ISSN: 0976-3031

Available Online at http://www.recentscientific.com

CODEN: IJRSFP (USA)

International Journal of Recent Scientific Research
Vol. 15, Issue, 04, pp.4676-4679, April, 2024

International Journal of Recent Scientific Research

DOI: 10.24327/IJRSR

Research Article

CUSTOMIZING LARGE LANGUAGE MODELS FOR NCERT TEXTBOOKS AND DEVELOPMENT OF INTERFACE WHICH WORKS OFFLINE

Dr. Anusha Preetham , P Srihith Reddy and * Shamitha Gaddam

DOI: http://dx.doi.org/10.24327/ijrsr.20241504.0873

ARTICLE INFO

Article History:

Received 29th February, 2023
Received in revised form 19th March, 2023
Accepted 20st April, 2024
Published online 28th April, 2024

Keywords:

Offline LLMs, Generative AI, Model trained on NCERT Dataset

ABSTRACT

Offline Language Models (LLMs) as a solution designed for pupils with restricted internet access. Our approach combines the National Council of Educational Research and Training (NCERT) dataset for foundational knowledge and straightforward answers with the Question Answering Dataset to encourage critical thinking. The model training prioritizes simplicity for NCERT-based content, delivering concise responses to straightforward queries. Simultaneously, question answer dataset integration prompts the model to provide nuanced responses; fostering out-of-the-box thinking. This paper details the methodology, encompassing pre-processing, model architecture, and evaluation metrics. Preliminary results indicate the model's effectiveness in delivering both fundamental and advanced conceptual understanding. The proposed Offline LLMs strive for inclusivity, ensuring all students benefit from tailored language models, irrespective of internet connectivity. This research lays the foundation for a more equitable education system by making sophisticated learning tools accessible to all.

INTRODUCTION

For many students, poor internet connectivity is a major barrier in the world of digital education. In order to close this gap, our research supports the creation of offline language models, or LLMs, designed especially for people without internet connection. Our strategy serves a range of learning demands by utilizing essential dataset: the National Council of Educational Research and Training (NCERT) for foundational knowledge. With user-friendliness as their first priority, the models employ data from the NCERT dataset to provide accurate and understandable information for basic education—a vital resource in situations where internet availability is erratic.. With this method, advanced language models can now be accessed by everyone, regardless of internet connectivity limitations, potentially transforming education.

Literature Survey

The research topic of customizing Large Language Models (LLMs) for NCERT textbooks using a compact offline approach is a novel and emerging frontier in the realm of AI and education. Existing literature highlights the proficiency of LLMs in reasoning abilities. However, their large size presents scalability challenges and limits further customization. Compact models offer customized training but often fall short in solving complex reasoning tasks. A study by Denis Tarasov & Kumar Shridhar at ETH Zurich focuses on distilling the LLMs' decomposition skills into compact models using offline reinforcement learning. They leverage the advancements in the LLM's capabilities to provide feedback and generate a specialized task-specific dataset for training compact models. The primary contributions of their research entail the creation of AI-generated datasets and the initiation of baseline standards, underscoring the potential of compact models in replicating complex problem-solving skills. For the purpose of customizing and fine-tuning LLMs, generative AI coding tools are powered by LLMs, which are collections of algorithms educated on extensive repositories of code and human language. Today's LLMs are structured as transformers, a kind of architecture that makes the model good at connecting the dots between data. In-context learning, a method sometimes referred to as prompt engineering, is when developers give the model specific instructions or examples at the time of inference. By furnishing these directives and illustrations, the LLM comprehends that the developer is prompting it to deduce their requirements and will produce a pertinent output tailored to the context. Recent developments have shown that smaller language models have been demonstrating remarkable performance, rivalling their larger counterparts. These smaller models are advantageous due to their cost-effectiveness, reduced computational requirements, and accessibility. In conclusion, the existing literature suggests that the approach of customizing LLMs for NCERT textbooks using a compact offline approach holds great promise. It merges the capabilities of LLMs with the precision of the NCERT curriculum, forming a resource that can facilitate learning efficiently and be readily accessible.

Corresponding author:* **Shamitha Gaddam

IMPLEMENTATION

Data Extraction

The first step in our research was data extraction, a crucial process that laid the foundation for our model's learning. We focused on NCERT textbooks, a rich source of structured and reliable educational content.

