Exploring Feedforward Networks as Routing Mechanisms in Composition of Experts Models

Sarah Tsai

University of California, Berkeley

sarahwtsai@berkeley.edu

Abstract

Composition of Experts (CoE) models provide an efficient alternative to large monolithic language models by routing inputs to one of several smaller, specialized experts, thereby reducing inference costs. We present Qwen-CoE, a CoE architecture that uses a feedforward network (FFN) to route user inputs to one of three Qwen2.5 models. This work evaluates the viability of FFNs as an alternative to conventional CoE routers and investigates their ability to generalize beyond their training distribution, including when trained on partially synthetic data.

1 Introduction

In recent years, large language models (LLMs) have achieved remarkable success across a wide range of natural language processing tasks, including translation, information extraction, and sentiment analysis.[1] The prevailing trend in model development has favored ever-increasing scale, with state-of-the-art models like GPT-4 and DeepSeek-R1 demonstrating that larger architectures tend to achieve superior performance.[2–4] While smaller, more efficient LLMs have been introduced as an alternative, they often struggle to match the broad capabilities of their larger counterparts.[5]

Despite this, fine-tuning has emerged as a powerful approach to optimize smaller models for specific tasks. Research has shown that domain-specific fine-tuned models can outperform larger general-purpose LLMs in specialized areas such as scientific reasoning and code generation.[6,7] This has led to the development of Composition of Experts (CoE) models, a flexible framework that allows for the integration of multiple fine-tuned models. CoE architectures use a routing mechanism to choose one out of many model experts to formulate a response. Existing CoE models, such as Samba-CoE, have demonstrated that this approach can achieve results comparable to larger models at a fraction of the inference cost by dynamically selecting the most relevant expert for each query.[8,9]

CoE models typically rely on a routing mechanism that is trained using a two-step process.[9] First, the router learns to assign user inputs to different categories using a labeled dataset and a multi-class text classifier. Then the router is trained to learn a mapping between these categories and the most suitable expert models, allowing for adaptive selection of experts based on the nature of the task. Unlike Mixture of Experts (MoE) models such as Mixtral 8x7B, which activate multiple experts within a single transformer layer during inference, CoE architectures operate at the model level, selecting entire fine-tuned models based on the query rather than mixing outputs from multiple experts within a unified model.[10,11]

This work introduces Qwen-CoE, a simplified interpretation of the CoE architecture that integrates three Qwen2.5 models and replaces the conventional routing mechanism with a feedforward network (FFN) (Figure 1). The FFN will determine which LLM should be used for inference based on an embedding of the user input generated by Multilingual-E5-large-instruct.[12] The primary objective of this study is to assess the viability of FFN-based routing as an alternative to more complex, traditional routing mechanisms. In the absence of an existing Qwen-CoE implementation with a conventional router for direct comparison, we adopt the best-performing of the three Qwen models as a baseline since the only difference in inference cost would be the computational overhead from the router and embedding model. Using the Massive Multitask Language Understanding (MMLU) benchmark for evaluation, we aim to provide insights into the effectiveness and limitations of simplified routing mechanisms within CoE architectures.[13] Additionally, we investigate the impact of

training data quality and its alignment with the model's end use case on an FFN's ability to accurately select experts. Our experiments involve training three distinct FFNs, each on a different dataset: one incorporating synthetic data, one excluding synthetic data, and one utilizing the MMLU training subset.

2 Methods

Qwen-Coe is composed of three open-source Qwen models: Qwen2.5-1.5B-Instruct, Qwen2.5-Math-1.5B-Instruct, and Qwen2.5-Coder-1.5B-Instruct.[14–16] Moving forward, these models will be referred to as BaseQwen, MathQwen, and CodeQwen, respectively. The training pipeline consists of three main components: (1) compiling a diverse training set (2) assigning labels to the training set based on each Qwen model's responses, and (3) training the routers. We use the MMLU benchmark, a collection of multiple-choice questions covering 57 subjects, to evaluate the trained routers and all three Qwen models. The routers and the datasets they were trained on are listed in Table 1 below.

