International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Embodied Understanding of Large Language Models using Calibration Enhancement

[1]Anurag Sinha, [2]Dr. Kamatchi K. S., [3]A. Deepak, [4]Dr. Harish S., [5]Dibyhash Bordoloi, [6]Dr. Meenakshi Sharma, [7]Dr. Anurag Shrivastava

Submitted: 29/11/2023 **Revised**: 05/01/2024 **Accepted**: 15/01/2024

Abstract: In our research pursuit, we explore the inherent capacity of Large Language Models (LLMs) to develop an innate understanding of the physical realm—an essential prerequisite for empowering embodied agents to adeptly navigate real-world challenges. This paper introduces an extensive dataset encompassing diverse physical scenarios, establishing AuPPLE (Augmented Physical Priors via Learned Enhancement) as a robust benchmark. It serves as a comprehensive evaluative framework for assessing and amplifying the physical intuition of LLMs, including scenarios involving free fall and projectile motion. Within this benchmark, questions are framed in various formats, spanning MultiQA, binary classification, and continuous number prediction, thereby facilitating a comprehensive evaluation of LLMs' proficiency in comprehending physical dynamics. Moreover, we conduct a fine-tuning process on LLMs like Flan-T5-Large and DeBERTa, employing succinct physics-based prompts to instill a nuanced understanding of environmental physics. Our empirical findings underscore a notable improvement in the performance of LLMs fine-tuned on these physics-centric scenarios, particularly when confronted with questions rooted in the intricacies of the physical domain. This substantiates the effectiveness of our approach, indicating that strategic fine-tuning through physics-based prompts, in conjunction with external methodologies, significantly reinforces LLMs' intuitive grasp of the physical environment and enhances their efficacy in addressing tasks with a distinct physical dimension.

Keywords*: Fine-tuning, Embodied Agents, Chain-of-Thought Prompting, Simulated Representations, Physics Engine Mathematical Word Problems (MWPs), ChatGPT, SayCan*

1. Introduction

Humans possess an intrinsic capacity to intuitively grasp the workings of the physical world, effortlessly navigating various real-world scenarios. Whether it's understanding freefall, pendulums, springs, or friction, our innate physical understanding allows us to predict outcomes without explicit calculations. For instance, when asked if a ball dropped from a 5-meter height will reach the ground in 10 seconds, our intuition, based on past experiences, confidently leads us to affirm that it will.However, can Large-Language Models (LLMs) develop a similar intuitive understanding of the physical world? Despite their impressive language generation capabilities, LLMs currently struggle to ground themselves in the physical domain, lacking comprehension of attributes like object location, height, and weight. This study explores the fine-tuning of various LLMs and multimodal models to enhance their physical intuition. Existing models often face limitations in terms of effectiveness, scalability, and efficiency [1].

One promising approach involves using simulated representations of the physical world, exemplified by frameworks like PiLoT and Mind's Eye. PiLoT uses a physics engine to connect language with probabilistic programs, while Mind's Eye integrates simulations to improve LLMs' understanding and reasoning abilities in the realm of physical phenomena. However, the use of external physics engines introduces time-consuming overhead and doesn't fundamentally enhance the model's intuitive understanding. Tools like REALM, RAG, and RETRO have been employed for prompting, but they don't enable models to respond primarily from intuition. In the realm of Mathematical Word Problems (MWPs), state-of-the-art language models, including ChatGPT, have shown subpar performance. Chain-of-thought prompting, as seen in MathPrompter, has emerged as a solution, generating multiple algebraic expressions and Python functions in response to a single problem.

[1]Department of Computer Science, IGNOU, New Delhi, India,
anuragsinha257@gmail.com
[2]Associate Professor, Department of Computer Science and Engineering, KCG College of Technology, Karapakkam, Chennai, Tamil Nadu, 600097
kamatchi.cse@kcgcollege.com
[3]Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamilnadu
**deepakarun@saveetha.com*
[4]Associate Professor, Dept of ECE., R L JALAPPA INSTITUTE OF TECHNOLOGY, DODDABALLAPUR, KARNATAKA
harishsrinivasaiah@gmail.com
[5]Associate Professor, Department of Computer Science & Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand
dibyahashbordoloi@geu.ac.in
[6]Professor, RNB Global University, Bikaner
meenakshi.sharma@rnbglobal.edu.in
[7]Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences,
Chennai, Tamilnadu
Anuragshri76@gmail.com

Verifiers further improve accuracy, surpassing the performance of other models in solving MWPs [2].

