FINAL PROJECT

Shunian Chen, Juhao Liang, Xidong Wang, Ke ji, Rui Huang, Yuqi Fei, Benyou Wang*
The Chinese University of Hong Kong, Shenzhen
wangbenyou@cuhk.edu.cn

1 INTRODUCTION

By the end of the semester, we believe you will have a solid understanding of Natural Language Processing (NLP). For your final project, you will delve into NLP and complete a scientific piece of research. First, you need to write a final project proposal (Sec. 3) outlining your research planning, upon completion at the end of the semester you need to submit a project report (max 6 pages). Additionally, you will need to prepare a project poster presentation. After the final project deadline, feel free to make your project open source; we appreciate if you acknowledge this course.

Regarding the topic selection: you could

- 1) select one topic from Sec. A **without the need to submit a proposal**;
- or 2) write your own proposal following Sec. 3.

In this **final project**, we encourage you to apply your acquired knowledge to a LLM-related task or problem that sparks your interest. Your goal is to develop a solution using the skills and concepts learned, culminating in a final paper and a in-class presentation.

2 SUBMIT YOUR PROPOSAL

First of all, you need to finalize your team and your project topic. Please submit your team member information and project topic.

2.1 TEAM WORK

Projects can be completed individually or in teams, with the following guidelines:

- **Team Size:** Projects can be done solo or in teams of up to 3.
- **Teamwork Encouraged:** We recommend completing the final project in a team. Larger teams are expected to undertake correspondingly larger projects.
- **Contribution:** The final report should include each member's contributions.
- **External Collaborators:** You may collaborate with non-course participants but must clarify your specific contributions in the final report.

2.2 TOPIC SELECTION

For topic selection, you may:

- 1) choose a topic from Sec. A, where we provide 10 topics;
- or 2) Propose your own topic, but it must be related to NLP.

3 REQUIREMENTS

The final project requires both a poster presentation and a final paper:

*Benyou Wang is the instructor.

- **Poster Presentation [Date: Apr 24, 2025]:** You need to design a poster and give an on-site presentation showing your work. A poster template is provided at: https://docs.google.com/presentation/d/1pBJuB-wazGyGHTiDNQ6msihS2cyyZDYL71VgE6o3FrE/edit#slide=id.p1.
- **Final Paper [Deadline: May 9, 2025]:** Submit a clear and concise report (in PDF format) introducing your final project. The report template can be accessed at https://www.overleaf.com/read/ypsxqykqwkyr#65832b.

4 Grading Criteria

For Students in CSC4100, please refer to our website for detailed grading policy. For students in CSC6052/DDA6307/MDS6002, the final project accounts for 55% of the total grade. It consists of two parts: **Project Poster (10%)** and **Project Report (45%)**.

- **Project Poster (10%)**: You are required to design your project poster using the specified Poster template. Your poster presentation will be rated by at least 3 experts (TAs and at least one external professor or scientist from industry). The average rating will be the final credit.
  - **Content quality (1%)**: Well-presented posters or slides are highly valued.
  - **Oral presentation (4%)**: Clear and enthusiastic speaking is encouraged.
  - **Overall subjective assessment (5%)**: Although subjective assessment might be biased, it happens everywhere!
- **Project Report (45%)**: The project report will be publicly available after the final poster session. Please let us know if you do not wish so.
  - **Technical excitement (10%)**: It is encouraged to do something that is either interesting or useful!
  - **Technical soundness (15%)**: A) Discuss the motivation on why you work on this project and your algorithm or approach. Even if you are reproducing a published paper, you should have your own motivation. B) Cite existing related work. C) Present your algorithms or systems for your project. Provide key information for reviewers to judge whether it is technically correct. D) Provide a reasonable evaluation protocol, detailed enough to contextualize your results. E) Report quantitative results and include qualitative evaluation. Analyze and understand your system by inspecting key outputs and intermediate results. Discuss how it works, when it succeeds and fails, and try to interpret why it works and why not.
  - **Clarity in writing (15%)**: The report is written in a precise and concise manner so the report can be easily understood.
  - **Individual contribution (5%)**: This is based on individual contribution, probably on a subjective basis.
- **Bonus and penalty**: Note that the project credit is capped at 55%.
  - **TA favorites (2%)**: If one of the TAs nominates the project as their favorite, the involved students would get 1% bonus credit. Each TA could nominate one project and could reserve their nomination. This credit could only be obtained once.
  - **Instructor favorites (1%)**: If the instructor nominates the project as their favorite, the involved students would get 1% bonus credit. The instructor could nominate at most three projects. One could get both TA favorites and Instructor favorites.
  - **Project early-bird bonus (2%)**: If you submit the project report by the early submission due date, 2% bonus credit will be entitled.
  - **Code reproducibility bonus (1%)**: One could obtain this if TAs think they could easily reproduce your results based on the provided material.
  - **Ethics concerns (-1%)**: If there are any serious ethics concerns by the ethics committee (the instructor and all TAs), the project would get a 1% penalty.