Router Name	Dataset	Description
Router_Combine	CombineQA	Human-curated and synthetic
Router_NoSynth	NoSynthQA (CombineQA excluding ComSciQA)	Human-curated only
Router_MMLU	MMLU_train	MMLU subset

Table 1: Routers and their associated training datasets, along with a description of each dataset.

2.1 Datasets and Data Generation

CombineQA is a composite dataset containing multiple-choice Q&A questions drawn from five distinct sources:

- ARC AI2 Reasoning Challenge:[17] 7,761 grade-school science questions
- OpenBookQA:[18] 11,914 science and reasoning questions
- MathQA:[19] 30,946 math word problems
- CommonsenseQA:[20] 8,799 general reasoning questions
- ComSciQA: 33,540 computer science questions generated by Llama-3.3-70B-Instruct[21]

With the exception of ComSciQA, which was generated using Llama-3.3-70B-Instruct specifically for this study (details in Section 2.1.1), the remaining four multiple-choice datasets were manually compiled. These human-curated datasets are presumed to be of higher quality due to a greater likelihood of factual correctness. To assess coverage and alignment of our selected datasets relative to the MMLU benchmark, we apply t-SNE to visualize a sample of question embeddings from each dataset alongside those from MMLU (Figure 2). This analysis allows us to confirm that ComSciQA helps to bridge certain content gaps between MMLU and the curated datasets. To assess whether the expanded topic diversity introduced by synthetic data outweighs its potential quality limitations, we investigate the impact of excluding ComSciQA during router training.

Although the t-SNE plot reveals considerable overlap between CombineQA and MMLU, it also highlights regions of MMLU that remain underrepresented, even with the inclusion of ComSciQA. Such regions include subjects like moral scenarios, professional law, professional medicine, and physics. To further investigate this limitation and assess the importance of training on data that is reflective of a model's end use case, we experiment with training directly on the MMLU training subset while reserving its test subset for evaluation. With this in mind, we define three distinct training datasets for our router experiments: CombineQA, which includes both curated and synthetic data; NoSynthQA, which excludes ComSciQA from CombineQA; and MMLU_train, which comprises the training portion of MMLU.

2.1.1 ComSciQA

To address domain discrepancies between MMLU and the four human-curated datasets—and in the absence of a suitable existing computer science dataset—we employed Llama-3.3-70B-Instruct with a one-shot prompting strategy to generate a diverse set of multiple-choice questions covering 100 distinct, LLM-generated computer science topics (Figure 3; see Appendix A for the list of topics). The model outputs were parsed to extract valid JSON-formatted content, yielding an initial set of 33,728 questions. To ensure format consistency, we filtered out entries with improperly formatted answers (e.g., responses containing full answer strings instead of a single letter A–D) as well as questions with fewer than four answer choices. After filtering, the final dataset consisted of 33,540 multiple-choice questions.

