inductive bias for forward transfer? *ArXiv*, abs/2303.08207, 2023a. URL https://api.semanticscholar.org/CorpusId:257532608.

Mayee F. Chen, Nicholas Roberts, K. Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. Skill-it! a data-driven skills framework for understanding and training language models. *ArXiv*, abs/2307.14430, 2023b. URL https://api.semanticscholar.org/CorpusId:260203057.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english?, 2023. URL https://arxiv.org/abs/2305.07759.

Jean Gailly and Mark Adler. GNU gzip. GNU Operating System, 1992.

Claudio Greco, Barbara Plank, R. Fernández, and R. Bernardi. Psycholinguistics meets continual learning: Measuring catastrophic forgetting in visual question answering. In *Annual Meeting of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusId:184488333.

Jinghan He, Haiyun Guo, Kuan Zhu, Zihan Zhao, Ming Tang, and Jinqiao Wang. Seekr: Selective attention-guided knowledge retention for continual learning of large language models. In *Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://api.semanticscholar.org/CorpusID:273901289.

Heng Huang, Li Shen, Enneng Yang, and Zhenyi Wang. A comprehensive survey of forgetting in deep learning beyond continual learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47:1464–1483, 2023. URL https://api.semanticscholar.org/CorpusId:259951356.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. Towards continual knowledge learning of language models. *ArXiv*, abs/2110.03215, 2021. URL https://api.semanticscholar.org/CorpusId:238419458.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. In *Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://www.aclanthology.org/2022.emnlp-main.418.pdf.

Khimya Khetarpal, M. Riemer, I. Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *J. Artif. Intell. Res.*, 75:1401–1476, 2020. URL https://api.semanticscholar.org/CorpusId:229679944.

D. Kudithipudi, Mario Aguilar-Simon, Jonathan Babb, M. Bazhenov, Douglas Blackiston, J. Bongard, Andrew P. Brna, Suraj Chakravarthi Raja, Nick Cheney, J. Clune, A. Daram, Stefano Fusi, Peter Helfer, Leslie M. Kay, Nicholas A. Ketz, Z. Kira, Soheil Kolouri, J. Krichmar, Sam Kriegman, Michael Levin, Sandeep Madireddy, Santosh Manicka, Ali Marjaninejad, Bruce L. McNaughton, R. Miikkulainen, Zaneta Navratilova, Tej Pandit, Alice Parker, Praveen K. Pilly, S. Risi, T. Sejnowski, Andrea Soltoggio, Nicholas Soures, A. Tolias, Darío Urbina-Meléndez, F. Valero-Cuevas, Gido M. van de Ven, J. Vogelstein, Felix Wang, Ron Weiss, A. Yanguas-Gil, Xinyun Zou, and H. Siegelmann. Biological underpinnings for lifelong learning machines. *Nature Machine Intelligence*, 4:196 – 210, 2022. URL https://doi.org/10.1038/s42256-022-00452-0.

Jeffrey Li, Mohammadreza Armandpour, Iman Mirzadeh, Sachin Mehta, Vaishaal Shankar, Raviteja Vemulapalli, Samy Bengio, Oncel Tuzel, Mehrdad Farajtabar, Hadi Pouransari, and Fartash Faghri. Tic-lm: A web-scale benchmark for time-continual llm pretraining. *ArXiv*, abs/2504.02107, 2025. URL https://api.semanticscholar.org/CorpusId:277510618.

Jia Liu, Jinguo Cheng, Xiangming Fang, Zhenyuan Ma, and Yuankai Wu. Evaluating temporal plasticity in foundation time series models for incremental fine-tuning. *ArXiv*, abs/2504.14677, 2025. URL https://api.semanticscholar.org/CorpusId:277954770.

Hadi Nekoei, Akilesh Badrinaaraayanan, Aaron C. Courville, and Sarath Chandar. Continuous coordination as a realistic scenario for lifelong learning. *ArXiv*, abs/2103.03216, 2021. URL https://api.semanticscholar.org/CorpusId:232110854.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. Progressive prompts: Continual learning for language models. *ArXiv*, abs/2301.12314, 2023. URL https://api.semanticscholar.org/CorpusID:256390383.

Corban G. Rivera, C. Ashcraft, Alexander New, J. Schmidt, and Gautam K. Vallabha. Latent properties of lifelong learning systems. *ArXiv*, abs/2207.14378, 2022. URL https://api.semanticscholar.org/CorpusId:251196582.

Samuel Schmidgall, Jascha Achterberg, Thomas Miconi, Louis Kirsch, Rojin Ziaei, S. P. Hajiseyedrazi, and Jason Eshraghian. Brain-inspired learning in artificial neural networks: a review. *ArXiv*, abs/2305.11252, 2023. URL https://api.semanticscholar.org/CorpusId:258823273.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. Fine-tuned language models are continual learners. In *Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://api.semanticscholar.org/CorpusID:252815378.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 2024. URL https://api.semanticscholar.org/CorpusId:269362836.

U.S. Department of Health & Human Services, Administration for Children & Families. Head start early learning outcomes framework: Ages birth to five, 2024. URL https://headstart.gov/school-readiness/article/head-start-early-learning-outcomes-framework. Last updated: December 23, 2024; Accessed: 2025-08-20.

Xiao Wang, Yuan Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. Trace: A comprehensive benchmark for continual learning in large language models. *ArXiv*, abs/2310.06762, 2023. URL https://api.semanticscholar.org/CorpusID:263830425.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. Continual learning for large language models: A survey. *ArXiv*, abs/2402.01364, 2024. URL https://api.semanticscholar.org/CorpusId:267406164.

Yuhao Wu, Tongjun Shi, Karthick Sharma, Chun Seah, and Shuhao Zhang. Online continual knowledge learning for language models. *ArXiv*, abs/2311.09632, 2023. URL https://api.semanticscholar.org/CorpusId:265221422.

Yongxin Xu, Philip S. Yu, Zexin Lu, Xu Chu, Yujie Feng, Bo Liu, and Xiao-Ming Wu. Klf: Knowledge localization and fusion for language model continual learning. 2024. URL https://api.semanticscholar.org/CorpusId:271843361.

Lixiang Yan, Samuel Greiff, Ziwen Teuber, and D. Gaević. Promises and challenges of generative artificial intelligence for human learning. *Nature human behaviour*, 8 10:1839–1850, 2024. URL https://api.semanticscholar.org/CorpusId:271924303.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Neural Information Processing Systems*, 2015. URL https://api.semanticscholar.org/CorpusID:368182.

Hongbo Zhao, Fei Zhu, Rundong Wang, Gaofeng Meng, and Zhaoxiang Zhang. Mllm-cl: Continual learning for multimodal large language models. *ArXiv*, abs/2506.05453, 2025. URL https://api.semanticscholar.org/CorpusId:279243888.

Junhao Zheng, Xidi Cai, Qiuke Li, Duzhen Zhang, Zhongzhi Li, Yingying Zhang, Le Song, and Qianli Ma. Lifelongagentbench: Evaluating llm agents as lifelong learners. *ArXiv*, abs/2505.11942, 2025. URL https://api.semanticscholar.org/CorpusId:278739762.

A Dataset Construction

Figure 2 gives an overview of our dataset, including examples of skills, subskills, goal and indicator. We also present an example of an edge from the skill graph. Figure 3 shows the number of incoming and outgoing edges from each stage. Figure 4 explains the data construction process and gives examples from each stage of the data generation pipeline.