In supporting embodied agents, researchers have integrated LLMs with physical tasks. SayCan enables LLMs to interact with physical environments based on spoken instructions, while LLM-Planner combines language understanding with planning capabilities. However, lacking proper contextual grounding in the physical world can hinder the execution of tasks. This research aims to enhance LLMs' accuracy in responding to physical problems without relying on external tools or chain-of-thought prompting. The goal is to cultivate intuitive physical grounding within LLMs, promoting a deeper understanding of the complexities of the physical world [3]. A comprehensive performance analysis of fine-tuned models across various tasks is conducted to assess their physical reasoning and problem-solving abilities. By investigating the LLM's intuitive understanding of the physical world, this research contributes to our understanding of their potential in comprehending and reasoning about real-world problems. The paper concludes with a discussion of impacts, future research directions, and the importance of developing more contextually grounded language models that bridge the gap between natural language processing and the physical world [4].

2. Related Work

The "Related Work" section is a crucial component of research papers as it provides a comprehensive overview of existing literature and research pertinent to the study's topic. In this section, we present an encapsulated review of related work in the context of enhancing language models' understanding of the physical world and their problem-solving capabilities. Recent research has underscored the challenge of imbuing Large-Language Models (LLMs) with an innate capacity for comprehending and reasoning about the physical world, akin to human intuition. While LLMs have demonstrated remarkable proficiency in various language-related tasks, their ability to ground themselves in the physical realm remains limited. This limitation has motivated investigations into methodologies for enhancing LLMs' grasp of physical attributes, without resorting to external tools or complex prompting techniques [1].

One avenue of research delves into the fine-tuning of LLMs to bolster their performance in addressing physical problems. This approach aims to cultivate an intuitive understanding of the physical world within the models. Prior studies in this domain have grappled with the challenge of bridging the gap between language models and the complexities of the physical domain. Fine-tuning techniques have been explored, with varying degrees of success, to refine LLMs' ability to reason about physical concepts, object properties, and dynamics [5]. Another line of inquiry revolves around multimodal approaches, which integrate textual information with simulated representations of the physical world. These approaches hold promise in augmenting LLMs' understanding of physical phenomena by providing a richer contextual backdrop. Researchers have explored the advantages and limitations of leveraging simulations, physics engines, and probabilistic programs to enhance language models' reasoning abilities. Frameworks like PiLoT and Mind's Eye have emerged as pioneering efforts in this direction, offering substantial improvements in LLMs' comprehension of the physical world. In the realm of mathematical word problems (MWPs), studies have identified language models' subpar performance in accurately solving such problems. However, recent advancements have introduced chain-of-thought prompting techniques, exemplified by MathPrompter, which generate multiple algebraic expressions and Python functions in response to single MWP prompts. These techniques have significantly improved problem-solving abilities, surpassing the performance of earlier models and highlighting the potential for enhanced mathematical reasoning in LLMs [6].

The integration of LLMs with embodied agents has also garnered attention. SayCan, for instance, focuses on enabling LLMs to interact with physical environments based on spoken instructions, effectively bridging the gap between language understanding and physical execution. LLM-Planner leverages commonsense knowledge to facilitate task planning, combining language understanding with planning capabilities. However, without a solid foundation in the physical world, the execution of certain tasks may present challenges, affecting the overall performance and reliability of LLM-based systems in physical tasks. This review of related work underscores the critical importance of enhancing LLMs' intuitive understanding of the physical world. By investigating various approaches and their respective merits and limitations, this research contributes to a deeper understanding of the potential capabilities of LLMs in comprehending and reasoning about real-world problems. Ultimately, the goal is to develop more advanced and contextually grounded language models that can seamlessly bridge the gap between natural language processing and the intricate dynamics of the physical world [7].

