Beluga: Boosted Explanatory Learning Using GPT Assistance

1st Kendre, Aditya
College of Engineering
Penn State University
PA, USA
axk6052@psu.edu

2nd Nguyen, Hien
School of Science, Engineering, and Technology
Penn State University
PA, USA
nguyen.hien@psu.edu

3rd Saha, Suman
School of Electrical Engineering and Computer Science
Penn State University
PA, USA
szs339@psu.edu

4th Kabir, Faisal
School of Science, Engineering, and Technology
Penn State University
PA, USA
mpk5904@psu.edu

***Abstract*—This paper introduces Beluga, a novel approach to boosting explanatory learning in large language models (LLMs) using GPT assistance. We propose a method that combines boosted explanation tuning, knowledge distillation, and a focus on deployability to create a more efficient and practical model. By creating a non-linear thought graph for better logic extraction and distilling knowledge from larger teacher models into a smaller Pythia-1B model, Beluga aims to improve performance while maintaining a compact size suitable for resource-constrained devices. We highlight the advantages of deploying LLMs closer to consumers, including decreased latency, enhanced privacy, and offline functionality. The key contribution includes an improved explanatory dataset, an efficient knowledge distillation process, and a focus on deployability for resource-constrained environments.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Large Language Models (LLMs) have made significant breakthroughs in natural language processing (NLP) tasks, such as language translation and text generation [12]. These models, powered by advanced deep learning techniques, have revolutionized the field of NLP and opened up new possibilities for understanding and generating human-like text. By leveraging massive amounts of data and complex algorithms, LLMs have achieved unprecedented levels of performance in various language-related tasks, captivating researchers and industry professionals alike.

The emergence of LLMs has been driven by the exponential growth of computational power and the availability of vast quantities of textual data. These models, characterized by their vast size and complex architectures, are trained on enormous datasets, which allow them to capture intricate patterns and nuances in language. As a result, LLMs can understand and generate text that is remarkably coherent, contextually appropriate, and even creative [24].

These models typically consist of millions and billions of parameters, making them computationally intensive and necessitating specialized hardware infrastructure [12]. To address the challenges posed by the large size and computational requirements of LLMs, some researchers have been focusing on the technique of knowledge distillation [21]. This approach aims to distill information and insights from larger models into smaller ones while maintaining similar performance levels.

Knowledge distillation involves training a smaller model, often referred to as a student model, to mimic the behavior and knowledge of a larger, more complex model known as the teacher model. The teacher model acts as a knowledgeable guide, providing supervisory signals to the student model during training. By leveraging the knowledge and representations learned by the teacher model, the student model can achieve comparable performance to the larger model, even with limited computational resources [14]. Using this concept, it is possible to make LLM technology more accessible and deployable in resource-constrained environments [9]. By distilling the knowledge from larger models into smaller ones, researchers can bridge the gap between the immense power of LLMs and the practical limitations imposed by hardware constraints [11].

Deployment Application: Deploying LLMs closer to the consumer, both in terms of hardware requirements and deployment strategies, offers several notable advantages. Local models, situated directly on user devices or within localized infrastructure, present positive outcomes such as decreased latency and enhanced privacy. By having the LLM reside closer to the consumer, there is a substantial reduction in the time it takes for data to travel back and forth between the user's device and a remote server. This decreased latency translates into quicker and more seamless interactions, enabling near real-time language processing and significantly enhancing the user experience. Moreover, local models can contribute to improved privacy protection. By keeping the data processing and analysis on the user's device or within localized infrastructure, the need to transmit sensitive information to external servers is minimized. This mitigates potential privacy risks associated

with data transfer and storage, as the user maintains greater control over their personal information. Local models allow for on-device data processing, preserving user privacy and potentially reducing the exposure of sensitive data to external entities. Additionally, local models can operate effectively in offline or low-connectivity scenarios, where internet access may be limited or unreliable. By having an LLM installed directly on the user's device, individuals can continue to utilize language processing capabilities even without a stable internet connection. This offline functionality empowers users to access and leverage LLM technology in various contexts, regardless of their internet connectivity status.