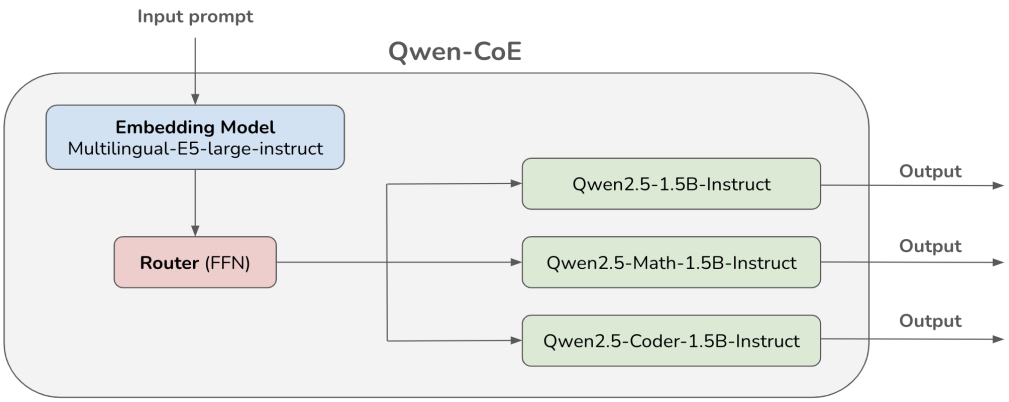

Figure 1: Qwen-CoE architecture: The input prompt is converted into an embedding using Multilingual-E5-large-instruct, which the FFN router uses to select the single best Qwen model to use for inference.

As a synthetic dataset, ComSciQA lacks the human curation of the other four datasets, and any inaccuracies or hallucinations can introduce noise during label assignment (details in Section 2.2) and propagate through training, affecting expert model selection. However, it demonstrates a scalable approach to expanding datasets in underrepresented fields, and can be applied to other domains with sparse data. As such, we briefly investigate whether the benefits of increased training data diversity outweigh the potential risks of errors and inconsistencies in the generated data.

Dataset	Pre-Filter	Post-Filter
ARC	7787	7761
OpenBookQA	11914	11914
MathQA	37901	30946
CommonsenseQA	10962	8799
ComSciQA	33728	33540
CombineQA	102292	92960
NoSynthQA	68564	59420
MMLU_train	99842	99842

Table 2: Number of questions from each dataset before and after filtering.

2.1.2 Data Cleanup

In order to maintain consistency across datasets during the label assignment and model evaluation processes—particularly with respect to format alignment with MMLU, whose questions contain only four answer choices—we applied a filtering step to remove questions where the fifth option (E) was the correct answer. This ensured that all remaining questions could be represented using only the first four choices (A–D), allowing for uniform prompt construction across all datasets. The number of questions retained before and after this filtering step is summarized in Table 2.

2.2 Label Assignment

To generate supervision labels for training the routers, we performed inference on each question in the CombineQA, NoSynthQA, and MMLU_train datasets using all three Qwen models in a five-shot prompting setup (see Appendix B for prompt). For each response, only the first token of the output was extracted and compared against the ground truth answer to determine whether the model successfully answered the question. The label assignment process introduces potential noise from ComSciQA because it may contain factual inaccuracies or hallucinated answers, which could lead to incorrect labels. We analyze the effects of these incorrect labels by excluding ComSciQA when we train Router_NoSynth.

Notably, the Qwen models tended to output numeric values (1–4) rather than letter choices (A–D). This behavior can be attributed to their relatively small size and their limited instruction-following capabilities. To address this, we mapped numerals to their corresponding letter choices (e.g., "1" $\rightarrow$ "A") when evaluating correctness. For each question, we then generated a one-hot encoded label indicating which experts produced the correct answer. An additional fourth label was introduced to represent the case where none of the models answered correctly. In a deployment scenario, this case would signal the need to fallback to a designated default LLM for response generation.