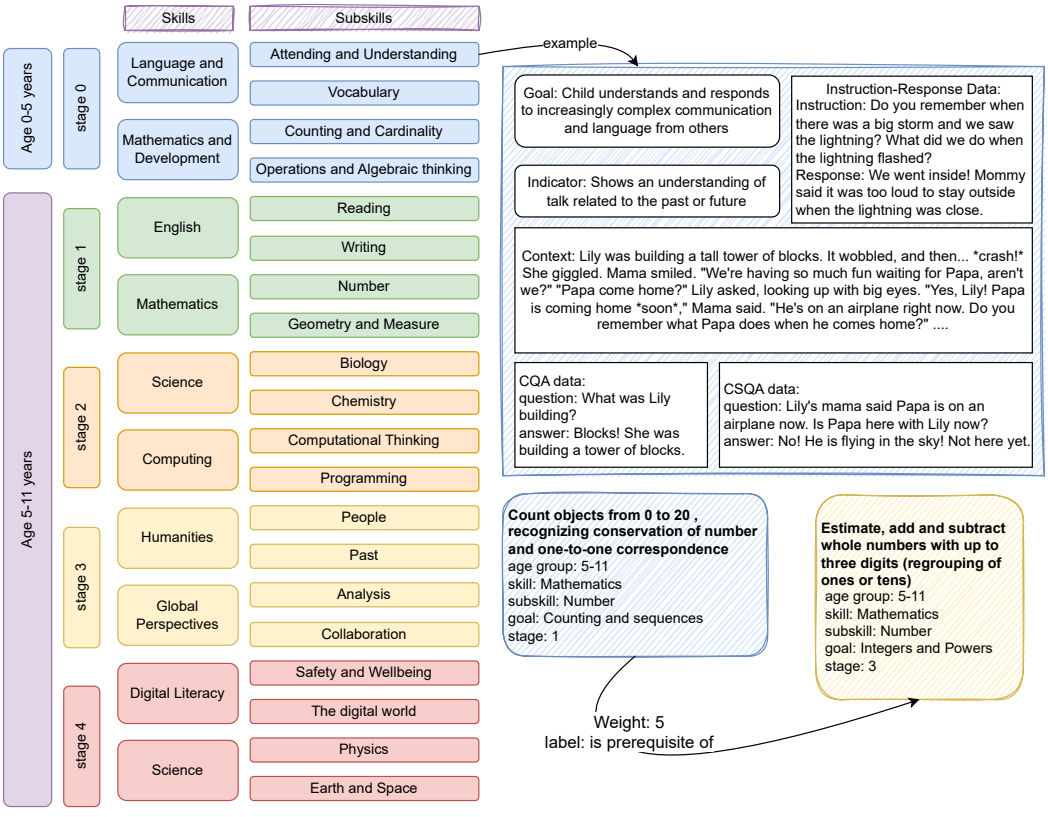

Figure 2: Developmental framework for children aged 0-11 years, categorized into stages (0-4). Only examples of skills and subskills are mentioned here. An example of how the data looks like is given in the top right. Two nodes and an edge from the skill graph is given in the bottom right.

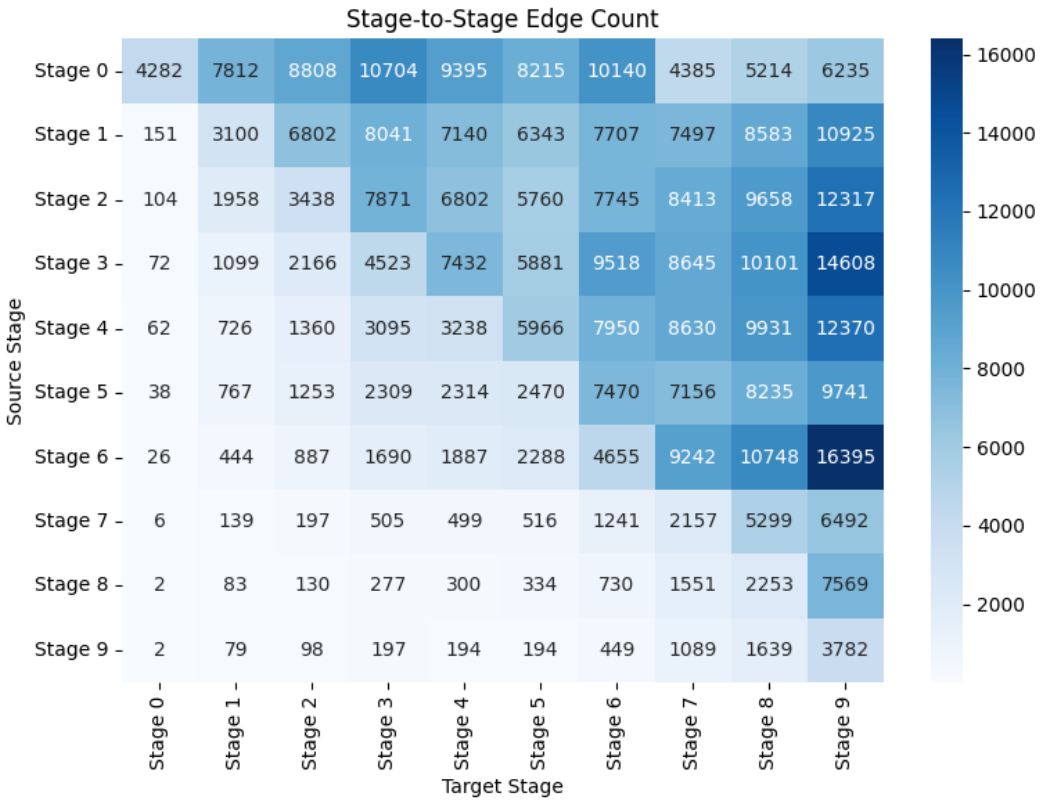

Figure 3: Heatmap showing the number of prerequisite edges between stages in the skill graph. Rows correspond to source stages, columns to target stages, and color intensity indicates the number of connections.

B Data Verification

For both the methods, 500K texts are sampled from each of the Paragraphs and Instruction-response pairs.

B.1 Diversity

For the diversity of the text, we follow Chang et al. [2024] and calculate the compression ratio of the text as

$$\mathrm{CR}(D) = \frac{Originalsizeof D\ (bytes)}{Compressedsizeof D\ (bytes)},$$

and define diversity by

$$\mathrm{Dr}(D) = 1/\mathrm{CR}(D).$$

A higher compression ratio $\mathrm{CR}(D)$ indicates greater redundancy, meaning lower diversity in the text. Thus, diversity $\mathrm{Dr}(D)$ increases when redundancy decreases. We see diversity ranging between 30.77% and 35.60%, which is similar to other work. As a comparison, we also calculated the diversity of 500K samples from the validation set of TinyStories, a paper exploring synthetic data generation to train a small language model. Their text diversity ranges from 31.04% to 32.66% within the pretraining and instruct data, respectively Eldan and Li [2023].

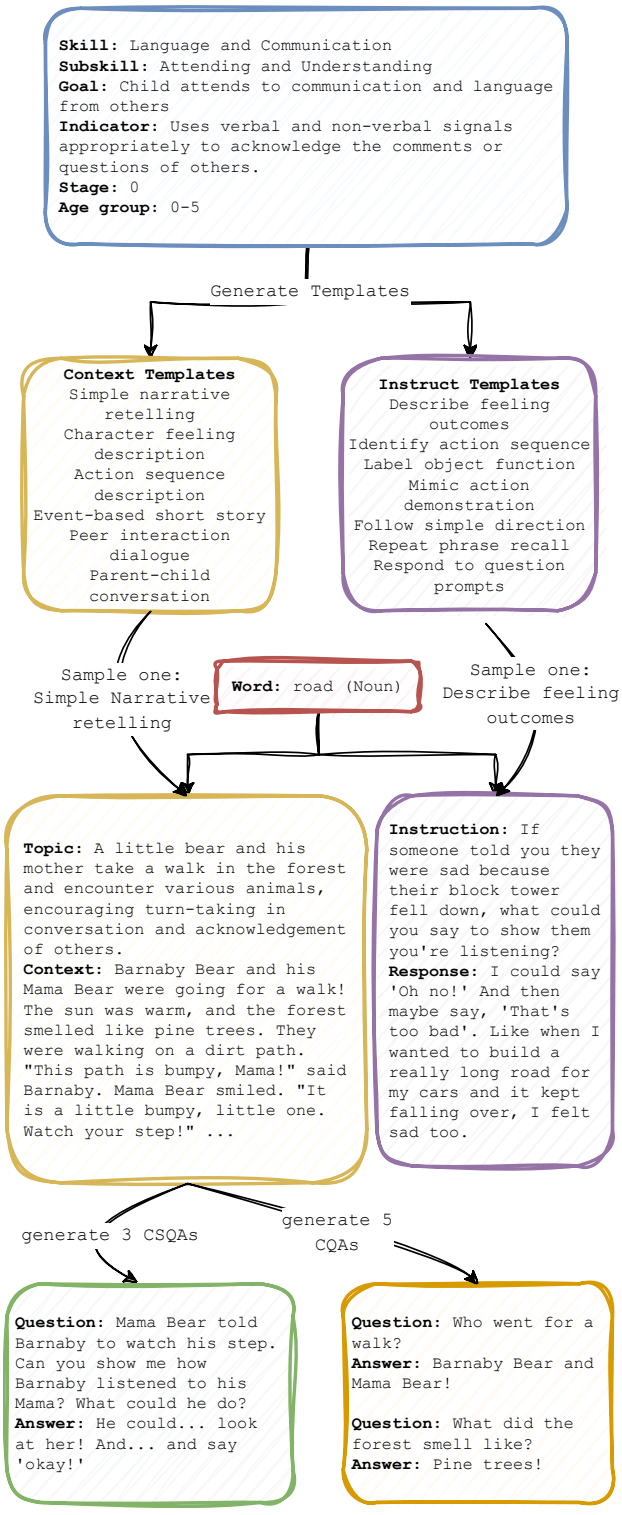

Figure 4: Synthetic data generation pipeline

Table 2: Diversity and Deduplication metrics for context and instruction–response data across stages