Key Contributions:

- **Boosted Explanation Tuning**: We improve current explanatory datasets by creating a non-linear thought graph to better extract logic. This approach goes beyond instruction tuning and chain-of-thought prompting, by including detailed explanations of possible solution paths. In this manner the student model gets more insights into the teacher model's reasoning process. This can help the student model mimic the thought process of the teacher model, leading to improved performance and understanding.
- **Distilling Knowledge**: By leveraging the expertise and knowledge present in very large teacher models, the student model benefits from their advanced language capabilities, reasoning abilities, and comprehension skills. Here we specifically train the Pythia-1B model on a boosted explanation dataset using knowledge distillation
- **Deployability**: In specifically choosing a well small pre-trained model, we can deliver a better computation-to-performance ratio. This becomes especially crucial when deploying the model on resource-constrained devices like smartphones. By carefully selecting the 1B parameter model, we can optimize computational efficiency while still achieving satisfactory performance.

II. Literature Overview

Large-scale language models have seen significant advancements in recent years, with models like:

- GPT-2 (124M, 355M, 774M, 1.5B): Pre-trained for 30B Tokens [13]
- Bloom (560M, 1.1B, 1.7B, 3B, 7.1B, 176B): Pre-trained for 350B Tokens [20]
- LLaMA (7B, 65B): Pre-trained for 1T Tokens [17]
- Pythia (70M, 160M, 410M, 1B, 1.4B, 2.8B, 6.9B, 12B): Pre-trained for 300B Tokens [2]
- OpenLLaMA (3B, 7B, 13B): Pre-trained for 1T Tokens [6]
- StableLM (13B): Pre-trained for 800B Tokens [4]
- Flacon (1B*, 7B, 40B): Pre-trained for 1.5T Tokens [1]
- MPT (1B*, 7B, 30B): Pre-trained for 1T Tokens [16]
- OPT (125M, 350M, 1.3B, 2.7B, 6.7B, 13B, 30B, 66B, 175B): Pre-trained for 300B Tokens [23]

These models have demonstrated impressive generalization capabilities, allowing for improved performance across various natural language processing (NLP) tasks [17]. However, smaller models have not received as much attention, typically being trained on datasets of around 300B tokens. Models such as Bloom [20], Pythia [2], Flacon [1], and MPT [16] fall into this category, with no models smaller than 7 billion parameters being trained on 1 trillion tokens [20] [2].

To make smaller models more practical and usable, researchers have explored knowledge distillation techniques, leveraging the knowledge and representations learned by larger models [7]. DistilBERT [14] and distilGPT [10] are early examples of models where knowledge is distilled from larger counterparts into smaller, more efficient versions. However, significant work is still to be done, as replicating the reasoning and logic of the larger models have proven to be challenging [21] [8].

Self-instruct [18] has gained momentum in recent literature, giving birth to various novel approaches:

- Zero-shot learning
- Few-shot learning [3]
- Chain-of-Thought [19]
- Tree-of-Thought [22]

Models like Alpaca [15] and Vicuna [5] have been fine-tuned on small (¡100K examples) instruction-following demonstrations generated from much larger models, such as ChatGPT. LaMini-LM [21] introduces a medium-scale instruction dataset (2.5M examples) derived from ChatGPT and finetunes a variaty of ¡1B parameter models on the instruction dataset. The authors suggest their extensive evaluations of these model reveal that their methods demonstrate that their proposed models achieve comparable performance to Alpaca while being nearly ten times smaller in size [21]. Despite papers reporting similar behavior as their larger counterparts, further evaluations reveal that the imitation models fail to close the performance gap with ChatGPT on tasks that lack strong support in the imitation data. It is noted that the imitation models excel in mimicking ChatGPT's style but struggle with factuality and logic. Concluding that model imitation is not a viable solution, as there exists a significant capabilities gap between open and closed LMs [8].