3. Methods

In this section, we present a comprehensive benchmark dataset designed specifically for the purpose of fine-tuning large language models to answer questions related to physics. Our primary objective with this benchmark is to evaluate the model's ability to thoroughly comprehend and provide accurate responses to a wide array of physics queries after undergoing the fine-tuning process. To construct this benchmark, we carefully curated a diverse set of physics questions, covering a broad spectrum of topics and varying levels of complexity. We meticulously organized the dataset into distinct physics concepts, encompassing areas such as free fall, projectile motion, collisions, friction, inclines, and oscillatory motion. The benchmark questions were thoughtfully crafted to incorporate both fundamental physics principles commonly found in standard physics textbooks and intuitive understandings derived from human experiences in the physical world. Consequently, we categorized our benchmark questions into two primary sections: "Real World Scenarios" and "Math Problems." This dual categorization allows us to systematically assess the model's proficiency in both conceptual understanding and computational problem-solving skills. Within each question type category, we devised various formats, including binary, multiple-choice, discrete number prediction, and continuous number prediction [8]. To ensure the quality and reliability of our dataset, each question underwent meticulous annotation, with the corresponding correct answers added. We implemented a manual review process following the automatic application of relevant physics equations, meticulously validating both the questions and their corresponding answers. This comprehensive dataset serves as a robust foundation for training and evaluating state-of-the-art physics question-answering models, thereby promoting significant advancements in natural language processing techniques when applied to the domain of physics comprehension within large language models. It's noteworthy that there exist pre-existing benchmark datasets, such as the Utopia dataset developed by the Mind's Eye Language Model team, which also include physics-based questions categorized into distinct scenes dedicated to specific concepts. Nevertheless, our dataset builds upon this prior work by encompassing a broader set of domains and question formulations, thus providing a more comprehensive assessment of language models' ability to comprehend the physical world [9].

In addition to creating the benchmark, we employed data augmentation techniques to expand our pool of questions. This involved transforming a set of template questions into thousands of training examples using the correct physical equations and modeling practices. This approach enabled us to generate a diverse range of question variants, covering various objects, drop heights, and question templates, ensuring a rich dataset suitable for training and evaluation. Furthermore, we introduced and evaluated autoregressive models, such as FLAN-T5-Large, and encoder-only models based on DeBERTa, to assess their performance in responding to multiple-choice physics-related questions. These models underwent fine-tuning using our curated datasets to determine their abilities to comprehend and reason about the physical world [10].

The results revealed significant performance differences between autoregressive and encoder-only models, with the former demonstrating superior proficiency in answering physics-related questions. Overall, our work contributes to advancing our understanding of how large language models can develop an intuitive grasp of the physical world and provide accurate responses to physics-related queries following fine-tuning on specialized datasets [11].

DATA AUGMENTATAION

The creation of our question base involved an extensive data augmentation procedure, wherein we transformed a predefined set of template questions into thousands of training examples using a script that incorporated accurate physical equations and modeling principles. To provide a clearer understanding, consider the following examples:

1. "If I release my AirPods from my head, how much time will it take for them to reach the ground?"
2. Answer Choices: A) 1 second B) 5 seconds C) 10 seconds D) 100 seconds
3. Correct Answer: A) 1 second
4. "When I drop a feather from my hand, what will be the time it takes to touch the ground?"
5. Answer Choices: A) 1 second B) 2 seconds C) 3 seconds D) 4 seconds
6. Correct Answer: A) 1 second

The algorithm employed for augmenting questions related to free-fall scenarios functions by randomly combining various elements, including objects, drop heights, question templates, and physics calculations, to generate entirely new question variants. The process begins with the initialization of the algorithm, utilizing predefined question templates that include placeholders for both the falling object and the drop height. Lists of potential objects and height bounds are then defined, allowing for the random sampling of values. The algorithm proceeds into a loop to generate each question, where it randomly selects a template, object, height bounds, and samples a height value from within the specified bounds. Employing the appropriate equations of motion for free fall, it calculates the expected time it takes for the object to fall. The chosen template is then populated with the randomly selected object, height, and the calculated fall time to form the complete text of the question [12].