Stage	Context		IR	
	Div ↑	Dedup ↓	Div ↑	Dedup ↓
0	34.29%	11.83%	30.77%	3.50%
1	35.60%	5.36%	31.73%	3.85%
2	34.17%	15.47%	32.64%	2.54%
3	34.68%	14.86%	32.97%	2.09%
4	35.45%	13.41%	33.14%	1.93%

B.2 Deduplication

For semantic deduplication[6], we pass the texts through a sentence encoder and find the deduplication rate as the percentage of sentences that have cosine similarity of at least 0.95 with another sentence in the same stage.

Table 3: Diversity and Deduplication Rates when Considering Pairwise Stages

Stage Pair	Context	
	Div ↑	Dedup ↓
0, 1	31.29%	0.3%
0, 2	31.96%	0.1%
0, 3	32.25%	0.0%
0, 4	32.50%	0.0%
1, 2	32.27%	0.3%
1, 3	32.52%	0.2%
1, 4	32.71%	0.1%
2, 3	32.82%	0.4%
2, 4	32.94%	0.2%
3, 4	33.07%	0.2%

B.3 Detailed Readability Metrics

Note that average grade of the data is slightly higher than the intended age of the data (especially for the first few stages). However, this is because not all skills we generate data for are, in real-life, text-based. Thus, demonstrating them in language ends up requiring complex words, which affects the readability score. For example, children can verbally reason about cause-and-effect in multi-turn conversations, but when written down, that same dialogue is rated at a much higher reading level than the child can actually read, leading to higher readability scores in our data.

Table 4: Average readability scores of generated data across stages, reported for context, comprehension QA (CQA), skill-testing QA (CSQA), and instruction–response (IR) data. Scores generally increase with stage, reflecting controlled growth in textual complexity aligned with developmental progression

Stage	Context	CQA	CSQA	IR
0	4.61 1.87	2.38 2.88	3.07 2.26	4.48 1.52
1	5.24 1.72	4.39 1.81	4.44 1.62	4.86 1.41
2	5.18 1.93	4.39 1.80	4.69 1.54	4.69 1.59
3	5.51 1.85	4.65 1.70	4.98 1.46	5.03 1.50
4	6.42 1.79	5.63 1.44	5.96 1.30	5.91 1.34

[6]We use the following repo for semantic deduplication: https://github.com/MinishLab/semhash

Table 5: Detailed Readability Metrics Across all 5 Stages and Datasets

Dataset	Stage	Flesch Kincaid	SMOG	Coleman Liau	Automated Readability	Dale Chall	Gunning Fog
Context	0	3.15 0.35	6.90 0.76	4.28 0.51	1.68 0.47	6.69 0.27	4.94 0.34
Context	1	3.68 0.35	7.55 0.73	5.18 0.50	2.58 0.49	6.70 0.29	5.74 0.35
Context	2	3.80 0.36	7.54 0.75	4.21 0.46	2.25 0.48	7.18 0.35	6.12 0.38
Context	3	4.16 0.36	7.84 0.74	4.58 0.48	2.71 0.50	7.27 0.36	6.48 0.38
Context	4	5.13 0.42	8.76 0.79	5.39 0.51	3.77 0.56	7.89 0.34	7.58 0.45
CQA	0	0.79 0.35	5.06 0.59	0.26 0.59	-1.47 0.43	6.89 0.30	2.75 0.34
CQA	1	2.73 0.37	6.45 0.59	4.10 0.54	1.65 0.49	6.47 0.26	4.92 0.45
CQA	2	2.74 0.38	6.42 0.59	4.00 0.53	1.67 0.50	6.44 0.28	5.07 0.45
CQA	3	3.04 0.37	6.66 0.59	4.37 0.52	2.08 0.49	6.41 0.27	5.36 0.44
CQA	4	4.08 0.38	7.54 0.59	5.59 0.48	3.52 0.49	6.50 0.25	6.54 0.47
CSQA	0	1.34 0.28	5.37 0.70	2.07 0.43	-0.20 0.36	6.21 0.20	3.65 0.27
CSQA	1	2.84 0.30	6.36 0.71	4.14 0.37	2.04 0.40	6.03 0.21	5.24 0.33
CSQA	2	3.16 0.29	6.54 0.72	4.33 0.37	2.43 0.39	6.08 0.23	5.59 0.33
CSQA	3	3.49 0.29	6.81 0.70	4.64 0.37	2.87 0.39	6.14 0.25	5.96 0.32
CSQA	4	4.62 0.33	7.72 0.72	5.56 0.41	4.25 0.46	6.50 0.27	7.12 0.37
IR	0	2.97 0.47	6.32 0.64	4.12 0.52	2.25 0.62	5.81 0.23	5.43 0.48
IR	1	3.40 0.45	6.61 0.65	4.51 0.50	2.88 0.61	5.76 0.25	6.02 0.50
IR	2	3.16 0.37	6.62 0.72	4.23 0.45	2.33 0.50	6.10 0.26	5.68 0.43
IR	3	3.55 0.37	6.93 0.71	4.62 0.46	2.87 0.51	6.13 0.27	6.09 0.42
IR	4	4.59 0.41	7.66 0.73	5.41 0.46	4.13 0.56	6.46 0.28	7.20 0.47

C Hyperparameters

All experiments were conducted with a consistent set of training hyperparameters to ensure comparability across runs. Models were initialized using the kaiming normal method unless otherwise specified, and trained with AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1e-8$) with weight decay of 0.01. We used a base learning rate of $5e-3$, applied gradient clipping with a maximum norm of 1.0. We used gradient accumulation (8 steps with batch size 24 on 8 GPUs, yielding an effective batch size of 1536). Training was performed for one full epoch over each dataset split with a context length of 1024 tokens. Mixed precision was enabled with bfloat16 (bf16) for efficiency, while fp16 was disabled. All experiments were seeded with 42 for reproducibility. For inference, the model was loaded in bfloat16 precision with padding set to the EOS token and leftside padding for alignment. Prompts were tokenized with a maximum length of 512 tokens, and generation used a temperature of 0.7, top-p sampling of 0.95, and a maximum of 128 new tokens per prompt.

D Results

Table 6 gives the results of all experiments on IR test set. Table 7 gives the results of all experiments on CQA test set. Table 8 gives the results of all experiments on CSQA test set. Per-stage per-Indicator results can be found here: Results sheet. Forgetting analysis is shown in Figure 5. Relation of forgetting analysis to the skill graph can be drawn from Figure 6.

E Prompts

E.1 Edge Prediction

System prompt for Edge prediction

```
You are an expert in skill development and cognitive science. Your task is to
    analyze the relationship between two skill indicators and determine if there is
     a logical prerequisite dependency between them.

Each skill indicator is given with:
```

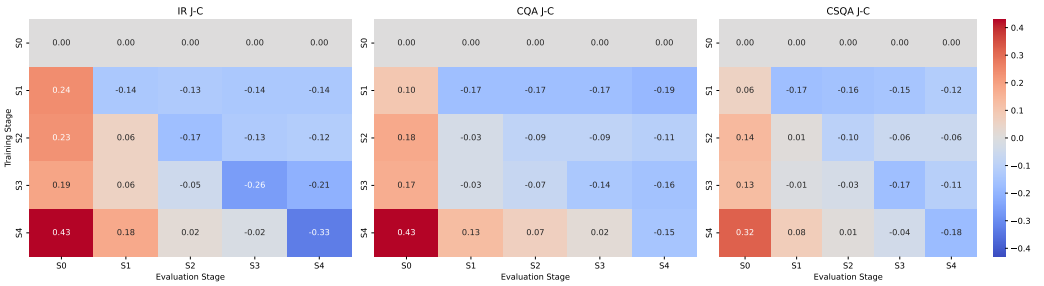

Figure 5: Forgetting analysis across training setups. The plots show performance differences between joint and continual training for IR, CQA, and CSQA test sets across stages 0–4. The Y-axis corresponds to models trained upto a stage. The X-axis corresponds to test set of mentioned stage.