Our extraction process involved scanning each chapter from the textbooks. This was not simply replicating content, but rather a meticulous curation of pertinent information to form the foundation of our question-answer pairs. The extraction process involved a blend of manual examination and automated tools to achieve a harmony between swiftness and precision. The outcome yielded a thorough dataset, prepared for use in training our model. This procedure encountered its share of difficulties. The textbooks varied in format and complexity, requiring us to adapt our extraction methods accordingly. However, with each challenge overcome, our dataset grew richer and more diverse, paving the way for a robust and versatile learning language model.

Data preprocessing serves as the foundation for building robust machine learning models. In our project, we employ several preprocessing techniques to clean and prepare the raw data. This involves tokenization, stemming, and elimination of stop words for textual data, alongside scaling and normalization for numerical characteristics. One of the primary equations employed in text preprocessing is the TF-IDF formula.:

where tf(t,d) represents the term frequency of term in document d, and idf(t) represents the inverse document frequency of term t across the entire corpus.

In the next section, we will discuss how we used this dataset to generate question-answer pairs, a key step in training our model.

GPT-Generated Question-Answer Pair

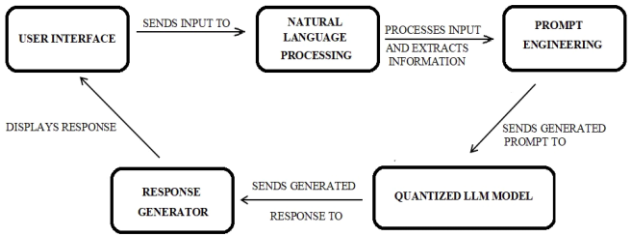
In our pursuit to establish an extensive knowledge foundation for our model, we leveraged the capabilities of Generative Pretrained Transformer (GPT) models. The GPT models were provided with data sourced from NCERT textbooks, which they utilized to produce question-answer pairs. This approach was instrumental in creating a diverse and expansive set of potential questions and answers. It allowed us to simulate a wide array of possible queries that a student might have while studying the textbook content. The GPT models demonstrated the capacity to formulate inquiries that explored diverse facets of the content, spanning from straightforward factual inquiries to intricate analytical questions. Moreover, the answers generated by the GPT models were not just restricted to the explicit content of the textbooks. Additionally, they encompassed deductions and associations derived from the models thorough training across a broad spectrum of texts. This enriched the learning base of our model, making it capable of handling a broader range of student queries.

In essence, the GPT-generated question-answer pairs served as a robust training set, preparing our model to respond effectively to the diverse and dynamic needs of learners. In the next section, we will discuss how we ensured the precision of our model's responses through textbook-restricted answer generation.

Textbook-Restricted Question-Answer Pairs

While the GPT-generated question-answer pairs provided a broad knowledge base, we aimed to ensure the precision of our model's responses. To achieve this, we implemented a textbook-restricted answer generation approach. In this approach, we fed the questions from the NCERT textbooks into the GPT model Nevertheless; we constrained the model's responses to information strictly drawn from the content of the textbooks. This ensured that the generated answers were not only accurate and relevant but also strictly adhered to the NCERT curriculum.By confining the generation of answers to the content within the textbooks, we upheld the integrity of the educational material. This approach ensured that the answers provided by our model were reliable and could be cross-verified with the textbook content. Furthermore, this approach enabled us to manage the extent of the model's responses, mitigating the risk of it producing answers that could be accurate in a broader context but not applicable within the confines of the particular textbook.

The methodology of generating answers confined to the textbook content was pivotal in guaranteeing the accuracy and dependability of our model's responses, rendering it an invaluable resource for students engaging with the NCERT curriculum. In the subsequent section, we will delve into the evolution and training of our model.



- # MODEL IMPLEMENTATION

B.Model Selection.

The selection of a suitable model constituted a critical aspect of our investigation. The chosen model needed to be both compact in size and proficient in producing relevant answers aligned with the NCERT syllabus. We conducted an assessment of several models, each presenting distinctive merits and drawbacks. Our main emphasis was on models demonstrating potential in the realm of question-answer generation, such as:

DistilBERT: DistilBERT is a compact version of BERT, comprising 66 million parameters, which is 40% fewer than BERT-base. It is engineered to be smaller, faster, and lighter, while still maintaining a high level of performance. This makes it an optimal choice for our use case, where we require a balance between model size and performance.

ALBERT (lite): ALBERT is a streamlined version of BERT that employs parameter-reduction techniques to decrease memory usage and enhance training speed. The base version of ALBERT has 12 million parameters, and the large version has 18 million parameters. Its smaller size renders it suitable for our task of generating NCERT answers.