ACKNOWLEDGMENT

Please acknowledge this course if you publish any materials based on this assignment.

A OVERALL PICTURE

A.1 TOPIC 1: VERTICAL LLMS

The domain of Vertical large language models (LLMs) encompasses a wide range of subjects including Medicine, Law, Finance, Software Engineering, Science, Geometry, Math, Optimization, and Education, among others. The challenge in this field lies in effectively injecting new knowledge into an existing LLM, which requires a comprehensive pipeline that integrates continuous learning and updating mechanisms. This process must ensure that the model can assimilate and apply the newly introduced information accurately and efficiently, while maintaining its existing knowledge base and performance. It involves sophisticated algorithms and techniques to ensure that the LLM remains relevant and enhances its capabilities without compromising the integrity of its existing structure and knowledge.

It might involve:

(1) Continued pre-training is an extension of the initial pre-training phase of large language models (LLMs), where the model is further trained on new data to enhance its performance and adapt to specific domains or tasks. This process often requires significant computational resources, such as GPUs, to handle the large scale of data and model parameters involved.

(2) Supervised-tuning involves fine-tuning the LLM with a smaller, task-specific dataset, which includes labeled examples to guide the model towards desired outputs. Data collection is a critical step that precedes both pre-training and fine-tuning, where high-quality corpora, either structured or unstructured, are assembled to serve as the foundation for model learning. Self-instruction is a technique where the model generates prompts and responses, creating a dataset that can be used for further training to improve the model's ability to follow instructions and generate relevant outputs.

(3) Reinforcement Learning from Human Feedback (RLHF) is a more interactive approach where human feedback is used to shape the model's behavior through a reward mechanism, aiming to align it with human values and preferences. Especially, direct preference optimization (DPO) could be a good practice.

(4) Retrieval-Augmented Generation (RAG) are advanced training methods that incorporate external knowledge to enhance the model's factual accuracy and generative capabilities. Each of these strategies contributes to the ongoing development and improvement of LLMs, ensuring they remain at the forefront of natural language processing technology.

A.2 TOPIC 2: IMPROVEMENT ON A SPECIFIC ABILITY

This project aims to improve one of the following ability of LLMs:

- **Alignment** (RLHF) involves the use of reinforcement learning from human feedback to align the behavior of large language models (LLMs) with human values and preferences, ensuring that the model's outputs are safe, helpful, and unbiased.
- **Math Reasoning**, as demonstrated in datasets like GSM8K, focuses on enhancing the LLMs' capabilities in mathematical reasoning and problem-solving.
- **Reducing LLM Hallucinations:** LLMs can generate incorrect or false content, known as hallucinations. Try to improve the reliability of generated content, perhaps by providing relevant knowledge or context.
- **Multi-turn conversation** benchmarks such as MT-Bench and Alpaca-eval aim to improve the fluency and coherence of LLMs in handling extended dialogues, maintaining context across multiple exchanges.
- **Tool using**, as seen in ToolBench, explores the integration of LLMs with external tools and APIs to extend their applicability beyond text generation, enabling tasks like data retrieval and calculation.
- **Agent** in AgentBench are critical for tasks requiring LLMs to act as intelligent agents, responding effectively to user prompts and performing complex tasks.