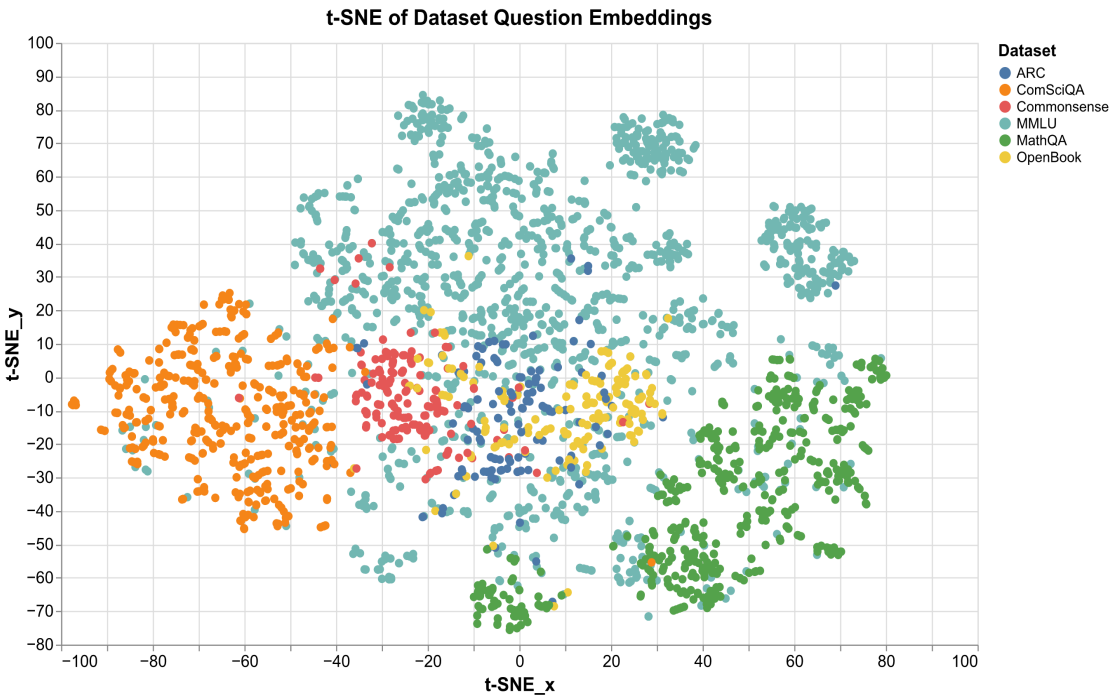

Figure 2: t-SNE visualization of a sample of question embeddings from the five datasets in CombineQA and MMLU. ComSciQA expands the coverage of our training data, however, many subjects in MMLU remain underrepresented.

2.3 Router Training

Since multiple models may be capable of answering a given question correctly, router training is framed as a multi-label classification problem, and we train the routers to predict which Qwen models are likely to produce correct answers based solely on question embeddings. To create input features for the FFNs, we use Multilingual-E5-large-instruct to generate 1024-dimensional embeddings for each question in our datasets. Then we train the routers to predict the 4-dimensional one-hot encoded labels corresponding to our three expert models and the fallback condition. While conventional CoE models would require the router to select a single expert, this study simplifies the task by allowing multiple experts to be considered correct due to limitations in the label assignment process. In a real-world deployment scenario, Qwen-CoE would simply select the expert in which it has the greatest confidence, thus we aim to incorporate expert selection confidence into future router training.

All three routers use the same FFN architecture which consists of an input layer of size 1024, two hidden layers with sizes 256 and 64, respectively, followed by an output layer of size 4. Each output node corresponds to one of the three Qwen experts, with the fourth output indicating that none of the models answered the question correctly (i.e., the fallback condition). For training, we use the Adam optimizer, ReLU activations, and BCEWithLogitsLoss (binary cross-entropy with logits). To address any class imbalances, we use the pos_weight parameter in our loss function to assign greater importance to underrepresented positive labels. This helps prevent the router from becoming biased toward the most frequently correct expert (or the fallback LLM), promoting more balanced learning across all output classes.

During evaluation on the MMLU test set, accuracy is computed as the ratio of correctly predicted answers to the total number of questions. A router prediction is considered correct if it identifies any of the positive labels in the one-hot encoded vector, indicating that it has successfully chosen at least one expert. It is considered incorrect if it selects any model that is not a correct expert.