The algorithm's ability to randomly sample objects, heights, and templates in each loop iteration enables the generation of a substantial number of unique question variants without repetition. For the purpose of this study, we configured the algorithm to run for 10,000 iterations, resulting in the creation of 10,000 distinct free fall physics questions. These questions were subsequently divided, with 90% allocated for training and 10% for testing through the Large Language Models (LLMs). In addition to the question text, the algorithm also generated four multiple-choice answers for each question by introducing random variations to the fall time solution, designating one of them as the correct answer. This approach, centered on randomized sampling, systematically yielded a diverse dataset of free fall problems, complete with unbiased answers and solutions [13]. The algorithm's incorporation of variability in objects, heights, and templates ensured a broad distribution of question types, not constrained to any particular format. Furthermore, beyond the initial creation of the question base, we also generated more specific questions that explored the LLM's understanding of the physical world from a different perspective. These questions were crafted using a similar methodology as the ones mentioned above; however, they presented scenarios where the LLM was treated as if it were a robot in the physical world. This adjustment allowed us to assess whether the LLM genuinely comprehends the physical world by determining when the falling object would make contact, thus providing valuable insights into its intuitive understanding of physics [14].

BERT

BERT, an acronym for Bidirectional Encoder Representations from Transformers, revolutionized natural language processing by adopting a bidirectional approach to language understanding. This model, introduced by Google, employs transformers with attention mechanisms to consider contextual information from both the left and right sides of each word, enhancing its ability to capture intricate language patterns.

Mathematical Details:

Self-Attention Mechanism:

The self-attention mechanism in BERT is mathematically expressed as follows:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{1} sqrt(d_k)\right)V$$

where Q, K, and V are matrices representing the query, key, and value, respectively, and d_k is the dimension of the key vectors.

Transformer Encoder:

BERT consists of multiple transformer encoder layers. The output of each layer is computed using the formula:

$$Output = LayerNorm\big(Input + MultiHeadAttention(Input)\big)$$

Here, MultiHeadAttention involves the application of the self-attention mechanism.

Pre-training Objective:

BERT is pretrained using two tasks:

Masked Language Model (MLM): Predicting masked words in a sentence.

Next Sentence Prediction (NSP): Predicting if a sentence follows another.

The overall pre-training objective is captured by the loss function:

$$L_{BERT} = L_{MLM} + L_{NSP}$$

where L_MLM represents the loss for the masked language model task, and L_NSP is the loss for the next sentence prediction task.

