Team SocrAItic Circle

AIDebater: Training LLMs to argue and learn

Gaurav Kothamachu Harish
Pengkun Ma
Joel Markapudi
Tanay Grover
Easha Meher Koppisetty

The Idea

- Two Large Language Models (LLMs) simulate opposing sides of a debate.

- A "judge" (human or AI) evaluates their arguments based on logical consistency, rhetorical strength, and factual accuracy.

- Feedback from the judge allows debaters to refine their arguments in an iterative process.

- The system integrates human participants into the loop, enabling both humans and AI to improve their debating skills.

Our Goal: what we aim to achieve

- Create a dynamic, feedback-driven debate training system.
- Combine AI and human input to enhance critical thinking and argumentation.
- Provide a platform for iterative skill refinement in debating for both AI agents and human participants.

Importance of our project

Why It’s Interesting:

- Explores how AI can simulate diverse perspectives effectively.
- Demonstrates the potential for AI-human collaboration in education.
- Encourages active learning through iterative improvement.

Why It’s Important:

- Fosters critical thinking and logical reasoning skills in human participants.
- Advances AI’s ability to understand and refine arguments.
- Builds trust in AI systems by integrating humans into the feedback loop.

What Makes It Hard:

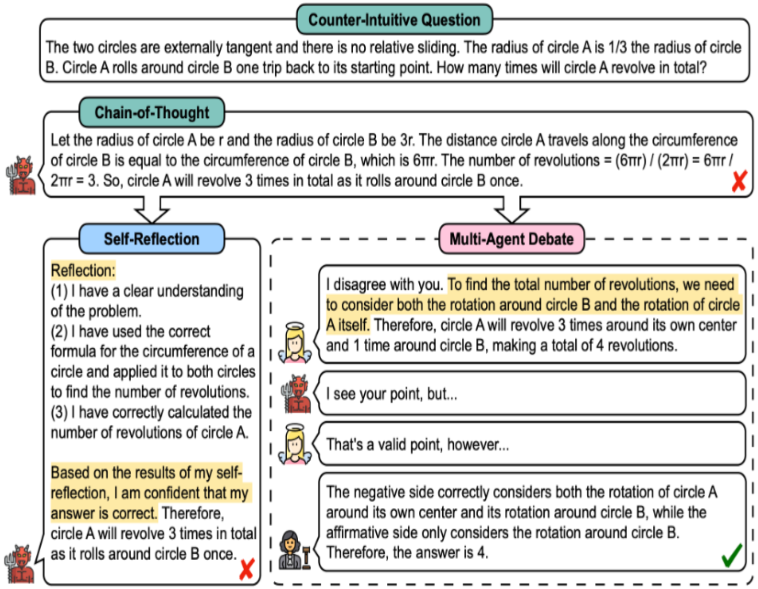
- Ensuring argument quality (logical, rhetorical, factual).
- Designing an effective feedback mechanism.
- Balancing AI and human roles for seamless interaction.

Related Works

Multi-Agent Debate (MAD) for LLMs

Problem: Degeneration of Thoughts (DoT) in Self-Reflection

- Self-reflection in LLMs can lead to:
  - **Bias & Distorted Perception** – Reinforcing incorrect beliefs.
  - **Rigidity & Resistance to Change** – Lack of adaptability.
  - **Limited External Feedback** – Missing alternative viewpoints.



Solution: Multi-Agent Debate (MAD)

- Two AI models debate to challenge and correct each other.
- Reduces bias by exposing flaws in reasoning.
- Encourages dynamic learning through mutual feedback.

CoEvol Framework

What is CoEvol?