However, more recent developments challenge such findings, Orca [11] is a notable development in chain-of-thought prompting [19], which was trained on a 5 million example (the largest knowledge distillation to date) dataset extracted from GPT-3 and GPT-4. Despite being a medium-sized model with 13B parameters, Orca demonstrated exceptional performance, achieving results comparable to 13x-35x larger models [11]. The success of Orca highlights the potential of leveraging large-scale instruct datasets and knowledge distillation techniques to enhance the capabilities of smaller models, making them more efficient and effective in complex problem-solving tasks [21].

This trend of knowledge distillation has shown promising results. Logic-based knowledge extraction aims to incorporate logical reasoning and inference into the process of distilling knowledge from large models. By incorporating logical principles and rules, we can aim to improve the interpretability

and explainability of the distilled knowledge. Allowing for compressed models that not only deliver high performance but also provide insights into their decision-making processes based on logical reasoning [21] [7] [18].

III. Methodology

The intuition behind training LLMs lies in the vast amount of data used to develop their understanding of language. Being trained on massive datasets that include a wide variety of texts, ranging from books and articles to websites and social media. Through this exposure, the model learns to recognize patterns, grammar rules, word associations, and contextual meanings present in human language. By tailoring the data to a particular domain or task, the model can learn context-specific information, patterns, and nuances that are crucial for excelling at that specific task

- **Pretraining**: Train model on trillions of words, often extracted from the internet - learn basics of language.
- **Finetuning**: Tune model on high quality texts that resemble ideal model output, e.g. conversations
- **Reinforcement Learning**: Using human assistance, the model receives feedback in the form of rewards or penalties.

A. Boosted Explanation Tuning

Boosted Explanation Tuning is an innovative approach that extends beyond traditional instruction tuning and chain-of-thought prompting. At its core, this method involves creating a non-linear thought graph that more accurately captures the complex reasoning processes inherent in large language models.

We begin by generating a comprehensive thought graph for each problem or task. This graph represents various solution paths, including correct approaches, common misconceptions, and potential pitfalls. Unlike linear chain-of-thought prompts, this graph effectively captures the branching nature of complex reasoning.

For each node within the thought graph, we provide detailed explanations that cover not just the next step in the process, but also the reasoning behind why that step is chosen, what alternatives were considered, and how this step contributes to the overall solution. Additionally, we integrate error analysis into the graph, highlighting common mistakes and explaining their inaccuracies. This helps the model to not only avoid these errors but also to understand the reasoning behind correct solutions.

Where applicable, multiple valid solution paths are included in the thought graph. This exposure to different problem-solving strategies enhances the model's flexibility and robustness. Furthermore, we incorporate metacognitive elements into the explanations, such as strategy selection, progress monitoring, and solution evaluation, helping the model develop a more human-like approach to problem-solving.

By training on this enriched dataset, the student model gains deeper insights into the teacher model's reasoning process. This approach is designed to enable the student model to internalize the underlying logic and reasoning strategies, rather than merely mimicking the surface-level behavior of the teacher.

B. Knowledge Distillation

Knowledge Distillation is the process of transferring knowledge from a large, complex model (the teacher) to a smaller, more efficient model (the student). In the Beluga approach, we employ an advanced form of knowledge distillation to train the Pythia-1B model.

The process begins with the selection of a very large, state-of-the-art language model as the teacher, which serves as the source of knowledge and expertise. Using this teacher model, we generate a large, diverse dataset of responses to a wide range of prompts and tasks. These responses incorporate the boosted explanations described earlier.

Instead of relying solely on the final output of the teacher model, we capture the probability distribution over its output vocabulary. This "soft target" contains more information than a simple classification, allowing the student model to learn the nuances of the teacher's decision-making process. We also apply temperature scaling to the teacher's output distributions, adjusting the "softness" of the targets to control the balance between the highest probability outputs and lower probability alternatives.

