From Static to Dynamic: Knowledge Metabolism for Large Language Models

Mingzhe Du[1,2], **Anh Tuan Luu**[1], **Bin Ji**[2], **See-Kiong Ng**[2]

[1] Nanyang Technological University
[2] National University of Singapore
{mingzhe001, anhtuan.luu}@ntu.edu.sg, {jibin, seekiong}@nus.edu.sg

Abstract

The immense parameter space of Large Language Models (LLMs) endows them with superior knowledge retention capabilities, allowing them to excel in a variety of natural language processing tasks. However, it also instigates difficulties in consistently tuning LLMs to incorporate the most recent knowledge, which may further lead LLMs to produce inaccurate and fabricated content. To alleviate this issue, we propose a knowledge metabolism framework for LLMs, which proactively sustains the credibility of knowledge through an auxiliary memory component and directly delivers pertinent knowledge for LLM inference, thereby suppressing hallucinations caused by obsolete internal knowledge during the LLM inference process. Benchmark experiments demonstrate DynaMind's effectiveness in overcoming this challenge. The code and demo of DynaMind are available at: https://github.com/Elfsong/DynaMind.

Introduction

The advent of Large Language Models (LLMs) signifies an unprecedented transformation in the field of artificial intelligence, demonstrating remarkable proficiency across a diverse spectrum of tasks (Brown et al. 2020; OpenAI 2023; Touvron et al. 2023; Penedo et al. 2023). With the unique abilities to assimilate knowledge in their parameters and solve complex queries, LLMs have significantly recontextualized the boundary of artificial intelligence. However, akin to any technological advancement, LLMs come bundled with intrinsic limitations.

One primary constraint lies in their static nature (Kemker et al. 2018). Since knowledge encapsulated in LLMs is defined by the parameters at the training stage, tunning pretrained LLMs to incorporate new knowledge is resource-intensive and prone to catastrophic forgetting (Scao et al. 2022). Moreover, the ambiguous LLM knowledge could potentially yield unfounded outputs, thereby compromising their reliability (Azamfirei, Kudchadkar, and Fackler 2023). These challenges obstruct LLMs from assimilating new knowledge and adapting to evolving scenarios.

Emerging studies explored enhancing LLMs through external components. AutoGPT (Significant-Gravitas 2023) aims to autonomously propel the "Chain-of-Thought" toward achieving designated objectives. Despite exhibiting the potential to execute elementary tasks, its applicability is substantially hampered in complex scenarios due to the absence of long-term memorization and task management capabilities. Furthermore, BabyAGI (Yoheinakajima 2023) integrates task prioritization and dense vector retrieval to enhance inference context. Nonetheless, it cannot discern and replace outdated knowledge stored in its memory. In contrast, Langchain (Chase 2022) offers an LLM development toolkit instead of a specific solution.

To mitigate the above challenges, we present DynaMind, an innovative continual learning framework for LLMs. Infused with a standalone memory module, it takes inspiration from attributes of human learning (Eichenbaum 2004; Hadsell et al. 2020), allowing LLMs to administer knowledge without modifying model parameters. As substantiated by our investigation, DynaMind markedly enhances LLMs' ability for continual learning.

DynaMind

In this section, we will delve into the architecture and workflow of DynaMind. As illustrated in Figure 1, the intellectual core, *Inference Engine* retrieves the knowledge required for decision-making from *Memory Manager* based on contexts and then generates subsequent instructions, which will be executed by specific *Operators*.

Inference Engine

Inference Engine is powered by one or multiple LLMs that excel in natural language reasoning. As depicted in Figure 2, *Inference Engine* integrates context into meticulously structured templates and recursively produces a First In First Out (FIFO) *Operator* command queue, which empowers the engine to deconstruct intricate queries into a set of simpler subproblems. The LLM employed can vary based on the varying capabilities of the models. Specifically, while a small-scale LLM may suffice for text summarization, a large-scale LLM may be essential for logical reasoning. Thus, the selection of appropriate LLM significantly enhances both the speed and performance of the inference process. Currently, DynaMind supports LLMs such as OpenAI GPT-3.5 (Brown et al. 2020), GPT-4 (OpenAI 2023), LLaMA (Touvron et al. 2023), and Falcon (Penedo et al. 2023).