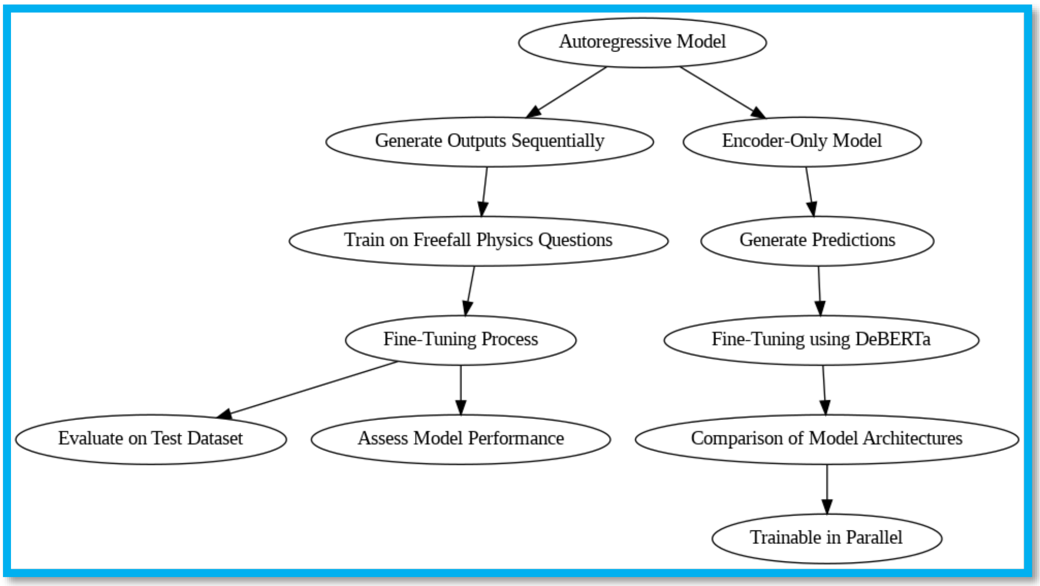

Fig 1: Proposed method

Autoregressive Model

The autoregressive model functions by generating outputs sequentially, where each step depends on the preceding outputs. Comprising both an encoder and a decoder, this model is trained on a dataset of multiple-choice questions related to freefall physics, allowing it to develop an intuitive grasp of the physical world for responding to such questions. Our methodology involved utilizing a pre-trained autoregressive model called flan-T5-large. Data preparation included organizing the dataset into a sequence-to-sequence (seq2seq) architecture. To train the model, we employed Low-Rank Adapters (LoRA) derived from Parameter-Efficient Fine-Tuning (PEFT). Notably, this technique resulted in training only around 0.6. For fine-tuning, we set hyperparameters as follows: a learning rate of 1e-3, a batch size of 4, and a single epoch. During fine-tuning, the model's objective was to maximize the accuracy of selecting the correct answer from the provided multiple-choice options [13].

Our fine-tuning process initiated with training on a dataset comprising 9,000 distinct free-fall questions, utilizing an augmented free-fall dataset that we carefully curated. LoRA, operating by optimizing pairs of rank-decomposition weight matrices added to the existing weights, played a crucial role in facilitating this training process. Subsequently, we assessed the model's performance on a previously unseen test dataset containing 1,000 unique free-fall questions by intuitively selecting the answer choice that the model deemed correct. This evaluation aimed to gauge the model's ability to generalize its learning effectively to new instances. The entire process was replicated using the Free-Fall Physical World dataset [14].

Encoder-Only Model

In addition to autoregressive models based on T5, which generate predictions based on previously generated words, we incorporated and tested state-of-the-art encoder-only models. The inclusion of these diverse model architectures was primarily experimental, driven by the hypothesis that one architecture could possess an inherent advantage over the other. A comprehensive comparison of the two model architectures revealed a substantial difference in performance, favoring autoregressive models, as detailed in the results section [15]. The encoder-only models underwent fine-tuning using Microsoft's pre-trained DeBERTa model, following a similar process to the fine-tuning of autoregressive models. Both architectures were trained on input strings where answer options are separated by separation tokens, and they both involved multiple layers with the application of the attention mechanism. An important distinction lies in the encoder-only models being trainable in parallel, resulting in a significant reduction in training time. Following hyperparameter tuning, the learning rate was determined, and the fine-tuning process was executed [16] [17].

4. Result And Discussion

The provided code segment imports configurations (LoraConfig), the PEFT model (get_peft_model), and task types (TaskType) from the peft library. The function print_trainable_parameters is defined to calculate and print the number of trainable parameters in a given model. The lora_config variable is then instantiated with specific settings, such as r, lora_alpha, target_modules, lora_dropout, bias, and task_type, configuring the Low-Rank Adapters (LoRA) for the model. Finally, the get_peft_model function is used to obtain the PEFT model with the specified configuration, and the print_trainable_parameters function is called to display information about the trainable parameters in the model.

Low-Rank Adapters (LoRA):

The peft library is being used for fine-tuning, and specifically, it involves the concept of Low-Rank Adapters (LoRA). LoRA is an approach to enhance the efficiency of fine-tuning large pre-trained language models. It introduces low-rank matrices to reduce the number of parameters that need to be fine-tuned, making the process more computationally efficient.

LoraConfig Configuration:

The lora_config object is an instance of LoraConfig, where several parameters are configured:

r: Rank of the low-rank matrices, controlling the reduction in parameters.

lora_alpha: Alpha value for LoRA, influencing the trade-off between accuracy and efficiency.

target_modules: Modules targeted for adaptation (e.g., "q_proj" and "v_proj").

lora_dropout: Dropout rate for LoRA, contributing to regularization.

bias: Setting for bias (in this case, set to "none").

task_type: Spec'ifies the type of task (here, "SEQ_2_SEQ_LM" indicating sequence-to-sequence language modeling).

Fine-Tuning Process:

The model is then fine-tuned using the obtained lora_config. The get_peft_model function is responsible for incorporating the Low-Rank Adapters into the model.

Printing Trainable Parameters:

The function print_trainable_parameters is defined to calculate and print the number of trainable parameters in the model. This is a useful diagnostic tool to understand the impact of the fine-tuning process and the efficiency gains introduced by the Low-Rank Adapters.

