Feb 16, 2025

LLM Agents Study Session: Lecture 3

Summary

Arunesh Mishra presented lecture three, covering memory, reasoning, and planning in LLMs, highlighting differences between "LLM-first" and "agent-first" approaches and introducing HippoRag for improved long-term memory; Shristi Gautam contributed their experience with a similar system. The lecture also explored implicit reasoning, grokking, and planning challenges in web agents, with a focus on the Web Dreamer project's goal of creating reliable, complex web task execution; Yatharth Piplani contributed to the discussion on backtracking strategies in LLMs, although connection issues hampered participation.

Details

- **Lecture Three Overview:** Arunesh Mishra presented lecture three, focusing on memory, reasoning, and planning in LLM agents. They discussed Professor Yuzu's definition of an agent as something perceiving and acting on its environment, a definition still relevant for LLM-based agents. The lecture explored the differences between the "LLM-first" and "agent-first" views of LLM agent development. Key aspects included instruction following, in-context learning, and reasoning capabilities.
- **LLM Agents vs. Language Agents:** Arunesh Mishra explained the distinction between LLM agents (built upon existing LLMs) and language agents (a broader category encompassing multimodal interaction through language). They emphasized the expressiveness and flexibility of language-based interaction in agents.
- **HippoRag for Long-Term Memory:** The lecture covered HippoRag, a system inspired by human memory storage, aiming to improve the efficiency of non-parametric memory (RAG). Arunesh Mishra highlighted its ability to override

conflicting internal memory with external evidence, addressing limitations of current RAG systems that struggle with complex queries requiring multi-hop connections between information. Shristi Gautam shared their experience implementing a similar system using lightweight page ontologies. The system utilizes offline indexing and graph traversal for efficient retrieval.

- **Implicit Reasoning and Grokking:** The discussion explored implicit reasoning in LLMs, contrasting it with explicit chain-of-thought methods. Arunesh Mishra explained that LLMs struggle with implicit reasoning, particularly composition and comparison. They introduced the concept of "grokking," a phenomenon where models initially overfit but later generalize well after extended training. They described how grokking improves generalization, particularly for comparison reasoning, more than for composition. The use of logit lens and causal tracing helped analyze this process.
- **Planning and World Models:** The lecture concluded with a discussion on planning in LLM agents, emphasizing the increased expressiveness of goal specification and the challenges of managing open-ended action spaces, especially in web-based applications. Arunesh Mishra noted the difficulty of automating the process of evaluating action outcomes.
- **Web Agent Planning:** Arunesh Mishra discussed challenges in web agent planning, comparing a greedy, short-sighted approach (fast but inaccurate) with tree search (slow but potentially more accurate). They proposed using LLMs to create a world model, simulating actions and predicting outcomes to guide planning. This involves simulating actions, assigning scores, and choosing the highest-scoring path. While showing promising results compared to reactive approaches, the system doesn't match the performance of tree search but is more practical for web tasks. The use of natural language for goal specification expands the range of solvable problems. Improving LLMs for planning remains an open area of research.
- **LLM-Based Web Agent Implementation:** Arunesh Mishra described using LLMs to process HTML, identify possible actions (clicking links, entering text), and predict outcomes. They suggested a two-stage LLM approach: one for simulating actions and another for scoring potential paths. The agent needs access to the entire website to fully understand the action space. The agent autonomously takes actions based on the simulated outcomes, iterating through simulation and execution stages. The system's applications extend beyond web tasks, potentially including server monitoring.

- **Web Dreamer Project Goals:** The goal of the Web Dreamer project is to create an agent capable of reliably executing complex web tasks, such as finding products meeting specific criteria. This includes handling irreversible actions and unexpected events. The approach is generalizable to non-web applications, with potential for domain-specific fine-tuning. Challenges remain regarding performance, scalability, and adversarial attacks. Improving success rate and the number of steps per task are key areas for future development.
- **Continuous Learning and Reasoning:** Arunesh Mishra discussed the limitations of parametric learning in LLMs due to catastrophic forgetting and introduced HIPPO RAG as a solution. HIPPO RAG uses a graph-like data structure to link knowledge nodes, enabling more effective knowledge retrieval than embedding-based RAG. The system employs LLMs for knowledge extraction and graph traversal, potentially augmenting existing embedding-based databases.
- **Reinforcement Learning for Reasoning:** Arunesh Mishra explained the role of reinforcement learning (RL) in enhancing LLM reasoning capabilities. Unlike traditional LLM training, RL focuses on rewarding actions (thought processes) that lead to correct answers. This approach uses a pre-trained LLM as a basis, adding reasoning capabilities by rewarding successful thought processes and penalizing incorrect ones. The objective function maximizes rewards, employing gradient ascent. The reasoning strategies are learned implicitly within the embedding space, and the system's ability to reason and reflect emerges naturally.
- **RL Fine-tuning Details:** The RL fine-tuning process involves generating multiple completions for a given prompt, assigning rewards based on correctness and reasoning quality. The model does not maintain a history of previous actions; each prompt is treated independently. The reward information is used to update the model's parameters (θ), favoring outputs with better reasoning and higher accuracy. The resulting policy is embedded within the LLM itself. The process involves iterative improvement through reinforcement learning, with the ability to take snapshots of the model at various stages. The learning happens during the fine-tuning phase; it is not an online learning process.
- **Backtracking Strategies in Large Language Models (LLMs)** Yatharth Piplani and Arunesh Mishra discussed how backtracking strategies develop naturally in LLMs due to vocabulary size and inherent properties. They hypothesized that LLMs learn these strategies from the implicit reasoning present in human-generated text, with reinforcement learning (RL) optimizing their use.

Arunesh Mishra suggested that smaller models might not be suitable for adding reasoning capabilities because they lack sufficient exposure to reasoning data.

- **Reasoning Capabilities and Model Size** Arunesh Mishra posited that the effectiveness of reasoning strategies in LLMs depends on the model's size and training data. They explained that larger models like Llama 1.5B, having been exposed to more reasoning data, are more likely to exhibit such capabilities than smaller models. They also noted that the connection between Yatharth Piplani was spotty during the discussion.
- **Meeting Conclusion** The meeting concluded with thanks from JT S to all participants, and a suggestion from Arunesh Mishra to conduct a future session demonstrating the reasoning process more closely. They also noted the difficulty in hearing Yatharth Piplani due to connection issues.