Mixture-of-Agents Enhances Large Language Model Capabilities

Author: Wang et al.

created by paper2slides

arxiv:2406.04692
2024-06-07

Executive Summary

- Proposes a Mixture-of-Agents (MoA) methodology to enhance the capabilities of Large Language Models (LLMs).
- MoA constructs a layered architecture of LLM agents, where each agent utilizes outputs from previous layers for refinement.
- Achieves state-of-the-art performance on AlpacaEval 2.0, MT-Bench, and FLASK benchmarks, surpassing GPT-4 Omni.
- Leverages *collaborativeness* among LLMs to iteratively improve response quality.

Introduction and Motivation

- LLMs have advanced natural language understanding and generation but face limitations in model size and training data.
- Different LLMs specialize in various tasks, presenting an opportunity to harness their collective strengths.
- Key idea: Use multiple LLMs where each model improves its performance by leveraging responses from other models.

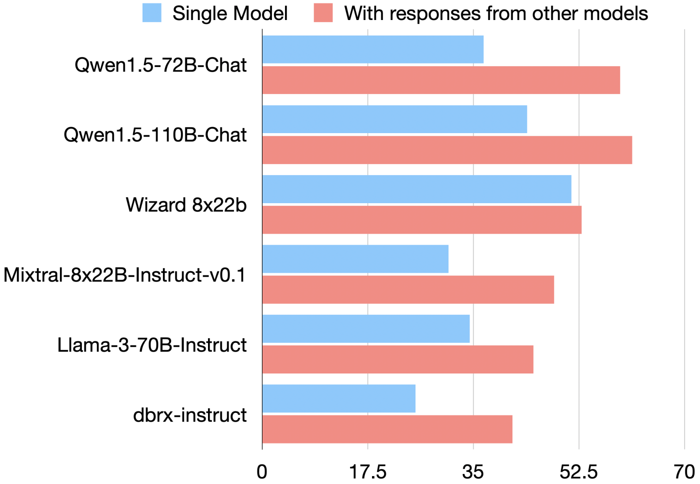

Figure: AlpacaEval 2.0 LC win rate improves when provided with responses from other models.

Collaborativeness among LLMs

- LLMs generate higher-quality responses when they have access to outputs from other models.
- Collaborativeness is observed even when auxiliary responses are of lower quality.
- In AlpacaEval 2.0, many LLMs improve their LC win rates significantly when referencing other models' outputs.
- This observation forms the basis for the Mixture-of-Agents methodology.

Mixture-of-Agents (MoA) Framework

- MoA involves multiple LLMs in each layer, where each agent in layer i takes inputs from agents in layer $i - 1$.
- Refinement process continues iteratively until the final response is achieved.
- The final layer uses an aggregator agent to integrate and synthesize responses, aiming for high-quality output.

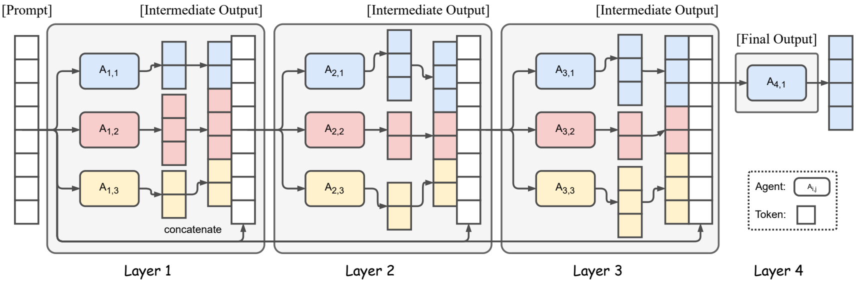

Figure: Mixture-of-Agents Structure with 4 layers and 3 agents in each layer.

Performance Metrics and Criteria

- Selection of LLMs for each MoA layer is based on:
  - **Performance Metrics**: Average win rate of models in layer i.
  - **Diversity Considerations**: Heterogeneous model outputs contribute more than homogeneous ones.
- Careful selection mitigates individual model deficiencies and enhances overall response quality.