- **Embodied AI** Develop tools and frameworks that integrate LLMs with embodied agents, such as robots or virtual avatars. This involves creating systems where LLMs can control or influence physical or virtual entities, making decisions based on sensor input, and interacting with the environment in meaningful ways. The challenge is to bridge the gap between abstract language understanding and real-world action.
- **Automatic theorem proving** and coding are specific skill sets that LLMs can be trained to develop, enabling them to solve mathematical proofs and write computer programs, respectively.
- **Instruction following** is a fundamental ability for LLMs to accurately execute tasks based on natural language instructions provided to them. By focusing on these areas, the overall performance and versatility of LLMs can be significantly enhanced, making them more capable and reliable for a wide range of applications.
- **Generation Detection:** Work on developing methods to distinguish between human-generated and model-generated text. This is crucial for identifying and labeling content created by LLMs, especially in contexts where transparency is important, such as news, academic research, or legal documents. Techniques could involve stylistic analysis, consistency checks, or incorporating digital signatures in model outputs.

A.3 Topic 3: Evaluation of LLMs

Investigate the large language models, like ChatGPT, Qwen, Baichuan, Jamba, GPT-4, Mxitral, to assess their capabilities, limitations, and potential risks. You are also encouraged to develop your own evaluation methods and datasets, assessing the performance of LLMs in a field you're familiar with.

- **Chinese culture**: Testing LLMs on Chinese culture involves assessing their understanding of the language's rich history, idioms, proverbs, and cultural references. The motivation is to create models that can accurately interpret and generate content relevant to Chinese-speaking users, thereby providing a more personalized and culturally sensitive experience.
- **Region stereotype**: Evaluating how LLMs handle region stereotypes is crucial for ensuring that AI does not perpetuate or reinforce harmful biases. The goal is to develop models that are aware of and can challenge stereotypes, promoting inclusivity and diversity in AI-generated content.
- **Sense making**: As discussed earlier, sense making is the ability of LLMs to interpret complex inputs and derive meaning. Testing this aspect aims to improve the model's comprehension and reasoning skills, leading to more nuanced and contextually appropriate responses.
- **Formal logics**: Assessing LLMs' capabilities in formal logics helps ensure that they can reason and make decisions based on structured arguments. This is important for applications that require logical consistency and sound reasoning, such as legal or philosophical discussions.
- **Humor**: Understanding humor is a complex task that requires cultural awareness, linguistic creativity, and emotional intelligence. Testing LLMs on humor aims to create models that can engage users in a more human-like manner, providing entertainment and fostering a more enjoyable interaction.
- **Multi-modal problems** (vision LLM and speech LLM): Evaluating LLMs on multi-modal problems involves assessing their ability to integrate and process information from different sensory inputs, such as vision and speech. The motivation is to develop models that can interact with the world in a more holistic way, similar to how humans perceive and understand their environment.
- **Long-context evaluation**: Testing LLMs on their ability to maintain and understand long contexts is essential for applications that require remembering and referencing extensive conversations or narratives. This capability ensures that models can provide coherent and contextually relevant responses in complex dialogues or storytelling.
- **Multiple-turn conversation**: Assessing LLMs on multiple-turn conversations helps improve their ability to engage in extended dialogues, maintaining context and coherence

throughout the interaction. This is crucial for applications like customer service, where understanding and addressing user needs over multiple exchanges is key.

- **EQ (Emotional Quotient)**: Testing the EQ of LLMs involves evaluating their ability to recognize, understand, and respond to emotions in text. High EQ is desirable for applications that require empathetic and emotionally intelligent interactions, such as mental health support or customer service.
- **Ethical LLM:** Explore the ethical implications of deploying LLMs in various sectors, such as education, healthcare, or finance. This includes assessing the impact of LLM advice or decisions on individuals and society, developing guidelines for responsible usage, and creating mechanisms to prevent misuse or unintended consequences.