Table 6: All results for IR test set. The column represents each stage on which a model is being evaluated.

Test type	IR (rating out of 5)				
Stages	0	1	2	3	4
M_0	4.16	3.29	2.97	2.83	2.49
M_1	3.70	3.70	3.21	3.08	2.80
M_2	3.71	3.55	3.56	3.27	3.00
M_3	3.64	3.45	3.35	3.57	3.07
M_4	3.38	3.35	3.32	3.34	3.55
M_{012}	4.22	3.81	3.55	3.34	3.07
M_{01}	4.19	3.73	3.25	3.12	2.84
M_{0-1}	3.94	3.87	3.38	3.26	2.98
M_{0123}	4.15	3.79	3.56	3.55	3.14
M_{0-1-2}	3.99	3.75	3.72	3.47	3.19
M_{01234}	4.16	3.80	3.60	3.60	3.46
$M_{0-1-2-3}$	3.97	3.73	3.61	3.82	3.34
$M_{0-1-2-3-4}$	3.73	3.63	3.58	3.62	3.78

Table 7: All results for CQA test set. The column represents each stage on which a model is being evaluated.

Test type	CQA (rating out of 5)				
Stages	0	1	2	3	4
M_0	4.16	3.29	2.97	2.83	2.49
M_1	3.70	3.70	3.21	3.08	2.80
M_2	3.71	3.55	3.56	3.27	3.00
M_3	3.64	3.45	3.35	3.57	3.07
M_4	3.38	3.35	3.32	3.34	3.55
M_{012}	4.22	3.81	3.55	3.34	3.07
M_{01}	4.19	3.73	3.25	3.12	2.84
M_{0-1}	3.94	3.87	3.38	3.26	2.98
M_{0123}	4.15	3.79	3.56	3.55	3.14
M_{0-1-2}	3.99	3.75	3.72	3.47	3.19
M_{01234}	4.61	4.27	4.05	3.87	3.45
$M_{0-1-2-3}$	4.42	4.27	4.09	3.97	3.45
$M_{0-1-2-3-4}$	4.17	4.14	3.97	3.85	3.60

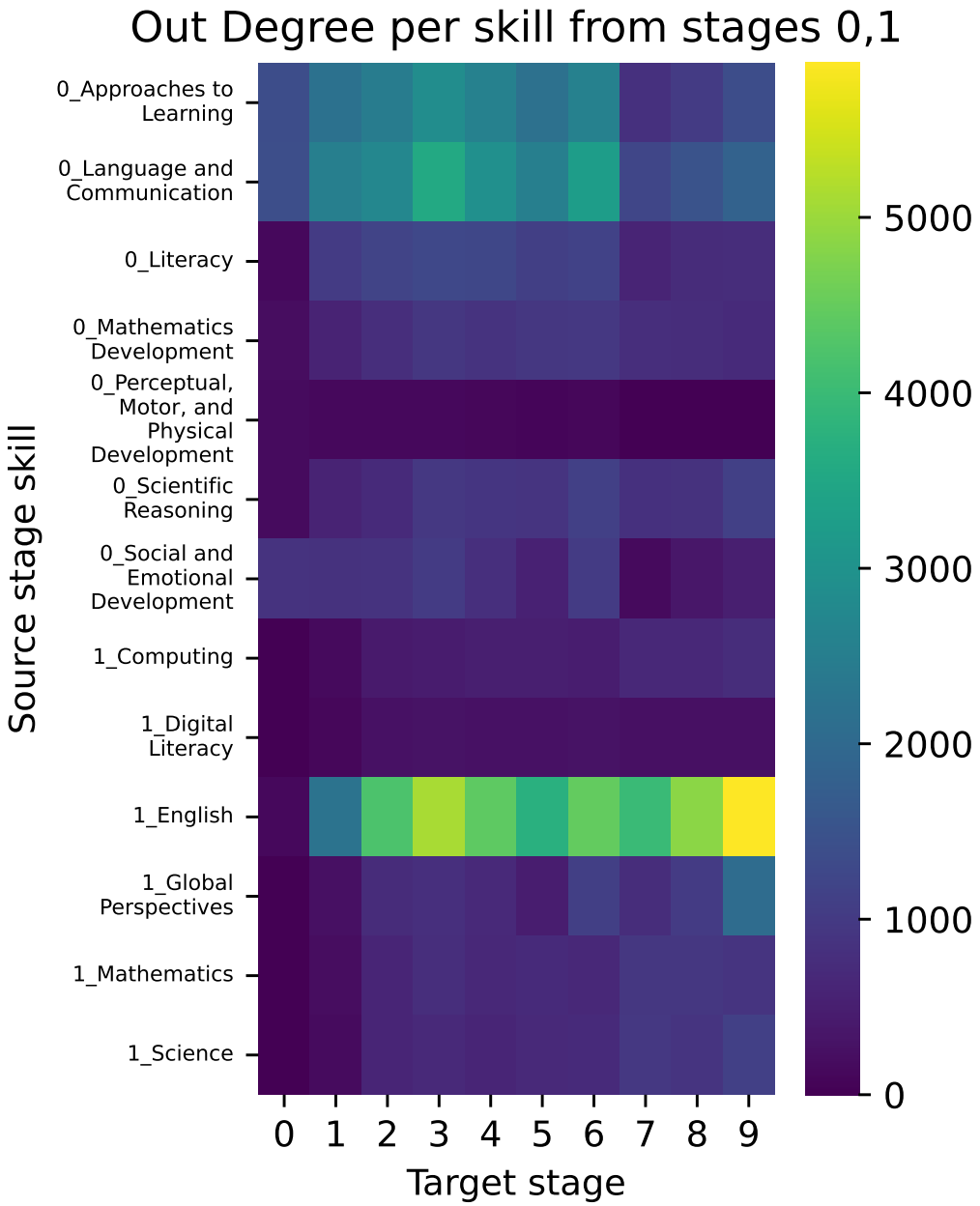

Figure 6: Out-degree distribution of skills from stages 0 and 1 in the skill graph. Skills with fewer outgoing prerequisite edges (e.g., Perceptual, Motor, and Physical Development; Digital Literacy) are less connected to later stages and are observed to be more vulnerable to forgetting in continual training.

Table 8: All results for CSQA test set. The column represents each stage on which a model is being evaluated.

Test type	CSQA (rating out of 5)				
Stages	0	1	2	3	4
M_0	3.89	2.85	2.52	2.33	1.95
M_1	3.63	3.35	2.92	2.75	2.39
M_2	3.53	3.25	3.15	2.87	2.53
M_3	3.51	3.22	3.03	3.10	2.61
M_4	3.29	3.13	3.00	2.93	2.89
M_{012}	3.97	3.48	3.21	2.96	2.61
M_{01}	3.93	3.37	2.93	2.76	2.40
M_{0-1}	3.87	3.55	3.09	2.91	2.51
M_{0123}	3.97	3.47	3.21	3.08	2.65
M_{0-1-2}	3.83	3.47	3.31	3.03	2.66
M_{01234}	3.97	3.49	3.24	3.13	2.88
$M_{0-1-2-3}$	3.83	3.48	3.24	3.26	2.76
$M_{0-1-2-3-4}$	3.65	3.41	3.23	3.17	3.05

```
- a_label and a_id
- b_label and b_id

These represent two distinct skill indicators. You must determine whether one is a
    prerequisite for the other.

Instructions:
- A skill X is a prerequisite for skill Y if Y logically requires understanding or
    demonstrating X beforehand.
- Compare the meaning of a_label and b_label to determine if:
  - A depends on B  edge from b_id to a_id
  - B depends on A  edge from a_id to b_id
  - No clear dependency  no edge

Output format:
Return a JSON object like:

‘‘‘json
{{
  "edge": true or false,
  "from": "source_id" or "NA",
  "to": "target_id" or "NA",
  "reason": "Brief explanation of the dependency or lack thereof"
}}
‘‘‘

- If there is a dependency, set edge: true, from as the prerequisite’s ID, and to as
     the dependent’s ID.
- If there is no clear prerequisite relationship, set edge: false and "from": "NA",
    "to": "NA" with a brief justification in reason.

Only base your answer on the textual meaning of the labels, and only report direct
    dependencies (not transitive or indirect ones).
```

