The Future of Writing: Cost-Effective Essay Generation with Advanced GPT Architecture

1st Hoa Dam Nguyen Quynh
Artificial Intelligence
FPT University
Thu Duc, Ho Chi Minh City, Viet Nam
hoadnqse171093@fpt.edu.vn

2nd Van Nguyen Phuc
Artificial Intelligence
FPT University
Thu Duc, Ho Chi Minh City, Viet Nam
vannpse172344@fpt.edu.vn

3rd Phuc Phan Van
Artificial Intelligence
FPT University
Thu Duc, Ho Chi Minh City, Viet Nam
phucpvse170209@fpt.edu.vn

4th Thanh Nguyen Phuoc
Artificial Intelligence
FPT University
Thu Duc, Ho Chi Minh City, Viet Nam
thanhnpse171408@fpt.edu.vn

5th An Dinh Ngoc
Artificial Intelligence
FPT University
Thu Duc, Ho Chi Minh City, Viet Nam
andnse171386@fpt.edu.vn

6th Hieu Tang Quang
Artificial Intelligence
FPT University
Thu Duc, Ho Chi Minh City, Viet Nam
hieutq10@fpt.edu.vn

***Abstract*—We are currently conducting research on a very interesting topic regarding the use of large language models in processing IELTS essays. Throughout the course of this study, we have utilized a dataset consisting of 5000 essays that were collected from various sources. In addition, we conducted a series of studies on LLM to compare and evaluate performance(which very few papers did before) in order to produce the best model for the topic of generating sentences for the IELTS writing task 2.**

The primary aim of this model is to facilitate the generation of ideas in writing by proposing pertinent English example sentences that align with the given topic and contextual framework. This approach serves to expedite the process of information retrieval and referencing, thereby enhancing the learning experience and fostering greater receptiveness among learners.

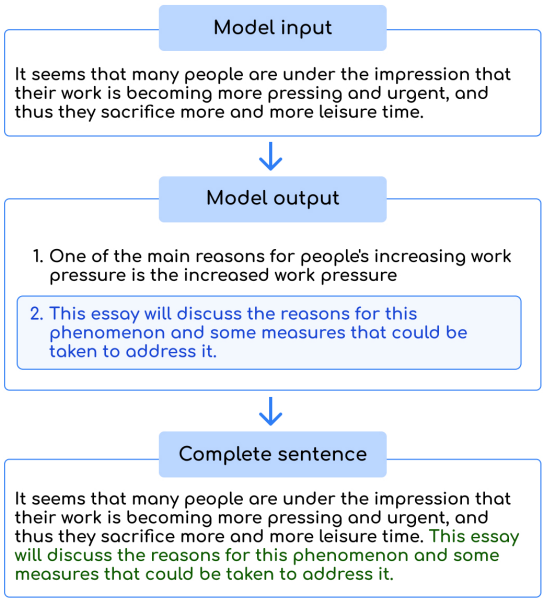

Fig. 1. The example of the model process when receiving input and generating output.

***Index Terms*—Text Generation Model, Natural Language Processing, IELTS Writing Task 2**

I. Introduction

Currently, IELTS is a widely recognized examination that most English language learners are familiar with. IELTS is an abbreviation for the International English Language Testing System, which is a globally used language proficiency assessment for non-native English speakers. In fact, even native speakers may face difficulties if they do not have a thorough understanding of the test format. Among the challenges, Task 2 of the Writing section often poses significant obstacles. It is rigorously evaluated based on criteria such as Task Response, Coherence and Cohesion, Lexical Resource, Grammatical Range and Accuracy, and Task Achievement. In order to address the challenges presented by IELTS Writing Task 2, we have developed a model to assist candidates in generating ideas for their essays, ensuring the essential characteristics required in the IELTS Writing Task 2 are met.

In reality, there are already numerous models specialized in text generation. However, most of these studies tend to handle multiple tasks simultaneously, resulting in their inability to excel in a specific task. Firstly, these models are trained on a vast amount of data without task-specific allocation, leading to a wide range of data sources and a lack of control over the vocabulary, style, and tone of the generated essays. To address this issue, we propose a model specifically designed for generating IELTS Task 2 essays. This model is fine-tuned based on the GPT Neo 1.3b model [1] developed by EleutherAI. The gpt-neo-1.3b model is trained on Pile [2], a large-scale dataset curated by EleutherAI specifically for training this model. We chose this model because it is trained on a domain-specific dataset related to the academic domain. Specifically, Pile comprises books, GitHub repositories, web pages, chat logs, and papers from the medical, physics, mathematics, computer science, and philosophy fields. This enables the

model to possess reasoning abilities in these specific domains and ensures the linguistic quality required for a Task 2 essay.

We utilize the GPT Neo 1.3b model and fine-tuned it by using IELTS Writing Task 2 essays. Specifically, we collected and filtered high-quality IELTS Writing Task 2 essays to expedite and optimize the training process.

With this model, we aim to make a significant contribution to the IELTS community, as well as the broader field of English language learning. Furthermore, we strive to create positive impacts on the development of applications and research related to the education sector.

II. Related Work

Through a development process spanning over 70 years, text-generation models in the field of natural language processing have undergone significant transformations and have brought forth numerous positive benefits. The ultimate goal of text generation models is to produce meaningful sentences while maintaining fluency and eloquence in their language.