Or evaluate the weakness of existing LLMs especially for

- **Numerical Sensitiveness for financial LLMs**: For financial applications, it's critical that LLMs can accurately handle and interpret numerical data. Testing their sensitivity to numbers ensures that they can provide reliable financial advice, analysis, and reporting without errors.
- **Errors of medical common sense for biomedical LLMs**: In the biomedical domain, LLMs must adhere to established medical knowledge and practices. Testing for common sense errors helps ensure that the models do not generate misleading or dangerous health-related content.
- **Jailbreak**: Evaluating the "jailbreak" capabilities of LLMs, which refers to their ability to overcome limitations or constraints imposed on them, is important for understanding the potential risks and ensuring that AI systems remain safe and ethical.
- **Safety and privacy risks**: Testing LLMs for safety and privacy risks involves assessing their susceptibility to generating harmful content or disclosing sensitive information. The goal is to develop models that prioritize user safety and data privacy.

Each of these aspects is tested to ensure that LLMs can operate effectively and safely in a wide range of applications, from everyday conversation to specialized tasks requiring domain-specific knowledge and skills. The ultimate goal is to create AI systems that can interact with humans in a natural, meaningful, and contextually appropriate manner.

A.4 Topic 4: Dataset building

One could build dataset for LLs, such as Pre-training corpora, supervised data, and preference data are essential components in the development of large language models (LLMs).

Pre-training corpora To construct a robust dataset for pre-training, one must gather diverse and high-quality information from sources like Common Crawl, which provides a broad spectrum of web-scraped content. This foundational data is then complemented with domain-specific datasets, such as those focused on mathematics, medicine, finance, and other areas where precision and expertise are crucial. The process of extracting super high-quality data involves rigorous cleaning and deduplication to ensure the integrity and reliability of the information that the LLMs will learn from.

Supervised data The collection of supervised data involves identifying a set of tasks relevant to a specific domain and amassing data that can guide the LLMs in performing these tasks effectively. This can be achieved by utilizing existing datasets and crafting prompts that align with the desired outcomes, as well as by creating new datasets through a combination of human expertise and LLM-generated content. Building a comprehensive taxonomy of tasks and data types facilitates a systematic approach to training LLMs and ensures that they are equipped to handle a wide range of scenarios and challenges.

Preference data Preference data is collected to fine-tune the LLMs and align their outputs with user expectations and preferences. This can involve creating prompts that elicit multiple responses, thereby providing a variety of options for users to choose from. Feedback mechanisms, such as yes/no feedback, pairwise ranking, and textual comments, help refine the model's performance. Sources of feedback can range from rule-based systems that verify the correctness of code execution to expert knowledge that enhances the model's understanding of complex subjects. Additionally,

LLM-generated feedback can be used in a recursive manner to further improve the model's capabilities. By integrating these diverse datasets and feedback loops, LLMs can be trained to deliver more accurate, contextually relevant, and user-centric responses.

A.5 Topic 5: HCI applications and Agent

Investigate the dynamics of how humans interact with LLMs. This area could include studying user behavior, developing more intuitive interfaces for interaction, or creating models that adapt to individual user preferences and styles of communication. The aim is to enhance the usability and effectiveness of LLMs in everyday tasks and professional settings. HCI applications usually involves multiple LLM agents through cooperation or interaction. This sometimes involve having multiple LLMs debate to reach better answers or collaborate on extraordinary tasks. See three examples as follows.

A.5.1 AI Town

AI Town is a concept that envisions a virtual environment where large language models (LLMs) are integrated into various aspects of the town's infrastructure and daily life. In this project, AI-driven systems would be responsible for managing and improving the efficiency of public services, facilitating communication between residents and local government, and providing personalized recommendations for activities, events, and community engagement.

For instance, AI Town could include an intelligent transportation system that uses LLMs to analyze traffic patterns and optimize routes for public transit. The AI could also be used in educational settings, offering tailored learning experiences for students based on their individual needs and interests. In addition, AI Town could feature a virtual assistant that residents interact with to access information, report issues, or get recommendations for local businesses and services.

A.5.2 LLM Powered Educational Games

LLM powered educational games represent a project that leverages the capabilities of large language models to create engaging and interactive learning experiences. These games could be designed to teach a variety of subjects, from language learning and history to mathematics and science, in a way that is both fun and informative.