3 Experiments and Results

3.1 Qwen Model Performance

We evaluate each of the three Qwen models on the MMLU test subset using the same procedure and five-shot prompt used for label assignment in Section 2.2 (Figure 4). BaseQwen achieves the highest overall ac-

```
Prompt: You are a question generator for a multiple-choice test on
Computer Science and Programming.
Generate ten unique and nuanced questions on the topic: "{topic}".
Questions should vary in difficulty and depth of knowledge, and every
question should have four answer choices.

**Format output as a Python list of dictionaries where every
dictionary is like the example below:**
{{
    "question": "What is the time complexity of quicksort in the
average case?",
    "choices": {{
        "label": ["A", "B", "C", "D"],
        "text": ["O(n)", "O(n log n)", "O(n^2)", "O(log n)"]
    }},
    "answerKey": "B"
}}
```

Figure 3: One-shot prompt used to generate the ComSciQA dataset. For each inference, a question topic is chosen at random from a list of 100 LLM-generated topics (see Appendix A for the list of topics).

curacy of 0.411 but demonstrates the greatest variability when analyzed based on the 57 individual subjects (Figure 5). CodeQwen falls behind BaseQwen with an overall accuracy of 0.35 and exhibits slightly less subject-specific variability. MathQwen, with the lowest overall accuracy of 0.276, shows the least variability by subject. Based on these results, we select BaseQwen as the baseline model to evaluate whether any of our three routers can achieve better performance. Additional information about model accuracy by subject can be found in Appendix C.

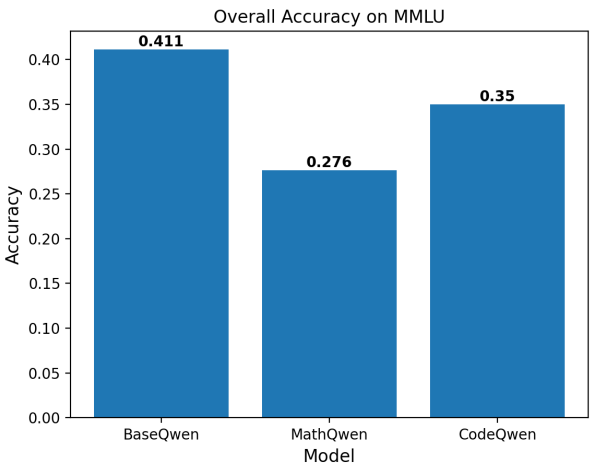

Figure 4: Overall accuracy of each Qwen model on the MMLU test set. Accuracies from left to right: 0.411, 0.276, 0.35.

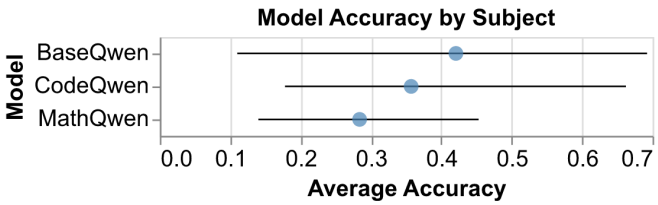

Figure 5: Performance comparison of the three Qwen models across 57 MMLU subjects. Average accuracy is calculated by taking the mean of the accuracies when questions are grouped and evaluated by subject. Error bars indicate the minimum and maximum accuracies for each model.

3.2 Router Performance

We performed a grid search to determine the optimal hyperparameters for training each FFN. The search explored batch sizes of [16, 32, 64], learning rates of [1e-4, 5e-4, 1e-3, 5e-3], and epoch counts of [50, 250, 500, 1000, 1500]. The selected hyperparameters for each router are presented in Table 3.

Router Name	Batch	Learning Rate	Epochs
Router_Combine	16	1e-4	1500
Router_NoSynth	16	1e-3	1500
Router_MMLU	16	5e-4	50

Table 3: Hyperparameters chosen to train each router based on grid search results.

The performance of each of the routers on both MMLU and the test sets from their respective training data are shown in Figure 6. Router_NoSynth achieved the highest accuracy on both MMLU (0.232) and the NoSynthQA test set (0.338). Router_Combine ranked second, with an MMLU accuracy of 0.124 and an accuracy of 0.275 on the CombineQA test set. Router_MMLU performed the worst on both metrics, with an MMLU accuracy of 0.006 and a MMLU_train test set accuracy of 0.073. None of the routers were able to outperform BaseQwen's accuracy of 0.411. We also observe that a router's ability to select the correct expert on its own test set strongly predicts its performance on MMLU, with a Pearson correlation of 0.964.