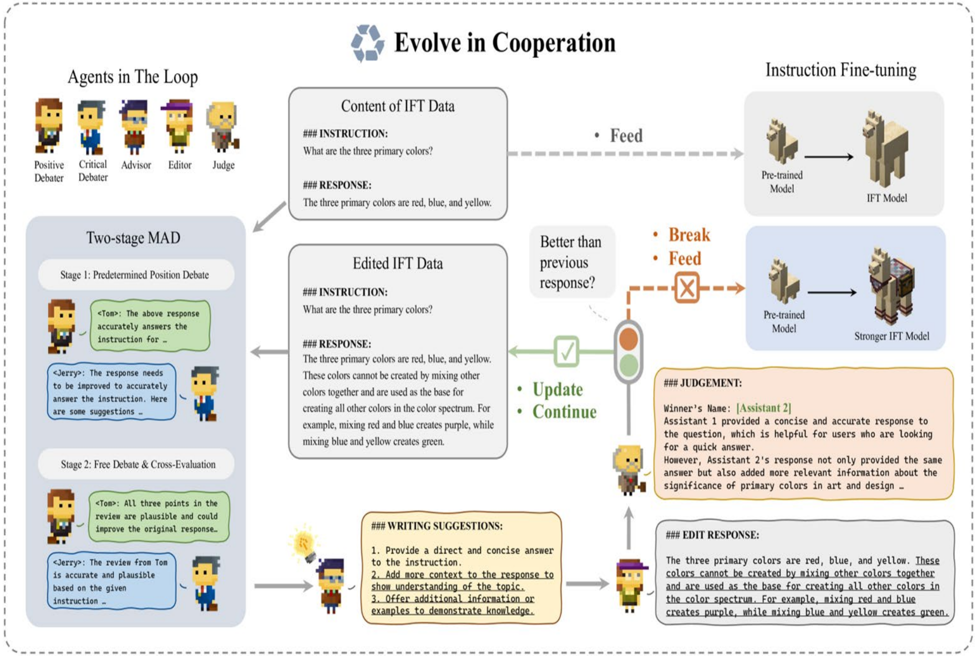
- A **multi-agent framework** to refine LLM-generated responses.
- Uses **Debate-Advise-Edit-Judge (DAEJ) paradigm** for iterative improvement

How It Works

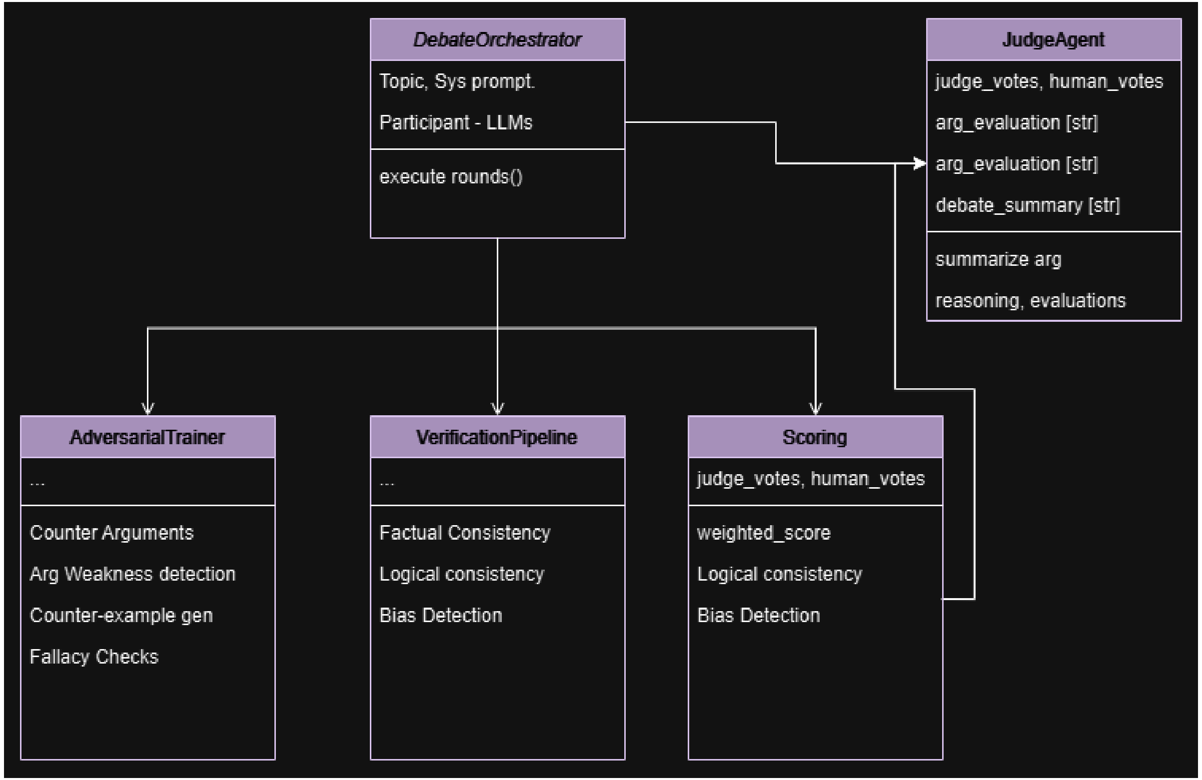
1. **Debate:** Two AI agents argue over the response's accuracy.
2. **Advise:** An advisor suggests refinements based on the debate.
3. **Edit:** An AI editor improves the response accordingly.
4. **Judge:** A separate AI evaluates if the new response is better.

