ToolkenGPT:
Augmenting frozen language models with massive tools via tool embeddings

Authors:
Shibo Hao, Tianyang Liu, Zhen Wang, Zhiting Hu

Project by:
Harshit T, Krithika I, Mrinaal D, Siddhant L

Contents

Background - ToolkenGPT

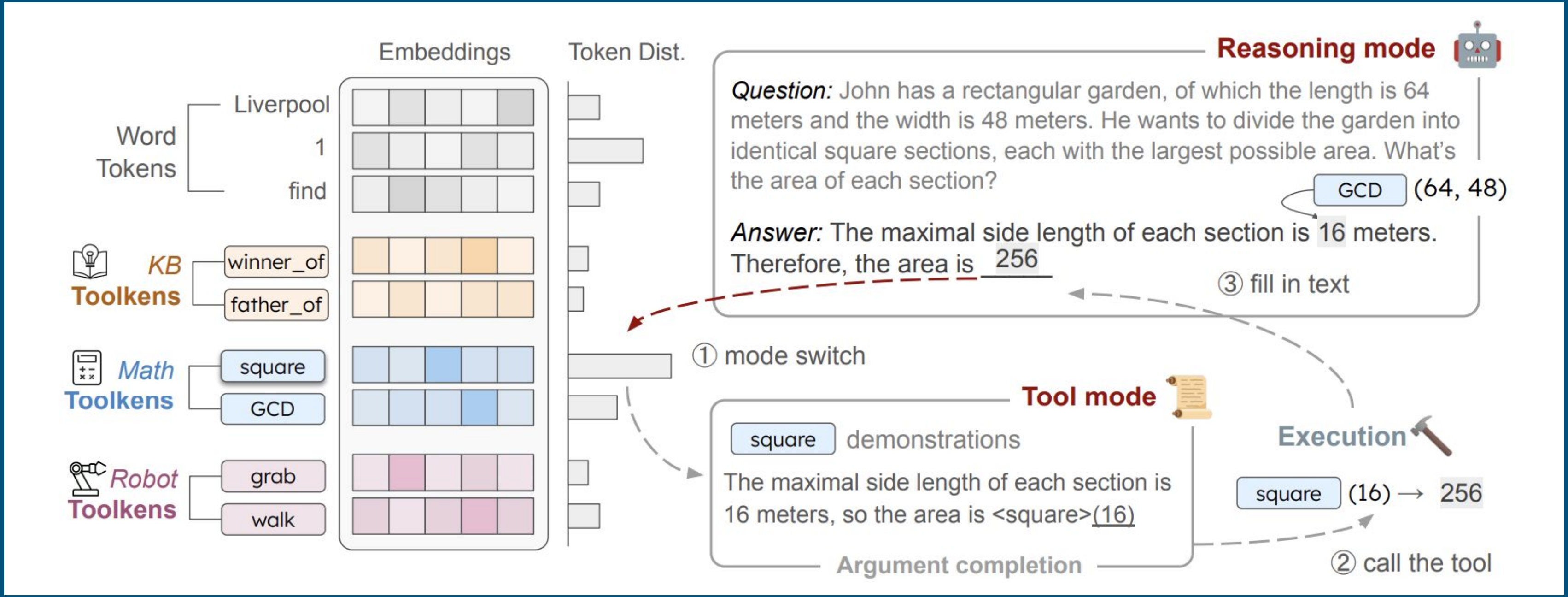
How can we augment LLMs with external tools efficiently?

Two existing approaches

1. Fine-tuning Approach:
    - Lacks flexibility to adapt to emerging or updated tools
    - Computationally costly and resource-intensive

2. In-context Learning Approach:
    - Restricted by context length limitations
    - Leads to suboptimal understanding of tools

Background - ToolkenGPT Fwk Overview



Background - ToolkenGPT Fwk Overview

- Represent tools as special tokens
  - Hence the name ***toolken*** (tool + token)
  - Each tool API is assigned unique token embedding outside LLMs vocabulary
  - Toolken embeddings are predicted alongside regular word token embeddings
- Framework Operates in two modes:
  - Reasoning Mode
  - Tool Mode

Background - Benefits of ToolkenGPT

- Enhancing LLM capabilities
  - Integrate external tools with LLMs.
  - Enables dynamic tool use without retraining.
- Scalability
  - Minimal GPU memory overhead
- Adaptability
  - Quickly adapts to new tools without expensive retraining

Research Questions

How can ToolkenGPT framework enhance the performance and adaptability of smaller and more recent LLMs?

How does task synergy in multi-task learning, especially between computational reasoning and knowledge-based tasks, affect efficiency and accuracy of ToolkenGPT framework?

Problem Statement

- The goal is to enhance LLMs with external tools, improving task performance without retraining
- Explore ToolkenGPT's effectiveness with smaller and more recent models like Llama-3.2
- Extend ToolkenGPT with multi-task training and analyze task combinations

Methodology

- Transition from deprecated Llama libraries to Hugging Face
- Transition from Llama-1 13B/30B to Llama-3.2 1B
- Update original source code to ensure compatibility with Llama-3.2
- Re-annotate dataset with Llama-3.2 tokenizer
- (*In-progress*) Fix errors in Inference pipeline
- (*In-progress*) Update the ToolkenGPT framework for multi-task learning

Experiments

- Explored other variants of Llama models before settling in with Llama-3.2 1B
- Explored dataset re-annotation strategies to make the dataset generic
- Explored quantization for Llama-3.2 but continued with full precision due to technical challenges
- Working datasets:
    - GSM8k-XL
    - FuncQA
    - KAMEL

Results

- Llama-3.2 1B model being trained to compare results with original ToolkenGPT results.
- Preliminary training results

	Precision	**Recall**	**F1**
GSM8K-XL	0.8777	0.9352	0.9055
FuncQA	0.7272	0.7999	0.7619
KAMEL (sup)	0.8649	0.89	0.8773
KAMEL (syn)	0.4775	0.6589	0.5537

Challenges

- Original source code tightly coupled with Llama-1 and deprecated libraries
- Errors in synthetic datasets
- Dataset re-annotation
- Differences in tokenization of special tokens between Llama-1 and Llama-3.2

Conclusion

- Adapted ToolkenGPT framework for Llama-3.2 1B, tackling budget and resource limitations
- Highlights feasibility and challenges of experimenting with huge LLMs
- Preliminary tests yielded promising results

Next Steps

- (*In-progress*) Fix errors in Inference pipeline
- (*In-progress*) Update the ToolkenGPT framework for multi-task learning
- Compare baseline for individual task training with original ToolkenGPT results
- Compare joint-task training (e.g., numerical reasoning and QA) against individual task trainings to evaluate performance changes

References

[1] S. Hao, T. Liu, Z. Wang, and Z. Hu, “Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings,” 2024.
[2] Meta AI, “Llama 3.2: Text-Only model.” https://huggingface.co/meta-llama/ Llama-3.2-1B, 2024. Last Accessed: Nov 11, 2024.
[3] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozi.re, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, “Llama: Open and efficient foundation language models,” 2023.
[4] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman, “Training verifiers to solve math word problems,” ArXiv, vol. abs/2110.14168, 2021.
[5] J.-C. Kalo and L. Fichtel, “Kamel: Knowledge analysis with multitoken entities in language models,” in Automated Knowledge Base Construction, 2022.

Thank you!

Q & A