A comparison between Router_NoSynth and Router_Combine suggests that the inclusion of synthetic data negatively impacts router performance. However, this decline may not stem from the lower quality of the synthetic data, but rather from the increased complexity introduced by a larger set of subjects that the FFN must now learn to distinguish. This is further supported by the poor performance of Router_MMLU, which, despite being trained directly on the MMLU benchmark, struggles significantly with selecting the correct expert during evalution. These findings suggest that the quality of the training data and its alignment with the model's end use case may be less critical than the CoE router's ability to categorize complex inputs.

One possible explanation is that the FFN may not be complex enough to fully distinguish between diverse user inputs, especially as the complexity of the task increases with more training data. Alternatively, the question embeddings themselves may lack the necessary level of differentiability, hindering the router from learning what features are unique to each subject. With this in mind, future work could focus on increasing the complexity of the FFN architecture to enhance its discriminatory power, or on improving question embeddings to better capture the nuances of user inputs. In conjunction with the high correlation between test set and MMLU accuracy, there is promising evidence that refined FFN-based routers can replace conventional routing mechanisms, particularly in scenarios where the CoE model's end use case spans a limited number of domains.

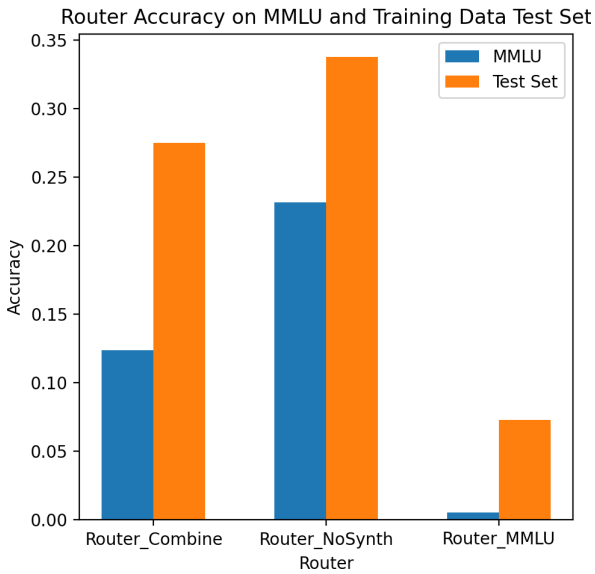

Figure 6: Router accuracy when evaluating on the MMLU benchmark and the test set of the router's training data. Accuracies in order from left to right: 0.124, 0.275, 0.232, 0.338, 0.006, 0.073.

4 Conclusion and Future Work

This work introduces Qwen-CoE, a Composition of Experts architecture that integrates three Qwen2.5 models and replaces the conventional two-step routing mechanism with a simple feedforward network. Using the MMLU benchmark for evaluation, we explore training the router on three datasets: one combining human-curated and synthetic data, one using only human-curated data, and one drawn directly from a subset of the MMLU benchmark. While none of the trained routers outperformed the baseline approach of selecting a single Qwen model for all queries, we attribute this primarily to limitations in the router architecture—specifically, its inability to effectively distinguish between user inputs across different subject areas. Our findings also suggest that the relevance and quality of the training data may be less important than the router's ability to identify subject-specific patterns in the input. Future work will focus on enhancing both the router's input representations and its underlying architecture to improve topic discrimination. We also intend to improve label assignment to train our routers for single-expert selection, favoring models that are both accurate and confident in their responses. Additionally, we aim to experiment with fully synthetic datasets to better understand their applications and limitations.

References

[1] Qin, L., Chen, Q., Feng, X., Wu, Y., Zhang, Y., Li, Y., ... & Yu, P. S. (2024). Large language models meet nlp: A survey. arXiv preprint arXiv:2405.12819.

[2] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

[3] Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., ... & He, Y. (2025). Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948.

[4] Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., ... & Mian, A. (2023). A comprehensive overview of large language models. arXiv preprint arXiv:2307.06435.