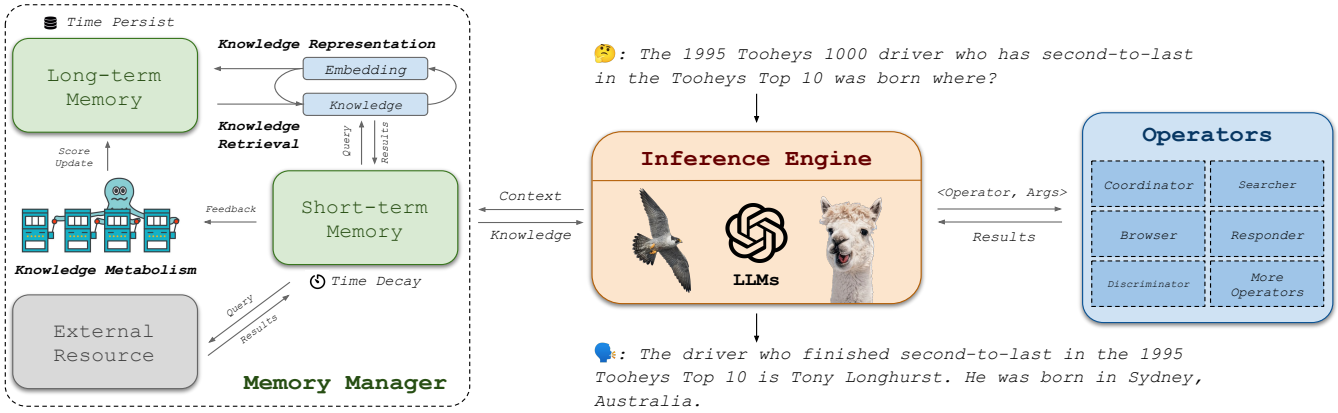

Figure 1: The system overview of DynaMind.

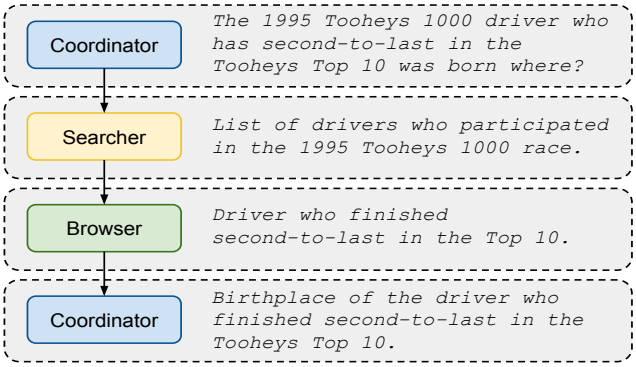

Figure 2: DynaMind Pipeline Example

Model Name	Baseline	DynaMind
OpenAI GPT-3.5	8.5	89.0 (+80.5)
OpenAI GPT-4	16.0	92.5 (+76.5)
Falcon-40B	17.5	85.0 (+67.5)
Llama-30B	6.0	56.5 (+50.5)

Table 1: Accuracy on Knowledge-driven Reasoning.

Model Name	Create	Update	Delete
OpenAI GPT-3.5	92.0	81.0	71.5
OpenAI GPT-4	95.5	85.0	71.5
Falcon-40B	90.5	83.5	74.0
Llama-30B	88.0	78.5	70.0

Table 2: Performance on Knowledge Manipulation.