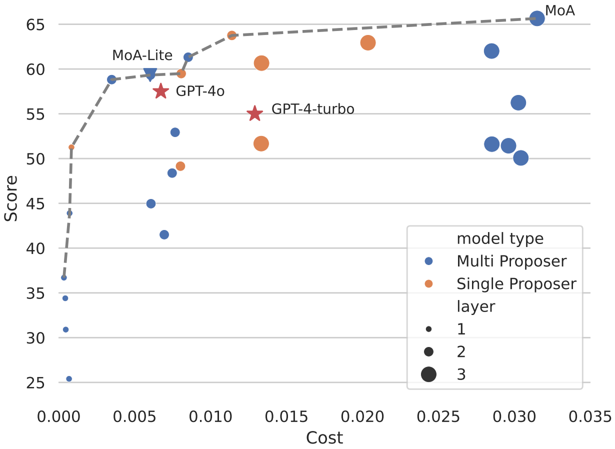

Figure: Performance trade-off versus cost.

Evaluation Setup

- Benchmarks: AlpacaEval 2.0, MT-Bench, and FLASK.
- Models included: Qwen1.5-110B-Chat, Qwen1.5-72B-Chat, WizardLM-8x22B, LLaMA-3-70B-Instruct, Mixtral-8x22B-v0.1, and dbrx-instruct.
- MoA configurations:
  - MoA (open-source models only).
  - MoA w/ GPT-4o (uses GPT-4o as the aggregator in the final layer).
  - MoA-Lite (more cost-effective, with fewer layers and a smaller aggregator).

Benchmark Results: AlpacaEval 2.0

- MoA achieved a remarkable 65.1% LC win rate, outperforming GPT-4 Omni's 57.5%.
- MoA w/ GPT-4o achieved a 65.7% LC win rate, significantly better than GPT-4o alone.
- Even the cost-effective MoA-Lite variant achieved a 59.3% LC win rate, surpassing both GPT-4 Turbo and GPT-4 Omni.

Benchmark Results: MT-Bench

- MoA w/ GPT-4o scored an average of 9.40, outperforming GPT-4 Turbo which scored 9.31.
- MoA (open-source) scored 9.25 on average, maintaining competitive performance.
- MoA-Lite scored 9.18, still above many state-of-the-art models.
- Improvements are incremental but MoA consistently ranked at the top.

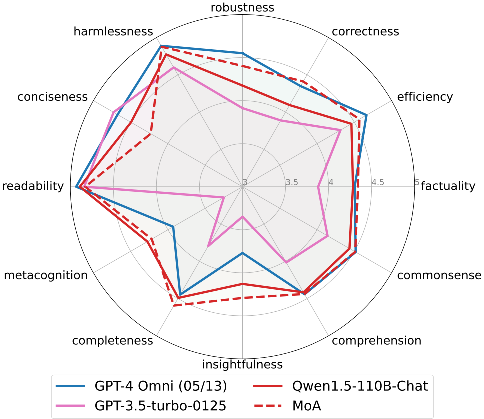

Figure: Benchmark results from FLASK.

Budget and Token Analysis

- Cost-effectiveness: MoA-Lite matches GPT-4o's cost while achieving a higher quality.
- Latency: MoA configurations provide better performance for the same level of computational resources.
- Performance trade-offs: MoA lies on the Pareto front, balancing quality and resource efficiency.

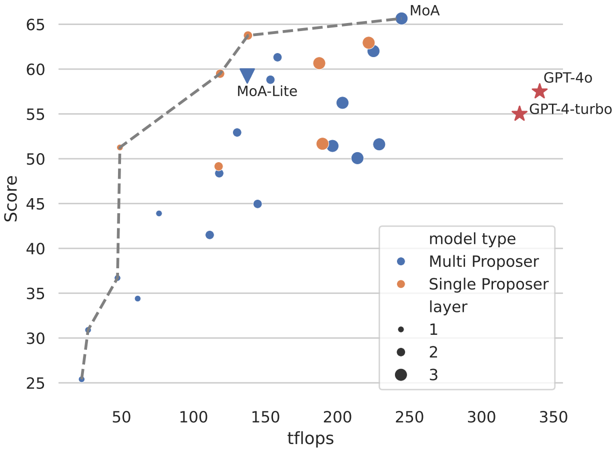

Figure: Performance trade-off vs. tflops.

Conclusion and Future Directions

- MoA leverages collaborative capabilities of LLMs to improve response quality.
- Achieved state-of-the-art performance across multiple benchmarks, even with open-source models.
- Future work could focus on optimizing MoA architecture and chunk-wise aggregation to reduce latency.
- The approach adds interpretability to models, facilitating human-aligned reasoning.