[5] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. arXiv preprint arXiv:2001.08361.

[6] Guo, D., Zhu, Q., Yang, D., Xie, Z., Dong, K., Zhang, W., ... & Liang, W. (2024). DeepSeek-Coder: When the Large Language Model Meets Programming–The Rise of Code Intelligence. arXiv preprint arXiv:2401.14196.

[7] Xie, T., Wan, Y., Huang, W., Yin, Z., Liu, Y., Wang, S., ... & Hoex, B. (2023). Darwin series: Domain specific large language models for natural science. arXiv preprint arXiv:2308.13565.

[8] R. Prabhakar et al., "SambaNova SN40L: Scaling the AI Memory Wall with Dataflow and Composition of Experts," 2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO), Austin, TX, USA, 2024, pp. 1353-1366, doi: 10.1109/MICRO61859.2024.00100.

[9] Jain, S., Raju, R., Li, B., Csaki, Z., Li, J., Liang, K., ... & Jairath, S. (2024). Composition of Experts: A Modular Compound AI System Leveraging Large Language Models. arXiv preprint arXiv:2412.01868.

[10] Artetxe, M., Bhosale, S., Goyal, N., Mihaylov, T., Ott, M., Shleifer, S., ... & Stoyanov, V. (2021). Efficient large scale language modeling with mixtures of experts. arXiv preprint arXiv:2112.10684.

[11] Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., ... & Sayed, W. E. (2024). Mixtral of experts. arXiv preprint arXiv:2401.04088.

[12] Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., & Wei, F. (2024). Multilingual e5 text embeddings: A technical report. arXiv preprint arXiv:2402.05672.

[13] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2020). Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300.

[14] Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., ... & Qiu, Z. (2024). Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115.

[15] Yang, A., Zhang, B., Hui, B., Gao, B., Yu, B., Li, C., ... & Zhang, Z. (2024). Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. arXiv preprint arXiv:2409.12122.

[16] Hui, B., Yang, J., Cui, Z., Yang, J., Liu, D., Zhang, L., ... & Lin, J. (2024). Qwen2. 5-coder technical report. arXiv preprint arXiv:2409.12186.

[17] Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., & Tafjord, O. (2018). Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv preprint arXiv:1803.05457.

[18] Mihaylov, T., Clark, P., Khot, T., & Sabharwal, A. (2018). Can a suit of armor conduct electricity? a new dataset for open book question answering. arXiv preprint arXiv:1809.02789.

[19] Amini, A., Gabriel, S., Lin, P., Koncel-Kedziorski, R., Choi, Y., & Hajishirzi, H. (2019). Mathqa: Towards interpretable math word problem solving with operation-based formalisms. arXiv preprint arXiv:1905.13319.

[20] Talmor, A., Herzig, J., Lourie, N., & Berant, J. (2018). Commonsenseqa: A question answering challenge targeting commonsense knowledge. arXiv preprint arXiv:1811.00937.

[21] AI@Meta. (n.d.). Llama 3 Model Card. Retrieved from https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

Appendix A

List of 100 LLM-generated computer science topics used to prompt Llama-3.3-70B-Instruct to generate multiple-choice questions for the ComSciQA dataset.

Algorithms and Data Structures
Introduction to Programming
Object-Oriented Programming (OOP)
Functional Programming
Basic Web Development (HTML, CSS, JavaScript)
Computer Networks
Databases and SQL
Operating Systems
Software Development Life Cycle (SDLC)
Data Structures: Arrays, Linked Lists, Stacks, Queues
Graph Theory
Sorting and Searching Algorithms
Dynamic Programming
Divide and Conquer Algorithms
Recursion
Big O Notation and Time Complexity
Cryptography
Computational Complexity and NP-Completeness
Distributed Systems
NoSQL Databases
Cloud Computing and Cloud Services
Data Mining and Data Warehousing
Machine Learning
Deep Learning
Neural Networks
Natural Language Processing (NLP)
Computer Vision
Artificial Intelligence (AI)
Reinforcement Learning
Supervised and Unsupervised Learning
Model Evaluation and Metrics
Feature Engineering and Feature Selection
Software Testing and Debugging
Test-Driven Development (TDD)
Agile Methodology and Scrum
DevOps Practices
Version Control with Git
Mobile App Development
Game Development and Game Engines
Web APIs (RESTful and GraphQL)
Internet of Things (IoT)
Cybersecurity and Information Security
Ethical Hacking and Penetration Testing
Malware Analysis and Prevention
Arithmetic Operations
Cryptocurrencies and Decentralized Apps