User prompt for Edge prediction

```
Given the following skill indicators:
- a_label: {label_1}
- a_id: {id_1}
- b_label: {label_2}
- b_id: {id_2}
```

````
Determine the dependency relationship and output the JSON:

```json
{{
  "edge": true or false,
  "from": "source_id" or "NA",
  "to": "target_id" or "NA",
  "reason": "Brief explanation of the dependency or lack thereof"
}}
```
````

E.2 Edge weight prediction

System prompt:

```
You are an expert in child development, skill acquisition, and cognitive science.
    Your task is to rate the strength of a prerequisite relationship between two
    skill indicators. Each input includes:
- from_label and to_label: the skill indicators (already determined to be in a
    prerequisite relationship, where from_label is a prerequisite for to_label)
- Additional metadata: age groups, subskills, goals, developmental stages, and a
    rationale for why the edge exists.

Instructions:
Rate the dependency strength on a scale from 1 to 5, where:
- 1 = Very weak dependency (minimal or contextual support, can often be developed
    independently)
- 2 = Weak dependency (some support role, but not always required)
- 3 = Moderate dependency (often occurs first, but not strictly necessary)
- 4 = Strong dependency (usually needed before progressing)
- 5 = Very strong dependency (essential foundational step for the next)

Your response should consider:
1. The specific behaviors or understandings described in the two indicators.
2. Whether the earlier skill is conceptually or procedurally required to perform the
     later one.
3. The closeness of developmental stages and subskills.

Output Format:
Return your decision as a JSON object:
```json
{{
  "weight": [an integer from 1 to 5],
  "reason": "[a brief explanation of why this weight reflects the strength of the
      dependency]"
}}
```
```

User prompt:

```
Given the following information about a prerequisite relationship between two skill
    indicators:

- from_label: {from_label}
- from_id: {from_id}
    - age group: {from_age_group}
    - skill: {from_skill}
    - subskill: {from_subskill}
    - goal: {from_goal}
    - stage: {from_stage}

-------------------------

- to_label: {to_label}
```

```
- to_id: {to_id}
    - age group: {to_age_group}
    - skill: {to_skill}
    - subskill: {to_subskill}
    - goal: {to_goal}
    - stage: {to_stage}

This relationship has already been labeled as a prerequisite edge (from_id  to_id).

Rationale for this dependency:
"{reason}"

Rate the strength of this dependency on a scale from 1 to 5.

Output a JSON object:
‘‘‘json
{{
  "weight": [an integer from 1 to 5],
  "reason": "Brief explanation of why this weight reflects the strength of the
      dependency"
}}
‘‘‘
```

E.3 Templates

System prompt for generating templates for IR data:

```
You are an expert in child development, skill acquisition, curriculum design, and
    language model pretraining. Your task is to identify developmentally
    appropriate and general **non-instructional text types** for synthetic
    pretraining of a language model.

Each input includes:
- indicator: a natural language description of the learning objective or task
- age_group: developmental age (e.g., 05, 511, 1114)
- skill: broad academic or developmental domain (e.g., Mathematics, English,
    Scientific Reasoning)
- subskill: a specific subdomain or area of focus (e.g., Listening, Measurement,
    Problem-solving)
- goal: the purpose or nature of the learning (e.g., Application, Reflection,
    Evaluation)
- stage: the curriculum stage (0 to 9, loosely corresponding to increasing age and
    complexity)

Instructions:
Return a list of **general non-instructional text types** that:
- Are suitable for the learner's developmental stage
- Reflect naturalistic or structured formats that don't rely on explicit
    instructionresponse pairs
- Can be used as abstract templates to generate content across many topics
- Are defined at a high level of abstraction (e.g., "peer dialogue", "narrative
    description", "cause-effect explanation")

**CRITICALLY IMPORTANT**:
- Provide format categories, NOT specific content or scenarios
- Text types should be 2-5 words that describe a general format, not complete
    sentences
- Each text type should be usable with ANY topic relevant to the age/skill
    combination

**Examples of appropriate non-instructional text types**:
- "Narrative story with characters"
- "Peer conversation transcript"
- "Process description passage"
```

```
- "Personal reflection monologue"

**Examples of inappropriate text types** (too specific):
- "Story about a child going to the zoo"
- "Conversation between friends about toys"
- "Description of a butterfly's life cycle"

Output Format:
Return your result as a JSON object with the following structure:

‘‘‘json
{{
  "text_types": ["...", "...", "..."]
}}
‘‘‘

Ensure the list is:
- 1520 items long
- Abstract enough to work across many topics
- Varied across narration, description, interaction, emotion, reasoning
- Appropriate in complexity for the given age group and learning goal

Only output the JSON object.
```

User prompt for generating templates for IR data:

```
Given the following information about a learning objective, return a list of general
    , reusable non-instructional text formats that can serve as templates for
    synthetic training data:

- indicator: {indicator}
- age_group: {age_group}
- skill: {skill}
- subskill: {subskill}
- goal: {goal}
- stage: {stage}

IMPORTANT: Provide ABSTRACT FORMAT CATEGORIES (2-5 words each), not specific content
     or scenarios.

Examples of good non-instructional formats:
- "Peer dialogue transcript"
- "Sequential process description"
- "Character-driven narrative"
- "Emotional experience monologue"

Examples of unsuitable formats (too specific):
- "Conversation between friends about toys"
- "Description of a butterfly's life cycle"
- "Story about going to the beach"

Ensure your list contains:
- 15 to 20 developmentally appropriate text formats
- General templates that can be combined with ANY relevant topic
- Varied format types that don't rely on explicit instruction-response pairs

Return only a JSON object in the following format:

‘‘‘json
{{
  "text_types": ["...", "...", "..."]
}}
‘‘‘
```

System prompt for generating templates for Context data:

```
You are an expert in child development, skill acquisition, curriculum design, and
    language model pretraining. Your task is to identify developmentally
    appropriate and general **instruction-response text types** for synthetic
    pretraining of a language model.

Each input includes:
- indicator: a natural language description of the learning objective or task
- age_group: developmental age (e.g., 05, 511, 1114)
- skill: broad academic or developmental domain (e.g., Mathematics, English,
    Scientific Reasoning)
- subskill: a specific subdomain or area of focus (e.g., Listening, Measurement,
    Problem-solving)
- goal: the purpose or nature of the learning (e.g., Application, Reflection,
    Evaluation)
- stage: the curriculum stage (0 to 9, loosely corresponding to increasing age and
    complexity)

Instructions:
Return a list of **general instruction-response style text types** that:
- Are suitable for the learner's developmental stage
- Can be used in instruction tuning and task-based language modeling
- Involve a clearly defined instruction format that can be applied across many
    topics
- Are defined at a high level of abstraction (e.g., "explain why X occurs", "compare
     and contrast X and Y")

**CRITICALLY IMPORTANT**:
- Provide abstract instruction formats, NOT specific prompts or questions
- Text types should be 2-5 words describing a general instruction format
- Each text type should be usable with ANY topic relevant to the age/skill
    combination

**Examples of appropriate instruction-response text types**:
- "Compare and contrast analysis"
- "Explain why reasoning"
- "Step-by-step instruction"
- "Open-ended reflection prompt"

**Examples of inappropriate text types** (too specific):
- "Explain why plants need water"
- "Compare dogs and cats"
- "Describe your favorite toy"

Output Format:
Return your result as a JSON object with the following structure:

‘‘‘json
{{
  "text_types": ["...", "...", "..."]
}}
‘‘‘

Ensure the list is:
- 1520 items long
- Abstract enough to work across many topics
- Varied across explanation, reasoning, reflection, comparison, instruction,
    imagination
- Appropriate in complexity for the given age group and learning goal

Only output the JSON object.
```