To train the student model, we use a combination of cross-entropy loss (to match the teacher's outputs) and a distillation loss (to match the teacher's soft targets). This encourages the student model to mimic both the outputs and the underlying reasoning of the teacher. Additionally, we employ an iterative process where the student model's outputs are compared not just to the teacher's immediate responses, but also to follow-up explanations and clarifications. This helps ensure that the student model captures deeper levels of understanding.

Finally, after the general knowledge distillation, we perform task-specific fine-tuning on specialized datasets. This step helps the model to excel in particular applications while retaining its general capabilities.

By combining these advanced knowledge distillation techniques with our boosted explanation dataset, we aim to create a smaller model that not only matches the performance of much larger models on specific tasks but also demonstrates a deeper understanding of the reasoning processes involved. This approach allows us to leverage the expertise of very large teacher models while creating a more deployable and efficient student model.

IV. Discussion

Our Beluga approach addresses several important challenges in the field of large language models, particularly focusing on making these powerful tools more accessible and practical for real-world applications.

One of the most significant contributions of this work is the introduction of boosted explanation tuning. By creating a non-linear thought graph, we aim to capture more complex reasoning processes than traditional instruction tuning or

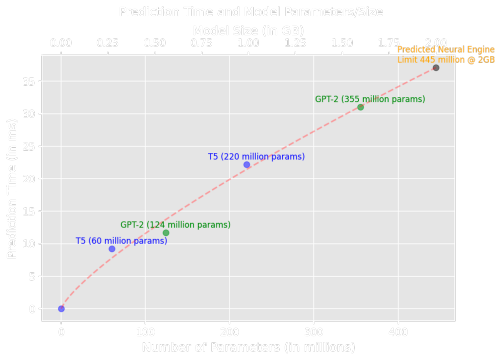

Fig. 1. Prediction on Apple's Neural Engine, which allows for fast and efficient inferences over various platforms like iOS and MacOS.

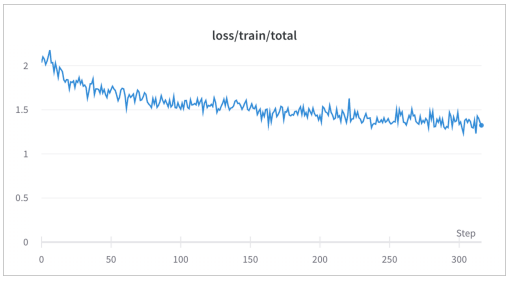

Fig. 2. Training graph of model

chain-of-thought prompting. This approach has the potential to significantly improve the student model's ability to mimic the teacher model's thought processes, potentially leading to better performance and understanding. The focus on knowledge distillation is another crucial aspect of our research. By leveraging the expertise of very large teacher models and distilling this knowledge into a smaller Pythia-1B model, we are addressing one of the major limitations of current LLMs: their size and computational requirements. This approach aligns with recent trends in the field, such as the Orca model, which demonstrated that smaller models can achieve performance comparable to much larger ones when trained on high-quality, diverse datasets.

Our emphasis on deployability is particularly noteworthy. By choosing a smaller pre-trained model and optimizing for a better computation-to-performance ratio, we are making strides towards more widespread adoption of LLM technology. This focus on practical deployment, especially for resource-constrained devices like smartphones, could have significant implications for the accessibility of advanced language processing capabilities. However, it's important to note that we haven't provided detailed performance metrics or comparisons with other state-of-the-art models in this paper. Future work could benefit from more comprehensive evaluations to clearly demonstrate the advantages of the Beluga approach over existing methods.

Additionally, while we mention improved privacy as a benefit of local deployment, it would be valuable to explore this aspect further. As privacy concerns continue to grow in the AI field, a more in-depth discussion of how Beluga addresses these issues could strengthen the paper's impact.

Our approach also raises interesting questions about the trade-offs between model size, performance, and interpretability. While smaller models are more deployable, it would be interesting to explore how much of the larger models' capabilities can truly be distilled without loss of performance or reasoning ability. In conclusion, our Beluga approach presents a promising direction for making LLMs more practical and accessible. By combining advanced training techniques with a focus on deployability, this work contributes to the ongoing efforts to bridge the gap between the immense power of large language models and the practical constraints of real-world applications. Future research could build upon this work by providing more detailed performance analyses, exploring privacy implications further, and investigating the scalability of this approach to even more complex language tasks.