Memory Manager

Memory Manager is a composite of four interconnected modules: (1) *Knowledge Representation* encodes knowledge for efficient processing using embeddings generated by LLMs. (2) *Knowledge Retrieval* then conducts vector searches for prompt identification and retrieval of relevant knowledge. (3) *Long-term memory* and (4) *Short-term memory* uphold and recall knowledge from various sources, with the former maintaining a permanent repository and the latter providing immediate information for the current session. Inspired by the Multi-Armed Bandit algorithm (Slivkins et al. 2019), *Knowledge Metabolism* adaptively changes the credibility of the held knowledge based on contextual conditions.

Operators

DynaMind leverages a suite of operators to coordinate tasks and solve complex queries. The *Coordinator* breaks down queries into sub-problems, which are then managed by other operators. The *Searcher* offers both keyword and vector-based searches, providing context-related knowledge for the Inference Engine and aiding in data control and knowledge retrieval. The *Browser* operator is engaged for reading and parsing files. When resources surpass the token limit, they will be summarized, and any irrelevant details will be discarded. Subsequently, *Responder* generates the tentative responses, and *Discriminator* will evaluate whether the response satisfies the query objective. If the response is qualified, *Discriminator* will adjust the credibility of each involved knowledge in the long-term memory based on its contribution to the response, promoting highly credible knowledge during retrieval while discounting less credible knowledge. Otherwise, *Coordinator* will regenerate the command queue to produce an improved response.

Benchmarks and Evaluations

We construct two benchmark datasets regarding continual learning to evaluate the DynaMind framework.[1] In the **Knowledge-driven Complex Reasoning** benchmark, DynaMind counteracts the LLM hallucination issue during reasoning complex tasks by incorporating external memory capabilities. Results indicate substantial improvement in performance when utilizing DynaMind, particularly when applied to OpenAI GPT-4. Regarding the **Knowledge Manipulation** benchmark, it delineates the capacity of DynaMind to explicitly manage knowledge in LLMs. Overall, DynaMind demonstrates commendable results, highlighting its efficacy for continual learning.

[1]Experiment details can be found in the demo.

Acknowledgments

This research is supported by A*STAR, CISCO Systems (USA) Pte. Ltd and National University of Singapore under its Cisco-NUS Accelerated Digital Economy Corporate Laboratory (Award I21001E0002).

References

Azamfirei, R.; Kudchadkar, S. R.; and Fackler, J. 2023. Large language models and the perils of their hallucinations. *Critical Care*, 27(1): 1–2.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chase, H. 2022. LangChain. https://github.com/hwchase17/langchain. Accessed: 2023-12-09.

Eichenbaum, H. 2004. Hippocampus: cognitive processes and neural representations that underlie declarative memory. *Neuron*, 44(1): 109–120.

Hadsell, R.; Rao, D.; Rusu, A. A.; and Pascanu, R. 2020. Embracing change: Continual learning in deep neural networks. *Trends in cognitive sciences*, 24(12): 1028–1040.

Kemker, R.; McClure, M.; Abitino, A.; Hayes, T.; and Kanan, C. 2018. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

OpenAI. 2023. GPT-4 Technical Report. *arXiv:2303.08774 [cs]*, 1.

Penedo, G.; Malartic, Q.; Hesslow, D.; Cojocaru, R.; Cappelli, A.; Alobeidli, H.; Pannier, B.; Almazrouei, E.; and Launay, J. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Scao, T. L.; Wang, T.; Hesslow, D.; Saulnier, L.; Bekman, S.; Bari, M. S.; Bideman, S.; Elsahar, H.; Muennighoff, N.; Phang, J.; et al. 2022. What language model to train if you have one million GPU hours? *arXiv preprint arXiv:2210.15424*.

Significant-Gravitas. 2023. Significant-gravitas/auto-GPT: An experimental open-source attempt to make GPT-4 fully autonomous. https://github.com/Significant-Gravitas/Auto-GPT. Accessed: 2023-12-09.

Slivkins, A.; et al. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2): 1–286.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yoheinakajima. 2023. Yoheinakajima/Babyagi.