In such games, LLMs could be used to generate dynamic content that adapts to the player's knowledge level and learning pace. For example, a language learning game might present sentences or dialogues for the player to translate or complete, gradually increasing in complexity as the player progresses. Similarly, a history game could present events or figures for the player to interact with, using the LLM to provide detailed information and context based on the player's choices and inquiries.

A.5.3 LLM Plus Metaverse

Combining large language models with the concept of the metaverse opens up a world of possibilities for immersive and interactive virtual experiences. In this project, LLMs would be an integral part of a virtual universe where users can interact with each other and the environment through natural language.

Within the metaverse, LLMs could be used to create non-player characters (NPCs) with rich backstories and the ability to engage in meaningful conversations with users. They could also be used to generate dynamic content, such as quests, storylines, and environments that evolve based on user interactions. For instance, an LLM could craft a personalized adventure for each user, taking into account their preferences, past actions, and in-the-moment decisions.

Moreover, LLMs could facilitate real-time translation between languages, making the metaverse a truly global platform where users from different linguistic backgrounds can communicate seamlessly. They could also be employed to moderate and filter content, ensuring a safe and inclusive environment for all users. This project represents the potential for LLMs to not only enhance our digital experiences but also to bridge the gap between the virtual and the real world.

A.5.4 FINANCIAL MARKET PREDICTION AGENT

Develop an agent that utilizes machine learning and data analysis to predict stock market trends and conduct simulated trading. This involves acquiring historical stock market data and using data science tools (such as Python's Pandas and Scikit-learn libraries) for data cleaning and analysis. The agent should learn to recognize market patterns and apply technical analysis indicators, such as moving averages and relative strength index, to predict future market trends.

The agent will test its predictive capabilities through a simulated trading environment and attempt to optimize the investment portfolio to maximize returns. Evaluating the agent's performance and considering risk management strategies, such as stop-loss and diversification, are essential components of this project. This project requires effectively applying machine learning techniques to handle real-world data.

A.5.5 NPC BEHAVIOR DESIGN IN GAMES

Design intelligent NPCs in an open-world game using AI technology to achieve complex behavior patterns and interactions. This involves creating behavior models for NPCs in the game, ensuring these characters can make reasonable decisions and interact in a dynamic game environment.

The agent needs to understand player actions and respond appropriately, such as providing clues or challenges when players approach. Game development engines (such as Unity or Unreal Engine) combined with AI techniques (such as finite state machines, behavior trees, or reinforcement learning) can be used to implement complex NPC behaviors.

The project's goal is to enhance game immersion, allowing players to experience more realistic interactions and challenges. This requires knowledge of game design and the application of AI techniques.

A.5.6 SOCIAL MEDIA CONTENT MANAGEMENT AGENT

Create an agent capable of automatically generating, filtering, and publishing social media content, and interacting with users. The system should be able to generate relevant content based on trend analysis and user preferences and publish it at appropriate times. The agent should possess natural language processing capabilities to understand and generate text content, as well as data analysis capabilities to evaluate the popularity and impact of the content.

The project may involve using social media APIs for data collection and content publishing, machine learning for trend prediction, and automation tools for content management and scheduling. This project focuses on applying AI technology in digital marketing to improve interaction efficiency and content quality.

A.6 TOPIC 6: ADAPT LLM TO A NEW LANGUAGE

This project aims to efficiently adapt LLaMa2/Mistral to a new language, Like in Chinese or Arabic. This could also a project for **Cross-Lingual Understanding**: focus on enhancing the ability of LLMs to understand and generate text across multiple languages, including low-resource languages. This includes research on translation, context preservation, and cultural nuances in communication. The goal is to make LLMs truly global tools that can bridge language barriers effectively.

A.7 TOPIC 7: MEDICAL APPLICATIONS

LLM for Triage (医院预分诊): Using a large language model (LLM) for triage in healthcare involves training the model to prioritize patients based on the severity of their conditions. This can help medical staff quickly identify cases that require immediate attention, thus improving the efficiency of emergency departments and potentially saving lives.