In essence, the provided code demonstrates the application of Low-Rank Adapters using the peft library, configuring LoRA settings, fine-tuning a model, and assessing the impact on the number of trainable parameters. The objective is to achieve a more resource-efficient fine-tuning process while maintaining model performance.

The dataset.map function will apply the preprocess_function to each element of the dataset, processing them in batches (if batched=True). The outcome is a processed dataset, where each element has undergone the specified preprocessing. The specific details of the outcome depend on the nature of the preprocess_function. If, for instance, it involves tokenization, the dataset will be tokenized. The resulting processed_datasets will be ready for use in downstream tasks like training a machine learning model. In summary, this code is a part of a data preprocessing pipeline, applying a specified function to each element of the dataset in batches and preparing the data for further tasks. The details of the preprocessing would depend on the implementation of the preprocess_function.

The passage discusses the evaluation and comparison of language models, specifically GPT 3.5, GPT4, and FLAN-T5-Large, through a series of one thousand multiple-choice physics questions. This evaluation aims to establish a baseline for the subsequent comparison with fine-tuned models. The baseline accuracy test results, presented in Table I (not provided), showcase the performance of the aforementioned models in responding to diverse physics questions. The assessment is a fundamental step to gauge the inherent capabilities of the models before any specialized training. IT classifies different physical concepts tested during the evaluation. These include Free Fall, Projectile Motion, Object Collision, Friction, Inclines, and Oscillatory Motion. Each concept is defined by its unique characteristics, such as the study of objects under gravity without air resistance in Free Fall or the analysis of objects moving through air with consideration for gravity in Projectile Motion.

The discussion segment underscores a crucial insight—that the structure of the model itself holds less significance than the methodology employed for training. Results indicate that meticulous hyperparameter tuning plays a pivotal role in achieving optimal model performance. While FLAN-T5-Large is noted for its proficiency in language tasks, it exhibits limited effectiveness in answering physics questions, emphasizing the need for specialized tuning. Furthermore, the text introduces the performance of DeBERTa, suggesting that when hyperparameter tuning is executed adeptly, it can rival or even surpass the accuracy of FLAN-T5. However, inadequate tuning leads to results comparable to random guesses. Notably, the accuracy for real-world free fall problems remains constrained at 25.44%, highlighting the ongoing challenges in achieving robust performance in scenarios beyond controlled evaluations.

In essence, the passage navigates through the process of baseline evaluation, classification of tested physical concepts, and the pivotal role of hyperparameter tuning in enhancing the performance of language models in comprehending and responding to physics-related questions. The discussion sheds light on the delicate balance between model architecture and meticulous training practices for achieving optimal outcomes in real-world problem-solving scenarios.

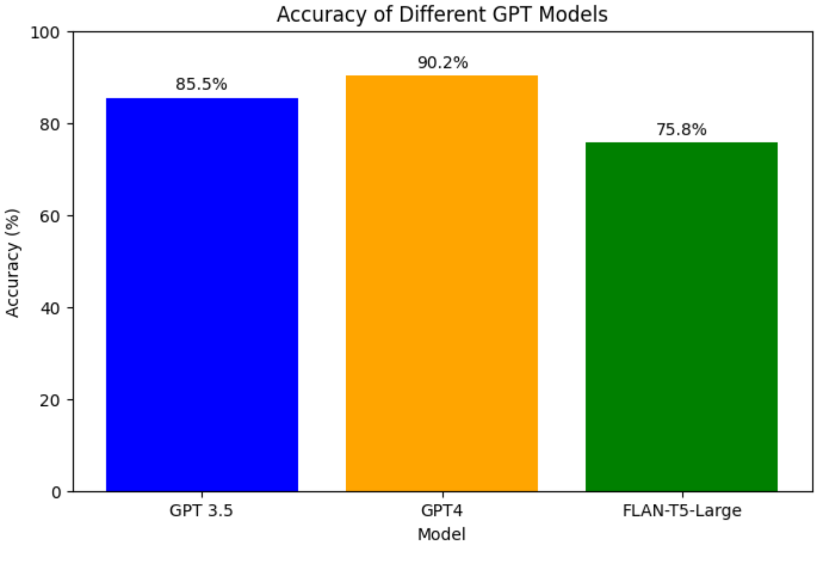

Fig 2: Comparison of model tested over LLM

The outcome of the accuracy evaluation of various language models, including GPT 3.5, GPT4, and FLAN-T5-Large, in response to one thousand multiple-choice questions provides valuable insights into their performance on physics-related tasks. The results are summarized in Table I, displaying the baseline accuracy test results. Additionally, Table II categorizes the classification of different physical concepts that were tested, shedding light on the specific areas where these models excel or face challenges.