Why It Matters?

- Improves instruction fine-tuning (IFT) data.
- Enhances LLM response quality through AI collaboration.
- Uses a structured debate approach to refine reasoning.



Design - Initial Ideas



- **Choosing API requests, responses** for debate, judging, and scoring.
- Hybrid Scoring System.
- Adversarial Training.
- Multi-LLM agents as Judge.
- Debate summarization, Argument analysis.
- Logical, Factual Consistency Analysis.
- Fallacy analysis.

Primary Hypotheses

H1: Multi-Judge Consensus Hypothesis

"A debate evaluation system using multiple LLM judges with diverse prompting strategies produces more reliable and consistent evaluations compared to single-judge systems"

Rationale: Multiple perspectives and evaluation approaches should reduce bias and increase evaluation reliability

H2: Adversarial Improvement Hypothesis

"Debate agents exposed to adversarial challenges during debates demonstrate improved argument quality and reduced logical fallacies in subsequent debates"

Rationale: The process of defending against and responding to challenges should strengthen argumentation skills

Fine Tuning For Adversarial Training

- Identify logical leaps, Find unstated assumptions, Question causal relationships.
- Request source, Challenge data interpretation, Identify cherry-picked examples.
- Present edge cases, Provide contradicting scenarios, Demonstrate exceptions.

Core things to attack: Logical reasoning patterns, Common fallacies, Argument structures, Academic and debate principles.