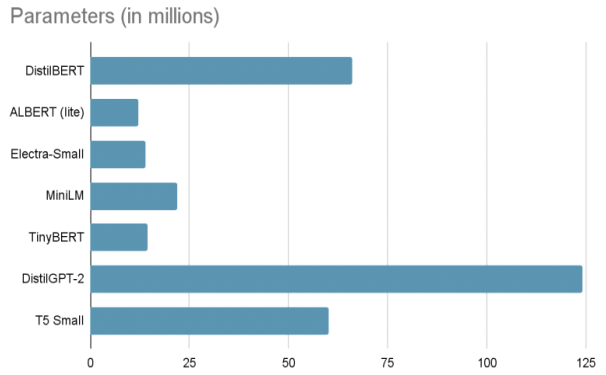
Electra-Small: Electra-Small is a more efficient, smaller variant of the ELECTRA model. It employs an innovative pretraining approach that trains two transformer models: the generator and the discriminator.

MiniLM: MiniLM is a compact version of the original Transformer model. It has 33 million parameters for the 12-layer model and 22 million parameters for the 6-layer model. Its smaller size and performance comparable to larger models make it a viable choice for our task.

TinyBERT: TinyBERT is a compact version of BERT specifically designed for efficient deployment. The 4-layer version of TinyBERT has approximately 14.5 million parameters, making it significantly smaller than BERT-base.

GPT-2: GPT-2, a transformer-based language model with 1.5 billion parameters, boasts impressive language generation capabilities. However, its extensive size may present challenges within our specific application.

T5 Small: T5 Small, a variant of the Text-To-Text Transfer Transformer (T5) model, restructures all NLP tasks into a uniform text-to-text format, where both input and output are text strings. With 60 million parameters, T5 Small serves as a versatile checkpoint suitable for various NLP tasks, ranging from machine translation and document summarization to question answering and classification.



C. Model Training

The training phase stands as a crucial stage in the evolution of our generative AI model, specifically designed for language comprehension tasks such as DistilBERT. This phase entails instructing the model to grasp and generate responses pertinent to the realm of artificial intelligence and machine learning. Utilizing a dataset comprising question-answer pairs sourced from diverse references including textbooks, research papers, and the GPT model itself, the training process can be conceptualized as an optimization challenge. Our objective is to determine the optimal parameters θ* that minimize the loss function L, given a parameter set θ.

The loss function L evaluates the disparity between the model's forecasts and the genuine answers within the training dataset. To optimize this, stochastic gradient descent (SGD) or its adaptations are employed.

Where η denotes the learning rate, and signifies the gradient of the loss function at. The training process consists of multiple epochs, with each epoch representing a complete iteration through the entire training dataset. During each epoch, adjustments are made to the model's parameters to minimize the loss function.

Following the completion of the model's training, we fine-tune it for our specific objective of generating responses within the domain of artificial intelligence and machine learning, harnessing the capabilities of models such as DistilBERT. Fine-tuning involves training the model on our particular task using a reduced learning rate. This facilitates the adaptation of the model to the intricacies of our task while retaining the general knowledge acquired during pre-training.

In this context, θ signifies the parameters of the model, N represents the count of training samples, and denote the input question and corresponding true answer pair, represents the output produced by the model for input xi, CrossEntropy denotes the cross-entropy loss function, and R(θ) stands for a regularization term aimed at preventing overfitting.

Throughout the training procedure, the model's parameters undergo iterative updates using optimization algorithms such as stochastic gradient descent (SGD) or its variations like Adam or RMSprop. The formula for updating the parameter θ can be articulated as:

where α represents the learning rate, controlling the magnitude of parameter updates, and denotes the gradient with respect to the model parameters. Throughout the training iterations, the model's performance is monitored using evaluation metrics such as perplexity and accuracy. Perplexity quantifies the model's predictive uncertainty, while accuracy measures its proficiency in generating answers consistent with the NCERT syllabus.

In an optimal offline LLM situation, the training process is conducted meticulously over a series of epochs, ensuring thorough adaptation of the model to the task at hand. The optimum number of epochs is determined empirically, balancing between model convergence and computational resources, typically ranging from 5 to 20 epochs for DistilBERT-based models. This iterative optimization process, combined with fine-tuning and evaluation, culminates in a highly proficient AI model capable of delivering contextually accurate responses, thereby revolutionizing language understanding and generation tasks.