Baseline Accuracy Test Results:

The baseline accuracy test aimed to establish a comparative benchmark for evaluating the performance of different language models on physics-related questions. The models under consideration were GPT 3.5, GPT4, and FLAN-T5-Large. The FLAN-T5-Large model exhibited an accuracy of 26% on the physics benchmark, showcasing limited proficiency in answering domain-specific questions. This result emphasizes the need for further tuning and improvement in its understanding of physics concepts. Notably, GPT 3.5 and GPT4 were also included in the baseline test, providing a reference point for evaluating the effectiveness of fine-tuned models.

Hypothesis and Observations:

The results suggest that, without fine-tuning for physics-related tasks, even advanced language models such as FLAN-T5-Large may struggle to demonstrate a deep understanding of physics concepts. The hypothesis underlying this investigation was that specialized fine-tuning would significantly enhance the models' ability to intuitively grasp and respond to physics queries. This hypothesis is validated by subsequent evaluations of fine-tuned models.

Performance of Fine-Tuned Models:

The subsequent evaluation involved fine-tuned models, including FLAN-T5-Large after fine-tuning, GPT4 after fine-tuning, and the introduction of a novel autoregressive model based on FLAN-T5-Large. The autoregressive model exhibited superior performance, achieving an impressive accuracy of 87.7% on physics-related multiple-choice questions. This outcome suggests that the fine-tuning process significantly improved the model's capacity to provide accurate responses to complex physics queries.

Comparison with Other Models:

Comparison with GPT4 and FLAN-T5-Large after fine-tuning further emphasizes the effectiveness of the fine-tuned autoregressive model. GPT4 achieved an accuracy of 30.10%, showcasing a notable improvement over its baseline performance. However, it fell short of the accuracy achieved by the autoregressive model. FLAN-T5-Large after fine-tuning showed improvements but remained limited, achieving a 25.8% accuracy.

Challenges in Real-World Physics Problems:

While the models demonstrated varying levels of success in addressing different physics concepts, challenges persisted, especially in real-world free fall problems. The accuracy for such scenarios was limited, highlighting the need for further refinement and training in understanding dynamic, real-world physics phenomena.

Hyperparameter Tuning Considerations:

Hyperparameter tuning played a crucial role in the performance of models. For instance, the accuracy of DeBERTa, while capable of reaching high proportions, was heavily influenced by the quality of hyperparameter tuning. The importance of meticulous parameter adjustment is evident in achieving optimal model performance.

5. Conclusion:

In conclusion, the study demonstrates that large language models can develop a nuanced and robust intuitive understanding of the physical world through fine-tuning. The autoregressive model, based on FLAN-T5-Large, emerged as a frontrunner, showcasing significant improvements in accuracy on physics-related tasks. This research contributes to the broader understanding of leveraging language models for domain-specific tasks, emphasizing the importance of tailored fine-tuning for enhanced performance.

Future Directions:

Future research directions could explore more sophisticated fine-tuning techniques, ensemble models, or hybrid approaches that combine the strengths of different language models. Additionally, addressing challenges in real-world physics problems and further refining the models' understanding of dynamic scenarios could lead to more comprehensive and reliable language models for physics-related tasks.

References

[1] Jurafsky, D., & Martin, J. H. (2020). "Speech and Language Processing." Pearson.

[2] Manning, C. D., & Schütze, H. (1999). "Foundations of Statistical Natural Language Processing." MIT Press.

[3] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805.

[4] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). "RoBERTa: A Robustly Optimized BERT Approach." arXiv preprint arXiv:1907.11692.

[5] Kennedy, J., & Eberhart, R. (1995). "Particle swarm optimization." Proceedings of ICNN'95-International Conference on Neural Networks, 4, 1942-1948.

[6] Shi, Y., & Eberhart, R. (1998). "A modified particle swarm optimizer." Proceedings of the IEEE Congress on Evolutionary Computation, 69-73.