- **Preparing a Dataset**: To train an LLM for medical triage, a dataset must be prepared, which includes detailed patient information, symptoms, medical histories, and the corresponding triage categories or levels of urgency. This dataset should be comprehensive,

representative of various medical conditions, and annotated with accurate outcomes to ensure effective learning.

- **Training its training split using LLMs**: Once the dataset is prepared, the training split is used to teach the LLM how to assess patient data and assign appropriate triage levels. During this phase, the model learns from examples and adjusts its predictions based on the annotated outcomes, gradually improving its ability to make accurate assessments.
- **Testing in the test split**: After training, the model's performance is evaluated using a separate test split of the dataset. This helps to measure the model's effectiveness in real-world scenarios and ensures that it generalizes well to new, unseen patient cases.

Medical FLAN: FLAN, short for "Federated Learning of Anomalies," is a technique that can be adapted for medical applications. In the context of triage, a Medical FLAN system could be trained to detect anomalies or rare conditions that might be overlooked by standard triage protocols, thereby improving the overall accuracy and safety of the process.

Large-scale instruction data: To further refine the LLM's capabilities, large-scale instruction data from medical professionals can be used. This includes verbal and written instructions, advice, and guidelines that provide the model with a deeper understanding of medical protocols and best practices.

Medical Pajama: For an LLM to be effective in medical applications, it requires extensive training data that covers a wide range of medical conditions and scenarios. This large-scale data ensures that the model can recognize patterns, understand the nuances of different medical cases, and make informed triage decisions.

Chain of Diagnosis (COD) in medical domain: The Chain of Diagnosis (COD) is a concept where the LLM is trained to follow a logical sequence of diagnostic steps when assessing a patient's condition. This structured approach helps the model to consider all relevant factors and make more accurate triage decisions.

X-ray report generation: LLMs can also be trained to generate X-ray reports or interpret radiology images, providing valuable support to radiologists and other medical professionals. By understanding the technical language and common findings in X-ray reports, the LLM can assist in quickly producing accurate and detailed reports, improving the speed and efficiency of diagnosis.

Patience simulator for doctors training: A "patience simulator" for doctors could refer to a training tool powered by LLMs that simulates patient interactions, allowing medical professionals to practice their triage and communication skills. This simulator could generate diverse patient profiles, symptoms, and responses, helping doctors to develop their decision-making and empathy skills in a controlled environment.

A.8 Topic 8: LLMs for other modality, e.g. speech or EEG

The integration of encoders and adapters in constructing new modalities for multimodal large models (MM-LLMs) is a pivotal approach to enhance the versatility of AI systems. By utilizing specialized encoders, such as CNNs for images or RNNs for audio, these models can process and interpret diverse data types beyond text, including visual and auditory information. The encoders transform raw data into numerical embeddings that capture the essence of the input. Adapters then serve as connectors, aligning these new modality embeddings with the model's existing structure, allowing for seamless integration without extensive retraining. This modular expansion of MM-LLMs enables more comprehensive and context-rich AI interactions, as the models can leverage combined information across modalities for tasks like understanding, generation, and translation, thus moving closer to human-like cognition and problem-solving capabilities

Shorter Output To address the issue of generating overly lengthy responses, especially in spoken scenarios where brevity is preferred, several strategies can be implemented. One approach is to introduce a prompt or a mechanism within the model that prioritizes conciseness, encouraging it to provide shorter, more focused answers.

Longer Input Moreover, to support long context requirements while maintaining brevity, the model can be trained to better understand and retain context, allowing it to provide relevant information

without needing to repeat previously mentioned details. This involves enhancing the model's memory and attention mechanisms, so it can effectively track conversational history and identify key points without excessive elaboration.

A.9 Topic 9: LLM on edges and embodied AI

This project aims to study a LLM running on a edge device that might has some operations controlled by LLMs. This involves reasoning and planning. Particularly, applications on Edge devices usually need real-time inference. Therefore, Acceleration is sometimes encouraged.