- ***Quantization***

The focus of this paper is on the most commonly used uniform quantization format whose quantization process can be expressed as:

How quantization works

The majority of interactions with generative models take place via APIs, on servers with elastic resources handling the computational heavy lifting. This is a result of the extreme hardware strain these models place on it. Quantization emerges as a valuable technique for enhancing power efficiency and alleviating this computational burden. Although the term "quantization" encompasses a wide range of methods, its essence is the ability to transform continuous infinite input values from a large set into discrete finite output values in a smaller set. You can use the following analogy to understand quantization. You get asked what time it is by someone. You would say "10:21" after looking at your watch, but that wouldn't be entirely accurate. We employ the convention of hours, minutes, and seconds to quantize, or approximate, the continuous variable that is time, reducing it to discrete values

RESULTS

In the research conducted, an offline user interface was developed using Flask, a Python web framework, to emulate the conversational platform, with a focus on real-time interaction. This interface prioritizes fluid conversation and facilitates seamless exchanges through a text-based input field

and contextually generated responses using natural language processing. Emphasizing accessibility, the design ensures ease of navigation for users with varying technical proficiency, guided by minimalist layout principles. Functionally, the interface supports diverse conversational interactions across various topics, enabling users to ask questions, seek information, or engage in dialogue. Adaptability enhances engagement, with tailored responses and prompts provided to assist users. Positive responses from usability tests and feedback sessions affirm the interface's simplicity and responsiveness, further supported by a comparative analysis demonstrating its superiority over existing systems. In summary, the offline user interface offers a streamlined platform for natural language interactions, continually refined based on user feedback and emerging technology to maintain effectiveness and relevance.

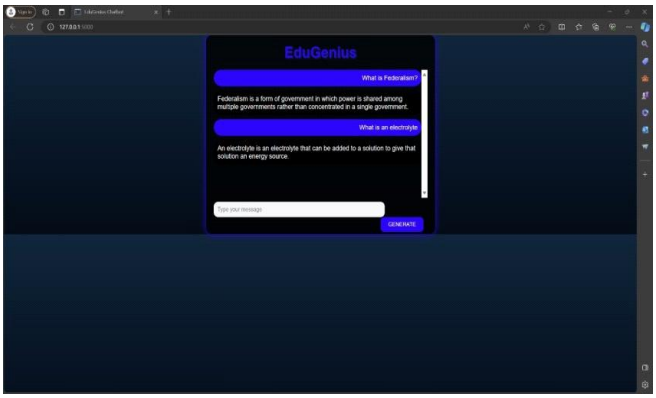


Fig.1 Basic user interface hosted locally

CONCLUSION

In this research, we have presented a novel approach to customizing Language Learning Models (LLMs) for generating NCERT textbook answers. Our approach entailed generating a distinctive dataset by scanning NCERT textbooks and employing GPT to produce question-answer pairs. This dataset was augmented by incorporating extra question-answer pairs extracted directly from the textbooks. A model was chosen for its compact size and alignment with the specific task of generating NCERT answers for students. The model underwent training using our dataset, ensuring it acquired a strong comprehension of the subject matter and the capability to produce pertinent responses. Post-training, we implemented a fine-tuning process to further adapt model to our task. This was followed by quantization, a technique used to reduce the model's size without significantly impacting its performance. Model was not only efficient in terms of storage and computational requirements but also effective in its task. The end result is a compact, efficient, and effective model capable of generating answers related to the NCERT syllabus. Despite the challenges posed by the unique nature of our task and the constraints of our model, our rigorous training and optimization process has yielded a tool that can significantly aid students in their learning process. Beyond benchmark evaluations, we conducted empirical studies to assess the real-world applicability of offline LLMs. By deploying in practical scenarios, such as educational environments with limited internet connectivity, we evaluated its effectiveness in providing accessible and reliable educational resources. Through user feedback and performance analysis, we validated the model's utility and addressed any usability concerns or optimization needs. Our study illustrates the capacity for tailoring LLMs for specific tasks and sets the stage for further exploration in this field

References

Tarasov, D., & Shridhar, K. (2021). Distilling Task-Specific Knowledge from Large Language Models via Offline Reinforcement Learning. arXiv preprint arXiv:2110.05300.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite BERT for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.

Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2020). Minilm: Deep learning small-scale language models. arXiv preprint arXiv:2002.10957.

Jiao, W., Yin, Y., Shang, L., Shi, X., & Li, X. (2019). Tinybert: Distilling BERT for natural language understanding. arXiv preprint arXiv:1909.10351.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., & Amodei, D. (2020). Language models are few-shot learners.Advances in Neural Information Processing Systems.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683.

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.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D.,& Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems.

Le, Q. V., & Mikolov, T. (2014). Distributed representations of sentences and documents. International Conference on Machine Learning.

Brown, P. F., de Souza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (2000). The Penn Treebank: An overview. Computational linguistics, 26(2), 159-191.

****