User prompt for generating templates for Context data:

```
Given the following information about a learning objective, return a list of general
    , reusable instruction-response text formats that can serve as templates for
    synthetic training data:

- indicator: {indicator}
- age_group: {age_group}
- skill: {skill}
- subskill: {subskill}
- goal: {goal}
- stage: {stage}

IMPORTANT: Provide ABSTRACT INSTRUCTION FORMATS (2-5 words each), not specific
    questions or prompts.

Examples of good instruction formats:
- "Compare and contrast analysis"
- "Explain why reasoning"
- "Problem-solving walkthrough"
- "Open-ended reflection prompt"

Examples of unsuitable formats (too specific):
- "Explain why plants need water"
- "Compare dogs and cats"
- "Solve this math problem"

Ensure your list contains:
- 15 to 20 developmentally appropriate instruction formats
- General templates that can be combined with ANY relevant topic
- Varied instruction types that address different cognitive processes

Return only a JSON object in the following format:

‘‘‘json
{{
  "text_types": ["...", "...", "..."]
}}
‘‘‘
```

E.4 Context

System prompt for generating context data:

```
You are an AI model generating training data to help language models simulate human
    developmental skills at various stages from early childhood through early
    adolescence.

Your task is to create engaging, developmentally appropriate texts based on provided
     developmental indicators, skills, and a tuple of word and its part of speech.

Strictly follow these guidelines:

1. **Developmental Appropriateness:**
   - Stage 0 (Age 5): Use simple sentences, concrete concepts, familiar experiences,
       present tense focus
   - Stages 1-3 (Ages 6-8): Introduce basic past/future concepts, simple cause-
       effect, familiar settings
   - Stages 4-6 (Ages 9-11): Include more complex reasoning, abstract thinking,
       varied sentence structures
   - Stages 7-9 (Ages 12-14): Incorporate hypothetical scenarios, multiple
       perspectives, sophisticated vocabulary

2. **Context Generation:**
```

```
   - Use the provided word and its part of speech to create a meaningful,
       developmentally appropriate topic
   - **Ensure the selected word and expanded topic fit the required Text Type
       Template (context_template)**
   - Expand the selected word into a more detailed, skill-aligned topic that
       resonates with the target age group
   - Generate a rich, complete, and engaging text matching the provided context
       template
   - The generated text must be **between 250 and 500 words regardless of
       developmental stage**
   - The text must clearly align with the skill, subskill, goal, and indicator
   - The selected word does not need to explicitly appear in the final text

3. **Writing Style by Stage:**
   - **Early Stages (0-3):** Simple vocabulary, short to medium sentences, concrete
       experiences, repetitive patterns for reinforcement
   - **Middle Stages (4-6):** More varied vocabulary, complex sentences,
       introduction of abstract concepts, problem-solving scenarios
   - **Later Stages (7-9):** Sophisticated vocabulary, complex sentence structures,
       abstract reasoning, multiple viewpoints

4. **Content Enrichment:**
   - Include age-appropriate actions, feelings, interactions, and sensory details
   - Incorporate social situations relevant to the developmental stage
   - Use scenarios that promote the specific skill being targeted
   - Avoid overly abstract or culturally specific references unless appropriate for
       the age group

5. **Output Format:** Strictly return the output in the following JSON structure:
‘‘‘json
{{
    "expanded_topic": "<expanded topic>",
    "generated_text": "<generated text between 250 and 500 words>"
}}
‘‘‘
Only output the JSON. No additional commentary.
```

User prompt for generating context data:

```
Generate a rich and engaging context text based on the following input:

- ID: {id}
- Indicator: {indicator}
- Skill: {skill}
- Sub-skill: {subskill}
- Goal: {goal}
- Age Group: {age_group}
- Stage: {stage}
- Text Type Template: {context_template}
- (Word, Part of speech): {word_list}

Instructions:
- Consider the developmental stage ({stage}) and age group ({age_group}) when
    crafting vocabulary, sentence complexity, and content themes
- Expand the selected word into a skill-relevant topic **that fits the Text Type
    Template**
- Generate a detailed text of **250500 words** following the context template
- Enrich the text with developmentally appropriate actions, emotions, and
    interactions
- Ensure the content promotes the specific skill and subskill being targeted

Output strictly in this format:
‘‘‘json
{{
    "expanded_topic": "<expanded topic>",
```

```
    "generated_text": "<generated text between 250 and 500 words>"
}}
```

E.5 CQA

System prompt for generating CQA data:

```
You are an AI model generating training data to help language models simulate human
    reading comprehension skills at various stages from early childhood through
    early adolescence.

Your task is to create 5 developmentally appropriate question-answer pairs based on
    a provided text, ensuring all questions test understanding of the given
    paragraph and can be answered directly from the text.

Strictly follow these guidelines:

1. **Developmental Appropriateness by Stage:**
   - Stage 0 (Age 5): Simple "what/who/where" questions, literal comprehension,
       single-step reasoning
   - Stages 1-3 (Ages 6-8): Basic "why/how" questions, simple cause-effect, sequence
        understanding, character feelings
   - Stages 4-6 (Ages 9-11): Inference questions, comparing/contrasting, predicting
       outcomes, understanding motivations
   - Stages 7-9 (Ages 12-14): Complex analysis, multiple perspectives, abstract
       concepts, theme identification

2. **Question Creation Standards:**
   - **All answers must be directly supported by information in the provided text**
   - No questions requiring outside knowledge or information not present in the text
   - Questions should test different types of comprehension appropriate to the
       developmental stage
   - Vary question types to assess different reading skills (literal, inferential,
       evaluative)
   - Use vocabulary and sentence complexity appropriate to the age group
   - Ensure questions are engaging and relevant to the child's interests and
       experiences

3. **Question Types by Stage:**
   - **Early Stages (0-3):** Literal recall, identifying main characters/objects,
       simple sequence, basic emotions
   - **Middle Stages (4-6):** Cause-effect relationships, character motivations,
       comparing details, simple predictions
   - **Later Stages (7-9):** Drawing conclusions, analyzing relationships,
       evaluating actions, understanding themes

4. **Answer Generation:**
   - Create authentic child responses that demonstrate comprehension at the target
       developmental stage
   - Use vocabulary and sentence structures appropriate to the age group
   - Include natural speech patterns and expressions typical of the developmental
       stage
   - Ensure answers are complete but not overly elaborate for the age group
   - Answers should sound conversational and natural, not textbook-like

5. **Content Guidelines:**
   - **Purely verbal exchanges** - no references to physical gestures or non-verbal
       actions
   - No formatting (bold, italics, markdown)
   - Questions should flow naturally and cover different aspects of the text
   - Ensure logical progression from simpler to more complex questions when
       appropriate
```

```
   - Include a mix of question types (factual, inferential, personal connection when
       text-supported)

6. **Quality Standards:**
   - Every question must be answerable using only information provided in the text
   - Questions should test genuine comprehension, not just memory of isolated facts
   - Avoid questions with obvious or trivial answers
   - Ensure questions are meaningful and help assess understanding of key text
      elements
   - Create questions that feel natural in an educational setting

7. **Output Format:** Strictly return the output in the following JSON structure:
'''json
{{
    "question_answer_pairs": [
        {{
            "question": "<question 1>",
            "answer": "<answer 1>"
        }},
        {{
            "question": "<question 2>",
            "answer": "<answer 2>"
        }},
        {{
            "question": "<question 3>",
            "answer": "<answer 3>"
        }},
        {{
            "question": "<question 4>",
            "answer": "<answer 4>"
        }},
        {{
            "question": "<question 5>",
            "answer": "<answer 5>"
        }}
    ]
}}
'''
Only output the JSON. No additional commentary or explanations.
```

User prompt for generating CQA data:

```
Generate 5 developmentally appropriate reading comprehension question-answer pairs
    based on the following input:

- Text: {output}
- Age Group: {age_group}
- Stage: {stage}

Instructions:
- Consider the developmental stage ({stage}) and age group ({age_group}) when
    crafting question complexity and answer expectations
- Create questions that test different types of comprehension appropriate to the
    developmental level
- **Ensure all questions can be answered directly from the provided text**
- Generate authentic child responses that demonstrate comprehension at the target
    stage
- Use vocabulary and sentence structures appropriate to the age group
- Create a mix of question types that genuinely assess understanding of the text

Output strictly in this format:
'''json
{{
    "question_answer_pairs": [
        {{
```

```
            "question": "<question 1>",
            "answer": "<answer 1>"
        }},
        {{
            "question": "<question 2>",
            "answer": "<answer 2>"
        }},
        {{
            "question": "<question 3>",
            "answer": "<answer 3>"
        }},
        {{
            "question": "<question 4>",
            "answer": "<answer 4>"
        }},
        {{
            "question": "<question 5>",
            "answer": "<answer 5>"
        }}
    ]
}}
```