Acceleration for Large Language Model : Focus on methods and technologies to speed up the training and inference processes of large language models (LLMs). This could include exploring new hardware architectures, optimizing algorithms for parallel processing, or developing more efficient data processing techniques. The goal is to make LLMs more accessible and cost-effective, enabling broader usage and experimentation.

A.10 Topic 10: A survey for a specific topic

Writing a survey paper is crucial for providing a comprehensive overview of a specific topic, identifying key areas of research, and summarizing the current state of knowledge. This type of paper is essential for understanding the trajectory of a field, highlighting significant contributions, and mapping out the progression of ideas over time. A well-crafted survey offers a taxonomy of existing literature, classifying papers based on methods, applications, or theoretical frameworks, which helps in identifying gaps and potential directions for future research. Additionally, it discusses limitations of current studies and suggests trends that may emerge, fostering informed predictions about the subject's evolution. Writing a survey paper involves a systematic review of relevant literature, critical analysis of methodologies and findings, and a clear presentation of information that is both accessible and insightful for readers, including researchers and practitioners alike. It serves as a valuable reference for those looking to understand the past, navigate the present, and anticipate the future of a particular area within academia or industry.

A.11 Special Topic I: AI Mathematical Olympiad (AIMO)

AIMO is a comprehensive project topic aimed at utilizing artificial intelligence to optimize the process of solving mathematical problems. As AI continues to expand its applications across various fields, using AI to enhance the efficiency, accuracy, and creativity in mathematical problem-solving is becoming increasingly important. This topic explores four key subtopics, engaging in practical applications and research to advance AI's role in mathematics.

Prompting Engineering and Sampling Strategies: The goal is to design effective prompts that guide AI models in solving mathematical problems more efficiently. This involves researching and implementing different prompt structures and sampling strategies to optimize inputs, thereby improving the quality and accuracy of model outputs. This approach is applied to complex mathematical problems, such as geometric proofs and algebraic calculations, using advanced prompting techniques.

Test Time Scaling: This subtopic focuses on optimizing model performance during test time, ensuring real-time computation and efficient inference. It involves studying methods for dynamically adjusting model size and resource allocation to meet diverse testing conditions. By enhancing AI system responsiveness and problem-solving efficiency, particularly in simulated mathematics competition environments, it effectively scales model performance.

How to Use Code and Tools This subtopic explores how large AI models can leverage code and external tools to enhance their ability to solve mathematical problems. The aim is to improve efficiency and accuracy in mathematical computations by integrating advanced software tools and programming techniques.

Long2short DPO Here, the objective is to compress complex mathematical problems and their solution steps into concise answers and key steps. It involves researching text compression and information extraction techniques to enhance the efficiency and readability of mathematical solu-

tions. This includes designing AI systems capable of converting lengthy mathematical proofs or computational processes into refined, step-by-step hints and answers.

A.12 Special Topic II: Advanced AI Agent Systems

A.12.1 Agent Workflow

Dify Workflow: Explore and master the Dify workflow, a structured framework for managing AI tasks and processes. Using resources from GitHub, learn how to design and optimize AI applications. Study how to utilize Dify to create efficient AI task management systems, including task scheduling, resource allocation, and performance monitoring, to develop scalable AI applications that operate effectively in real-world environments.

RAG Corpora: Investigate Retrieval-Augmented Generation (RAG) techniques to enhance AI models' performance in generating contextually relevant content. Combine large datasets to improve the accuracy and relevance of model outputs. Understand the fundamental principles and implementation methods of RAG technology to apply it in natural language processing applications such as question-answering systems and dialogue agents.

MCP (Model Context Protocol): MCP is a standard protocol that equips AI models with a "universal interface," allowing them to seamlessly interact with different data sources and tools. It functions like a USB-C interface, providing a standardized method to connect AI models to various data sources and tools. This subtopic focus on the Model Context Protocol to optimize how AI models interact with their contextual environments. Learn how to define and implement protocols that allow models to understand and adapt to the context in which they operate. This involves designing systems that enhance model performance by effectively leveraging contextual information.

Tool Hub: Explore platforms that integrate various tools and resources essential for developing and deploying AI solutions. Learn how to use these tools to streamline development processes, including accessing and utilizing technical libraries. Establish an efficient development environment to support the rapid development and deployment of AI applications.