Virtualization and Containerization (Docker, Kubernetes)
Microservices Architecture
Serverless Computing
Edge Computing
Operating System Scheduling Algorithms
Memory Management and Garbage Collection
File Systems and Data Storage
Computer Graphics and Rendering
3D Modelling and Animation
Computational Geometry
Formal Languages and Automata Theory
Parallel and Concurrent Programming
Multithreading and Synchronization
Cloud Native Development
Software Design Patterns
Dependency Injection and Inversion of Control
Model-View-Controller (MVC) Architecture
Microkernel Architecture
Event-Driven Programming
Blockchain Consensus Algorithms (Proof of Work, Proof of Stake)
Cloud Platforms (AWS, Google Cloud, Microsoft Azure)
Real-Time Systems
Distributed Databases and CAP Theorem
Network Protocols (TCP/IP, DNS, HTTP)
HTTP/2 and HTTP/3 Protocols
Artificial Neural Networks (ANNs)
Convolutional Neural Networks (CNNs)
Recurrent Neural Networks (RNNs)
Generative Adversarial Networks (GANs)
Data Visualization and Dashboards
Data Science and Statistical Analysis
Business Intelligence and Data Analytics
Quantum Computing
Introduction to Blockchain and Smart Contracts
Natural User Interfaces (NUI) and Gesture Recognition
Digital Signal Processing
Image Processing and Computer Vision
Video Streaming Technologies
Computer Forensics
Social Network Analysis
Cloud Security and Data Privacy
Zero Trust Security Model
Computer Simulation and Modeling
Database Indexing and Optimization
Internet Protocols and Routing Algorithms
Web Security and Secure Coding Practices
Augmented Reality (AR) and Virtual Reality (VR)
Human-Computer Interaction (HCI)
Cloud Computing Security Risks and Mitigation
Neural Network Optimization and Hyperparameter Tuning

Hardware Design and Embedded Systems
Compiler Design and Optimization
Digital Cryptography and RSA Algorithm
Software Architecture and Design Principles

Appendix B

Five-shot prompt used during label assignment and MMLU evaluation. Inference was run using all three Qwen models. The following fields were filled in dynamically for each question and its answer choices: question, choiceA, choiceB, choiceC, and choiceD.

```
Prompt: The following are multiple choice questions (with answers) about a
variety of topics.

Question: Which organelle is responsible for producing energy in a cell?
A. nucleus
B. mitochondrian
C. ribosome
D. golgi apparatus
Answer: B

Question: If 3x-5=16, what is the value of x?
A. 2
B. 5
C. 7
D. 11
Answer: C

Question: What type of energy is stored in an object due to its position?
A. chemical energy
B. kinetic energy
C. thermal energy
D. potential energy
Answer: D

Question: Which sentence is written in passive voice?
A. The lesson was explained clearly by the teacher.
B. The teacher explained the lesson clearly.
C. The students asked many questions.
D. The teacher enjoys explaining difficult topics.
Answer: A

Question: The Renaissance was a period of cultural rebirth that began in
which country?
A. France
B. England
C. Italy
D. Germany
Answer: C

Question: {question}
A. {choiceA}
B. {choiceB}
C. {choiceC}
D. {choiceD}
Answer:
```

Appendix C

All Qwen models had varying accuracies when grouping and evaluating MMLU questions by subject.