E.6 CSQA

System prompt for generating CSQA data:

```
You are an AI model generating training data to help language models simulate human
    developmental skills at various stages from early childhood through early
    adolescence.

Your task is to create 3 skill-based instruction-response pairs between an educator
    and a child that use a provided text as context to test specific developmental
    skills, rather than simple reading comprehension.

Strictly follow these guidelines:

1. **Developmental Appropriateness by Stage:**
   - Stage 0 (Age 5): Simple vocabulary, short sentences, concrete thinking, present
       -focused, immediate experiences
   - Stages 1-3 (Ages 6-8): Basic past/future concepts, simple reasoning, familiar
       contexts, beginning abstract thought
   - Stages 4-6 (Ages 9-11): Complex reasoning, abstract thinking, varied sentence
       structures, hypothetical scenarios
   - Stages 7-9 (Ages 12-14): Sophisticated vocabulary, multiple perspectives,
       advanced abstract reasoning, nuanced responses

2. **Skill-Based Instruction Creation:**
   - **Use the provided text as context, not as the primary focus**
   - Create instructions that test the specific skill, subskill, goal, and indicator
       provided
   - Instructions should prompt the child to demonstrate the target skill using
       elements from the text
   - Avoid simple recall questions - focus on skill application, analysis, synthesis
       , or evaluation
   - Vary instruction starters - avoid overusing "Imagine..." or "Tell me about..."
   - Include necessary context within the instruction if recall is required
   - Use developmentally appropriate language and concepts for the target stage
   - Make instructions engaging and thought-provoking for the age group

3. **Response Generation:**
   - Create authentic child responses that clearly demonstrate the target indicator
   - Use vocabulary, sentence complexity, and reasoning appropriate to the
       developmental stage
```

```
   - Include natural speech patterns and expressions typical of the age group
   - Ensure responses show genuine skill application, not just text recall
   - Responses should be verifiable through either:
     * Information provided in the instruction or text
     * Common world knowledge appropriate for the child's developmental level
     * Typical personal experiences for that age group
   - Avoid arbitrary claims or purely imaginative details unless the skill
       explicitly encourages creativity

4. **Context Integration:**
   - Use the provided text as a springboard for skill demonstration
   - Connect text elements to real-world applications of the skill
   - Encourage children to apply their skills to analyze, extend, or relate to the
       text content
   - Ensure the skill being tested is meaningfully connected to the text context

5. **Content Guidelines:**
   - **Purely verbal exchanges** - no references to physical objects, gestures, or
       non-verbal actions
   - No formatting (bold, italics, markdown)
   - Instructions should feel natural and appropriate for educational settings
   - Responses should sound natural and spontaneous, not rehearsed
   - Include appropriate emotional expressions and personal connections when
       relevant
   - Ensure logical consistency between instruction and response
   - Focus on the skill demonstration rather than text comprehension

6. **Quality Standards:**
   - The exchange must demonstrate clear alignment with the skill, subskill, goal,
       and indicator
   - Each instruction must clearly target the specific developmental parameters
       provided
   - Instructions should be distinct from each other, testing different aspects of
       the same skill
   - Both instruction and response should feel authentic to a real classroom or
       learning interaction
   - Responses must demonstrate clear mastery or development of the target skill
   - The text should serve as meaningful context, not just background information
   - Avoid overly abstract concepts for younger stages or overly simple concepts for
        older stages
   - Ensure developmental appropriateness in both challenge level and expectations

7. **Output Format:** Strictly return the output in the following JSON structure:
'''json
{{
    "skill_based_pairs": [
        {{
            "instruction": "<instruction 1>",
            "response": "<response 1>"
        }},
        {{
            "instruction": "<instruction 2>",
            "response": "<response 2>"
        }},
        {{
            "instruction": "<instruction 3>",
            "response": "<response 3>"
        }}
    ]
}}
'''
Only output the JSON. No additional commentary or explanations.
```

User prompt for generating CSQA data:

```
Generate 3 developmentally appropriate skill-based instruction-response pairs based
    on the following input:

- Text: {output}
- Age Group: {age_group}
- Stage: {stage}
- Skill: {skill}
- Sub-skill: {subskill}
- Goal: {goal}
- Indicator: {indicator}

Instructions:
- Consider the developmental stage ({stage}) and age group ({age_group}) when
    crafting instruction complexity and response expectations
- Use the provided text as context to create instructions that test the specific
    skill ({skill}) and subskill ({subskill})
- Create instructions that elicit demonstration of the goal ({goal}) and indicator
    ({indicator})
- **Focus on skill application and demonstration, not text comprehension**
- Generate authentic child responses that show clear mastery of the target skill at
    the developmental stage
- Use vocabulary and sentence structures appropriate to the age group
- Create 3 distinct instructions that test different aspects of the same skill

Output strictly in this format:
‘‘‘json
{{
    "skill_based_pairs": [
        {{
            "instruction": "<instruction 1>",
            "response": "<response 1>"
        }},
        {{
            "instruction": "<instruction 2>",
            "response": "<response 2>"
        }},
        {{
            "instruction": "<instruction 3>",
            "response": "<response 3>"
        }}
    ]
}}
‘‘‘
```

E.7 IR

System prompt for generating IR data:

```
You are an AI model generating training data to help language models simulate human
    developmental skills at various stages from early childhood through early
    adolescence.

Your task is to create realistic instruction-response pairs between an educator and
    a child, based on developmental indicators, skills, and a tuple of word and its
     part of speech.

Strictly follow these guidelines:

1. **Developmental Appropriateness by Stage:**
   - Stage 0 (Age 5): Simple vocabulary, short sentences, concrete thinking, present
       -focused, immediate experiences
   - Stages 1-3 (Ages 6-8): Basic past/future concepts, simple reasoning, familiar
       contexts, beginning abstract thought
```

```
   - Stages 4-6 (Ages 9-11): Complex reasoning, abstract thinking, varied sentence
       structures, hypothetical scenarios
   - Stages 7-9 (Ages 12-14): Sophisticated vocabulary, multiple perspectives,
       advanced abstract reasoning, nuanced responses

2. **Instruction Creation:**
   - Use the provided word and its part of speech to meaningfully inspire the
       interaction topic
   - **Ensure the topic aligns with the Text Type Template (instruct_template)**
   - Craft prompts that naturally elicit demonstration of the specific indicator and
        skill
   - Vary instruction starters - avoid overusing "Imagine..." or "Tell me about..."
   - Include necessary context within the instruction if recall is required
   - Use developmentally appropriate language and concepts for the target stage
   - Make instructions engaging and thought-provoking for the age group

3. **Response Generation:**
   - Create authentic child responses that clearly demonstrate the target indicator
   - Use vocabulary, sentence complexity, and reasoning appropriate to the
       developmental stage
   - Include natural speech patterns and expressions typical of the age group
   - Ensure responses are verifiable through either:
     * Information provided in the instruction
     * Common world knowledge appropriate for the child's developmental level
     * Typical personal experiences for that age group
   - Avoid arbitrary claims or purely imaginative details unless storytelling is
       explicitly encouraged

4. **Content Guidelines:**
   - **Purely verbal exchanges** - no references to physical objects, gestures, or
       non-verbal actions
   - No formatting (bold, italics, markdown)
   - Responses should sound natural and spontaneous, not rehearsed
   - Include appropriate emotional expressions and personal connections when
       relevant
   - Ensure logical consistency between instruction and response

5. **Quality Standards:**
   - The exchange must demonstrate clear alignment with the skill, subskill, goal,
       and indicator
   - Both instruction and response should feel authentic to a real classroom or
       learning interaction
   - Avoid overly abstract concepts for younger stages or overly simple concepts for
        older stages
   - Ensure the selected word meaningfully influences the dialogue topic

6. **Output Format:** Strictly return the output in the following JSON structure:
'''json
{{
    "instruction": "<instruction>",
    "response": "<response>"
}}
'''
Only output the JSON. No additional commentary or explanations.
```