A.12.2 Autonomous Agent

OpenManus: Study the capabilities of OpenManus and understand how autonomous agents manage workflows without human intervention. Develop and deploy systems that can automatically handle documents and tasks, enhancing productivity and automation.

OWL (Open World Learning): Explore the OWL framework to build AI systems capable of dynamically adapting to new information and environments. Create autonomous learning systems that operate in ever-changing environments, supporting continuous learning and adaptation.

B Available Resources

B.1 Model Pruning

- LLM-Pruner: On the Structural Pruning of Large Language Models. `https://github.com/horseee/LLM-Pruner`
- ShortGPT: Compressing models through layer pruning. `https://arxiv.org/abs/2403.03853`

B.2 Parameter-efficient Architecture for LMs

- Mixture-of-Expert (MoE): `https://arxiv.org/abs/2401.04088`
- State-of-the-art Parameter-Efficient Fine-Tuning (PEFT) methods `https://github.com/huggingface/peft`

B.3 LMs with Tools

- ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs `https://arxiv.org/abs/2307.16789`

B.4 MERGE SMALL MODELS

- All we need is just a few fine-tuned models: https://arxiv.org/abs/2403.19522
- Training-Free Pretrained Model Merging: https://arxiv.org/pdf/2403.01753.pdf

B.5 DECODING LEARNING

- Proxy Tuning: https://arxiv.org/abs/2401.08565
- Decoding-time Realignment of Language Models: https://arxiv.org/abs/2402.02992

B.6 JAILBREAKING

- Weak-to-Strong Jailbreaking on Large Language Models: https://arxiv.org/abs/2401.17256

B.7 TINYAGENT

- ReALM: https://arxiv.org/pdf/2403.20329.pdf
- Multi-Agent Collaboration: https://arxiv.org/abs/2404.01663

B.8 MEDICAL LLM

- HuatuoGPT https://arxiv.org/abs/2305.15075
- HuatuoGPT-II https://arxiv.org/abs/2311.09774

B.9 MULTIMODAL LLM

- LLaVA https://llava-vl.github.io/
- A-LLaVA https://arxiv.org/abs/2402.11684

B.10 LLM FOR DIAGNOSIS

- Automatic Diagnosis of Diseases https://arxiv.org/abs/2402.03271

B.11 SELF-IMPROVE

- Enable LLMs to self-learn and self-improve https://arxiv.org/pdf/2401.01335.pdf
- Synthetic data for instruction tuning https://arxiv.org/abs/2402.13064
- Math reasoning with LLMs https://arxiv.org/abs/2403.02884

B.12 GEOMETRIC DEEP LEARNING

- Hyperbolic Deep Reinforcement Learning https://arxiv.org/abs/2210.01542
- Neural Algorithmic Reasoning https://arxiv.org/abs/2105.02761
- Unsupervised Learning of Group Invariant and Equivariant Representations https://arxiv.org/abs/2202.07559

B.13 AI4SCIENCE

- Retrosynthetic planning https://arxiv.org/abs/2209.15315

B.14 PROMPT TUNING

- Prefix-Tuning: Optimizing Continuous Prompts for Generation https://arxiv.org/abs/2101.00190
- InfoPrompt: Information-Theoretic Soft Prompt Tuning for Natural Language Understanding https://arxiv.org/abs/2306.04933

B.15 CROSS FIELDS

- Hyperbolic Deep Reinforcement Learning https://arxiv.org/abs/2210.01542
- Transformer Hawkes Process https://arxiv.org/abs/2002.09291

B.16 MATH AI

- OVM, Outcome-supervised Value Models for Planning in Mathematical Reasoning https://arxiv.org/abs/2311.09724
- Let's Verify Step by Step https://arxiv.org/abs/2305.20050

B.17 ALIGNMENT

- Training language models to follow instructions with human feedback https://arxiv.org/abs/2203.02155
- Direct Preference Optimization: Your Language Model is Secretly a Reward Model https://arxiv.org/abs/2305.18290