Design - Strategic Prompting Layers

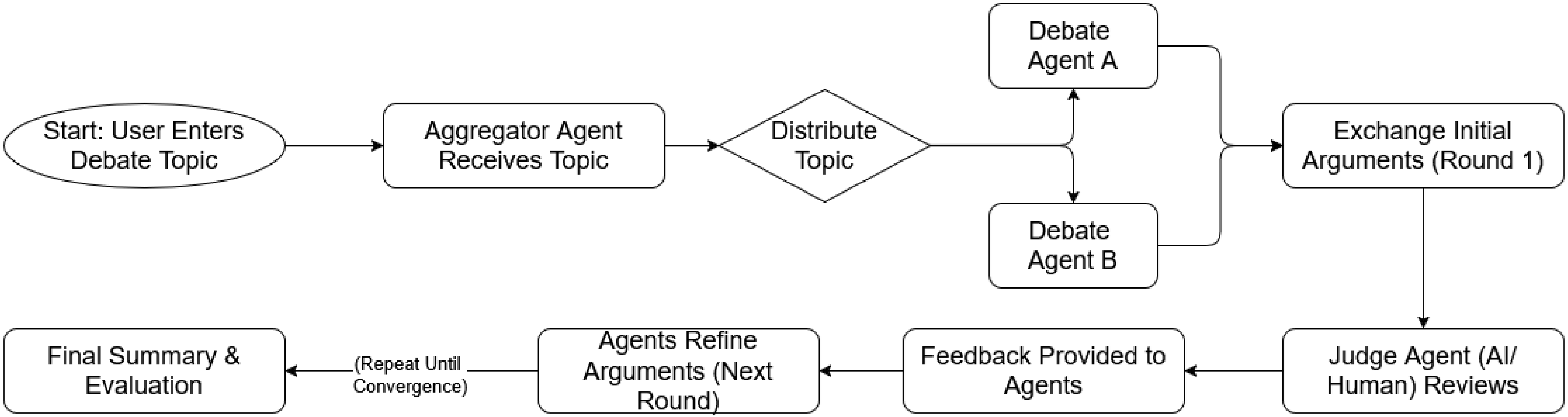
```
# Strategic prompting layers
analysis_layers = [
    {
        "focus": "Evidence Analysis",
        "prompt": "What specific claims are made without sufficient evidence?
                   For each claim, explain what type of evidence would strengthen it."
    },
    {
        "focus": "Logical Flow",
        "prompt": "Trace the logical steps in this argument. Identify any gaps or jumps in reasoning."
    },
    {
        "focus": "Assumption Check",
        "prompt": "What unstated assumptions must be true for this argument to work?
                   Which of these assumptions might be questionable?"
    }
]
```

- Each layer is a dictionary with a specific focus and prompt template.
- Templates are structured to force the LLM to analyze one specific aspect.

Alternative approaches

Approach	Pros	Pros & Cons
Pure LLM-based	Both the agents and Judge will be a custom trained model	Faster execution, consistent behaviour. Limited creativity, potential for bias. Harder to benchmark.
Hybrid Human-AI	Human would provide feedback like how arguments would be judged in professional judge competitions	More nuanced feedback, better quality. Slower, requires human availability.
Permanently train agents	Ensure previous feedback from older debates are used to generate responses	Ensures model’s common weaknesses are addressed, Needs RAG.

Architecture



Development Stages

Start

Initial Setup and Model Integration

Set up development environment with Python
Install core libraries

Configure API access for GPT-4 and other LLMs

Feb 17

Multi-Agent Framework Development

Implement two debater agents which interacts with different models, stores context and passes feedback
Create judge agent with feedback mechanisms
Set up basic logging and monitoring

Mar 10

Feedback Loop Implementation

Design iterative refinement system
Implement feedback processing

Create evaluation metrics

Test system robustness

Mar 24

Testing

Conduct initial testing
Refine based on feedback
Do performance analysis and provide baseline compared to metrics

Apr 10

Required Libraries/APIs

- Langchain, LangGraph, Autogen
- OpenAI/Mistral/Claude etc. APIs
- sentence-transformers, faiss-cpu, huggingface_hub, tiktoken, tenacity
- Any UI & visualization packages, orchestration, chat interface utilities.

Challenges

- Maintaining context across multiple debate rounds.
- Ensuring logical consistency in arguments.
- Implementing effective feedback mechanisms.
- Managing computational resources/tokens for models.
- Handling edge cases.

(Suggested) Measurable Metrics

- ‘Quality’ measurement when comparing text outputs from LLMs could be an ambiguous affair. Instead, we depend on a third-party models/frameworks to evaluate and critique the answer.
- Weighted judge scores, Categorized scoring (reliability, argument strength, no fallacies, etc.).
- Percentage of unanimous decisions.
- What seems to be the knowledge deviation among judges.

(Suggested) Measurable Metrics continued….

Baseline Comparisons

Comparison against vanilla LLM's

Comparison against
OpenCaselist (human debates)

Comparison against
Multi-Agents-Debate framework

Argumentation Strength Metrics

Factual Accuracy(TruthfulQA model)

Logical Consistency

Persuasiveness

Human Feedback Metrics

Compare AI-generated ratings
with human judgments.

Debate Flow Metrics

Response Refinement

Debate convergement time