User prompt for generating IR data:

```
Generate a developmentally appropriate instruction-response pair based on the
    following input:

- ID: {id}
- Indicator: {indicator}
- Skill: {skill}
- Sub-skill: {subskill}
- Goal: {goal}
```

```
- Age Group: {age_group}
- Stage: {stage}
- Text Type Template: {instruct_template}
- (Word, Part of speech): {word_list}

Instructions:
- Consider the developmental stage ({stage}) and age group ({age_group}) when
    crafting language complexity and content themes
- Use the selected word to meaningfully inspire the interaction topic **that fits
    the Text Type Template**
- Create an engaging instruction that naturally elicits demonstration of the target
    indicator
- Generate an authentic child response that clearly shows mastery of the skill and
    subskill
- Ensure the exchange feels natural and appropriate for a real educational
    interaction

Output strictly in this format:
‘‘‘json
{{
    "instruction": "<instruction>",
    "response": "<response>"
}}
‘‘‘
```

E.8 Evaluating CQA

System prompt for evaluating trained model's response for questions from CQA:

```
You are a developmental expert evaluating how well a child's answer to a reading
    comprehension question reflects appropriate understanding and reasoning for a
    specific developmental stage.

You will receive:
- The original **context** paragraph
- A **question** based on the context
- The child's **answer** to the question
- The child's **developmental stage** (09)
- The child's **age group** (e.g., '05', '511', '1114')

Your job is to:
1. **Rate the childs answer on a scale from 1 to 5**, using the following criteria:
   - **5  Excellent:** Fully correct, precise, and well-formed for the stage. Shows
       strong comprehension and reasoning.
   - **4  Strong:** Mostly correct and appropriate; may have minor phrasing issues
       or slight gaps in reasoning.
   - **3  Adequate:** Understands the gist but may be vague, partially incorrect, or
        simplistic for the stage.
   - **2  Limited:** Misunderstands part of the question or context; reasoning is
       weak or off-track.
   - **1  Inadequate:** Confused, incorrect, or clearly not appropriate for the
       stage.

2. **Consider developmental expectations** for language and reasoning:
   - **Stage 0 (Age 5):** Very basic phrases, literal recall, present-focused
       answers
   - **Stages 13 (Ages 68):** Simple reasoning, sequencing, basic cause-effect,
       clear answers
   - **Stages 46 (Ages 911):** Logical inference, comparative language, clear
       justification
   - **Stages 79 (Ages 1214):** Abstract reasoning, complex ideas, nuanced
       explanations

3. **Evaluate:**
```

```
   - Does the childs answer meaningfully address the question using the provided
       context?
   - Is the reasoning and language appropriate for the stage?
   - Does it reflect comprehension of the text and question?

4. **Output Format:**
Only return the following dictionary:
```json
{{
    "rating": <integer from 1 to 5>,
    "explanation": "<23 sentence rationale>"
}}
```
Do not add any other text or formatting. Only return the JSON object.
```

User prompt for evaluating trained model's response for questions from CQA:

```
Evaluate the childs answer to a reading comprehension question. Consider the context
     and the developmental stage.

Context:
{context}

Question:
{question}

Answer:
{answer}

Stage: {stage}
Age group: {age_group}
Index: {q_index}

**Output Format:**
```json
{{
    "rating": <integer from 1 to 5>,
    "explanation": "<23 sentence rationale>"
}}
```
```

E.9 Evaluating CSQA

System prompt for evaluating trained model's response for questions from CSQA:

```
You are a developmental expert evaluating how well a child's response demonstrates a
     specific developmental skill at a given stage, using a provided instruction
    and background text.

You will receive:
- A short **text** (used as context for the instruction)
- A **skill-based instruction** given to the child
- The childs **response**
- The childs **developmental stage** (09)
- The childs **age group** (e.g., '05', '511', '1114')
- The **target skill**, **subskill**, **goal**, and **indicator** that the
    instruction was designed to assess

Your job is to:
1. **Rate the child's response on a scale from 1 to 5**, using these criteria:
   - **5  Excellent:** Fully demonstrates the targeted skill/indicator with clarity
       and developmental appropriateness. Strong reasoning, appropriate expression,
       and alignment with instruction.
```

```
   - **4  Strong:** Mostly appropriate and well-formed. Some minor gaps in
       completeness, precision, or phrasing, but shows the intended skill.
   - **3  Adequate:** Response attempts the skill but may be vague, simplistic, or
       only partially aligned with the goal/indicator.
   - **2  Limited:** Weak or unclear demonstration of the skill. Response is
       partially off-track, underdeveloped, or barely relevant.
   - **1  Inadequate:** Fails to demonstrate the intended skill. Response is
       irrelevant, confusing, or clearly inappropriate for the stage.

2. **Use stage-specific developmental expectations**:
   - **Stage 0 (Age 5):** Short, concrete, present-focused responses with simple
       vocabulary
   - **Stages 13 (Ages 68):** Clear expression of ideas, simple cause-effect,
       emotional awareness, basic reasoning
   - **Stages 46 (Ages 911):** Logical structure, hypothetical thinking, connections
        to personal experience, comparisons
   - **Stages 79 (Ages 1214):** Advanced abstraction, multiple perspectives,
       justification, nuanced expression

3. **Evaluate:**
   - Does the childs response meaningfully follow the instruction?
   - Does it demonstrate the **targeted skill and indicator**?
   - Is the language, reasoning, and expression developmentally appropriate for the
       stage?
   - Is the response authentic and logically consistent with the instruction and the
        context text?

4. **Output Format:**
Return only the following dictionary:
‘‘‘json
{{
    "rating": <integer from 1 to 5>,
    "explanation": "<23 sentence rationale>"
}}
‘‘‘
Do not add any other text or formatting. Only return the JSON object.
```

User prompt for evaluating trained model's response for questions from CSQA:

```
Evaluate the child's response to a skill-based instruction using the provided text
    and developmental context. Focus on how well the response demonstrates the
    intended skill.

Context:
{context}

Instruction:
{instruction}

Response:
{response}

Stage: {stage}
Age group: {age_group}
Skill: {skill}
Subskill: {subskill}
Goal: {goal}
Indicator: {indicator}
Index: {q_index}

Output format:
‘‘‘json
{{
    "rating": <integer from 1 to 5>,
    "explanation": "<23 sentence rationale>"
```

```
}}
```
```

### E.10 Evaluating IR

System prompt for evaluating trained model's response for questions from IR:

```
You are a developmental expert rating how well a child's response to a prompt
    demonstrates age-appropriate reasoning and language for a given developmental
    stage.

You will receive:
- An **instruction** given to the child
- The child's **response**
- The child's **developmental stage** (09)
- The child's **age group** (e.g., '05', '511', '1114')

Your job is to:
1. **Rate the response on a scale from 1 to 5**, using the following criteria:
   - **5  Excellent:** The response fully addresses the instruction with clear,
       developmentally appropriate reasoning and language. It meets expectations for
        the stage with no major issues.
   - **4  Strong:** Mostly appropriate and coherent; minor gaps in clarity, depth,
       or completeness.
   - **3  Adequate:** A reasonable attempt that partially addresses the instruction;
        may be vague, brief, or contain small misunderstandings.
   - **2  Limited:** Weak or underdeveloped response; minimal reasoning or limited
       relevance to the instruction.
   - **1  Inadequate:** Response is off-topic, confusing, or clearly inappropriate
       for the stage.

2. **Use stage-specific developmental expectations**:
   - **Stage 0 (Age 5):** Very simple sentences, concrete ideas, focused on here and
        now
   - **Stages 13 (Ages 68):** Simple reasoning, some past/future thinking, familiar
       examples
   - **Stages 46 (Ages 911):** Logical structure, comparisons, abstract or
       hypothetical reasoning
   - **Stages 79 (Ages 1214):** Nuanced reasoning, multi-step thinking, advanced
       vocabulary

3. **Evaluate:**
   - Does the childs response meaningfully address the instruction?
   - Is the language and reasoning developmentally appropriate for the stage?
   - Is the response authentic and logically consistent?

4. **Output Format:**
Only return the following dictionary:
'''json
{{
    "rating": <integer from 1 to 5>,
    "explanation": "<23 sentence rationale>"
}}
'''
Do not add any other text or formatting. Only return the JSON object.
```

User prompt for evaluating trained model's response for questions from IR:

```
Evaluate the child's response to the instruction below based on the developmental
    stage and age group. Return a numerical rating (15) and a short explanation.

Instruction: {instruction}
Response: {response}
Stage: {stage}
```
```

````
Age group: {age_group}
Index: {q_index}

**Output Format:**
Only return the following dictionary:
```json
{{
    "rating": <integer from 1 to 5>,
    "explanation": "<23 sentence rationale>"
}}
```
````