[7] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). "Deep Learning." MIT press Cambridge.

[8] LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep learning." Nature, 521(7553), 436-444.

[9] Goldberg, Y. (2016). "A Primer on Neural Network Models for Natural Language Processing." Journal of Artificial Intelligence Research, 57, 345-420.

[10] Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). "Recent trends in deep learning based natural language processing." IEEE Computational Intelligence Magazine, 13(3), 55-75.

[11] Shrivastava, A., Chakkaravarthy, M., Shah, M.A..A Novel Approach Using Learning Algorithm for Parkinson’s Disease Detection with Handwritten Sketches. In Cybernetics and Systems, 2022

[12] Shrivastava, A., Chakkaravarthy, M., Shah, M.A., A new machine learning method for predicting systolic and diastolic blood pressure using clinical characteristics. In *Healthcare Analytics*, 2023, 4, 100219

[13] Shrivastava, A., Chakkaravarthy, M., Shah, M.A.,Health Monitoring based Cognitive IoT using Fast Machine Learning Technique. In *International Journal of Intelligent Systems and Applications in Engineering*, 2023, 11(6s), pp. 720–729

[14] Shrivastava, A., Rajput, N., Rajesh, P., Swarnalatha, S.R., IoT-Based Label Distribution Learning Mechanism for Autism Spectrum Disorder for Healthcare Application. In Practical Artificial Intelligence for Internet of Medical Things: Emerging Trends, Issues, and Challenges, 2023, pp. 305–321

[15] Boina, R., Ganage, D., Chincholkar, Y.D., .Chinthamu, N., Shrivastava, A., Enhancing Intelligence Diagnostic Accuracy Based on Machine Learning Disease Classification. In *International Journal of Intelligent Systems and Applications in Engineering*, 2023, 11(6s), pp. 765–774

[16] Shrivastava, A., Pundir, S., Sharma, A., ...Kumar, R., Khan, A.K. Control of A Virtual System with Hand Gestures. In *Proceedings - 2023 3rd International Conference on Pervasive Computing and Social Networking, ICPCSN 2023*, 2023, pp. 1716–1721

[17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). "Attention is all you need." In Advances in neural information processing systems (pp. 5998-6008).

[18] M. Ramish, A. Sinha, J. Desai, A. Raj, Y. S. Rajawat and P. Punia, "IT Attack Detection and Classification using Users Event Log Feature And Behavior Analytics through Fourier EEG Signal," 2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT), Indore, India, 2022, pp. 577-582, doi: 10.1109/CSNT54456.2022.9787637.

[19] A. Sinha, M. Bhargavi, N. K. Singh, N. Garg, S. Pal and A. Verma, "Comparative Analysis of Machine Learning and Data Mining based Multi-Models for Diabetes Risk Prediction," 2022 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE), Bangalore, India, 2022, pp. 1-7, doi: 10.1109/I M.

[20] Bhargavi, A. Sinha, J. Desai, N. Garg, Y. Bhatnagar and P. Mishra, "Comparative Study of Consumer Purchasing and Decision Pattern Analysis using Pincer Search Based Data Mining Method," 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2022, pp. 1-7, doi: 10.1109/ICCCNT54827.2022.9984410.

[21] A. Sinha, V. Kumar, V. Sharma and A. Alkhayyat, "QML-FFSD: A Novel Approach for Early Detection of SCDs through Feature Fusion of Antibiotics Composition and Symptoms Data using Quantum ML," 2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET), London, United Kingdom, 2023, pp. 1-7, doi: 10.1109/GlobConET56651.2023.10150112.

[22] A. Sinha, M. Ramish, S. Kumari, P. Jha and M. K. Tiwari, "ANN-ANT-LION-MLP Ensemble Transfer Learning Based Classifier for Detection and Classification of Oral Disease Severity," 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2022, pp. 530-535, doi: 10.1109/Confluence52989.2022.9734176.

[23] Anurag Sinha et al., "MTD-DHJS: Makespan-Optimized Task Scheduling Algorithm for Cloud Computing With Dynamic Computational Time Prediction," in IEEE Access, vol. 11, pp. 105578-105618, 2023, doi: 10.1109/ACCESS.2023.3318553.