REFERENCES

[1] Ebtesam Almazrouei et al. "Falcon-40B: an open large language model with state-of-the-art performance". In: (2023).

[2] Stella Biderman et al. *Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling*. 2023. arXiv: 2304.01373 [cs.CL].

[3] Tom B. Brown et al. "Language Models are Few-Shot Learners". In: *CoRR* abs/2005.14165 (2020). arXiv: 2005.14165. URL: https://arxiv.org/abs/2005.14165.

[4] CarperAI. *stable-vicuna-13b-delta*. 2023. DOI: 10.57967/HF/0588. URL: https://huggingface.co/CarperAI/stable-vicuna-13b-delta.

[5] Wei-Lin Chiang et al. *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. Mar. 2023. URL: https://lmsys.org/blog/2023-03-30-vicuna/.

[6] Xinyang Geng and Hao Liu. *OpenLLaMA: An Open Reproduction of LLaMA*. May 2023. URL: https://github.com/openlm-research/open_llama.

[7] Yuxian Gu et al. *Knowledge Distillation of Large Language Models*. 2023. arXiv: 2306.08543 [cs.CL].

[8] Arnav Gudibande et al. *The False Promise of Imitating Proprietary LLMs*. 2023. arXiv: 2305.15717 [cs.CL].

[9] Akzharkyn Izbassarova, Aziza Duisembay, and Alex Pappachen James. "Speech recognition application using deep learning neural network". In: *Deep Learning Classifiers with Memristive Networks: Theory and Applications* (2020), pp. 69–79.

[10] Tianda Li et al. "A Short Study on Compressing Decoder-Based Language Models". In: *CoRR* abs/2110.08460 (2021). arXiv: 2110.08460. URL: https://arxiv.org/abs/2110.08460.

[11] Subhabrata Mukherjee et al. *Orca: Progressive Learning from Complex Explanation Traces of GPT-4*. 2023. arXiv: 2306.02707 [cs.CL].

[12] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 [cs.CL].

[13] Alec Radford et al. “Language Models are Unsupervised Multitask Learners”. In: (2019).

[14] Victor Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *CoRR* abs/1910.01108 (2019). arXiv: 1910.01108. URL: http://arxiv.org/abs/1910.01108.

[15] Rohan Taori et al. *Stanford alpaca: An instruction-following llama model*. 2023.

[16] MosaicML NLP Team. *Introducing MPT-7B: A New Standard for Open-Source, ly Usable LLMs*. Accessed: 2023-03-28. 2023. URL: www.mosaicml.com/blog/mpt-7b (visited on 03/28/2023).

[17] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL].

[18] Yizhong Wang et al. *Self-Instruct: Aligning Language Models with Self-Generated Instructions*. 2023. arXiv: 2212.10560 [cs.CL].

[19] Jason Wei et al. “Chain of Thought Prompting Elicits Reasoning in Large Language Models”. In: *CoRR* abs/2201.11903 (2022). arXiv: 2201.11903. URL: https://arxiv.org/abs/2201.11903.

[20] BigScience Workshop et al. *BLOOM: A 176B-Parameter Open-Access Multilingual Language Model*. 2023. arXiv: 2211.05100 [cs.CL].

[21] Minghao Wu et al. *LaMini-LM: A Diverse Herd of Distilled Models from Large-Scale Instructions*. 2023. arXiv: 2304.14402 [cs.CL].

[22] Shunyu Yao et al. *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*. 2023. arXiv: 2305.10601 [cs.CL].

[23] Susan Zhang et al. *OPT: Open Pre-trained Transformer Language Models*. 2022. arXiv: 2205.01068 [cs.CL].

[24] Wanjun Zhong et al. *AGIEval: A Human-Centric Benchmark for Evaluating Foundation Models*. 2023. arXiv: 2304.06364 [cs.CL].