In contemporary times, the application of text generation technology spans across diverse domains, wherein the GPT model has gained remarkable popularity as one of the most widely recognized models worldwide, boasting an extensive parameter count of approximately one trillion. Alongside GPT [3], there exist other noteworthy text generation models, such as BertGeneration [4], which is a variant of Bert proposed in the scholarly work entitled "Leveraging Pre-trained Checkpoints for Sequence Generation Tasks" authored by Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. BertGeneration leverages the EncoderDecoderModel [5] to effectively address sequence-to-sequence tasks.

Another prominent architecture in the field is the Text-To-Text Transfer Transformer (T5) [6], which adopts an encoder-decoder framework. Specifically engineered for text-to-text transfer tasks, T5 facilitates the transformation of a given input into the desired output text. Through the process of fine-tuning, T5 models can be adapted to various tasks, including text classification, summarization, translation, and more. These models demonstrate versatility by performing multiple tasks without singularly focusing on a specific one.

Nevertheless, in order to cater to the unique demands and challenges of essay writing for the IELTS Writing Task 2, a dedicated solution called AI-ELTs (Artificial Intelligence for English Language Tests) has been developed. AI-ELTs are meticulously designed to provide specialized support for candidates, aiding them in composing well-structured essays for the IELTS Writing Task 2.

III. Project

A. Motivation

IELTS is one of the most popular English evaluation examinations and this form of test has been widely accepted across many universities as a standard English criterion for enrollment. As [7] stated, "the number of IELTS tests grew to a record 3.5 million in 2018", therefore the desire for a proper strategy for achieving high scores is understandably high. However, among the four main sections in an IELTS test, the Writing section proves to be the most difficult among the students, as their average score is lower than the rest of the sections. The box-and-whisker chart at 2 clearly demonstrates the difference in the mean score of IELTS Writing against the others.

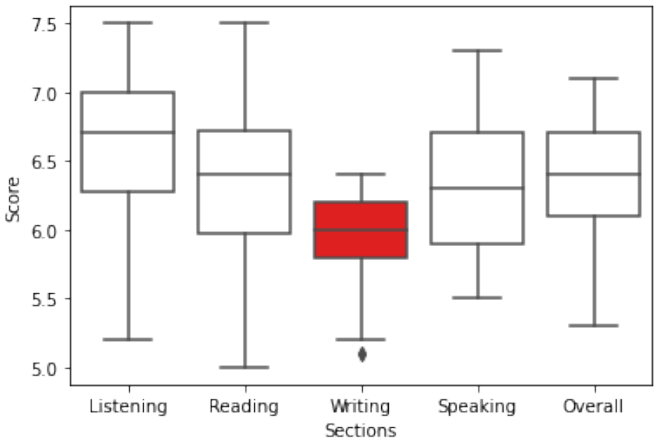

Fig. 2. IELTS Academic mean performance by nationality. Data acquired from [8]

Furthermore, being adept at writing not only helps in getting a high IELTS score, but also an impressive thesis for a scholarships application, or even further, a cover letter for a future job. This motivates our team to develop an AI system based on Natural Language Processing (NLP) to assist students in their writing essays so that they can brush up on their skills for the better.

B. Practical Use

There are many use case scenarios in which IELTS candidates can take full advantage of our system to advance their writing skills:

- Students and non-students who want a comprehensive and IELTS-standard writing style.
- Job seekers can also use this tool to help them work on a reliable cover letter or resume by generating the next sentence. As IELTS-standard writing is highly praised, a curriculum vitae (CV) in this form is acceptable amongst recruiters.
- Students who want to apply to universities will often be required to write a thesis essay, and this tool is a great source of inspiration for practice.
- Other use cases include: generating the next thought after a current thought, etc.

C. System

The overall system can be described as follows:

1) Back-end Model:

- The model takes a text string, which indicates the sentence you currently have, as its input.

- The text string will then go through a series of embedding and transformers layers, producing a vector of numeric values representing the text string, such as how correlated different words are. Each vector element can be the attribute of the corresponding word such as the type of word (noun, verb), etc.
- The vector of the text string will then be fed into a Fully-Connected Neural Network, which finally outputs the probability of the next sentence through a Softmax layer.

2) Front-end:

- After the final output has been generated in the front end, it will be transferred to the front end to be displayed to the user. A webpage will be our main front-end product for the model.
- User will have a variety of options that tweak the desired output, such as the length, temperature (creativity of the text), and the number of generated results, for more freedom of choice.

D. Comparison of LLM Models

1) LLM Models Description: Language models are integral components of Natural Language Processing (NLP) systems used in applications like virtual assistants, chatbots, and sentiment analysis. These models are trained on large text datasets, enabling them to generate text resembling human language and perform diverse NLP tasks, including translation, question-answering, and summarization.

The field of Natural Language Processing (NLP) has witnessed a transformative impact with the advent of large language models (LLMs) powered by deep learning and neural networks. This section focuses on comparing various state-of-the-art LLMs and their capabilities in the context of NLP. Numerous LLMs have been developed, each with its own unique strengths and weaknesses. The subsequent section provides an overview of some of the noteworthy LLMs in the field:

- *GPT-3* [9], an esteemed LLM model developed by OpenAI, stands as a pinnacle of power and recognition. With an impressive parameter count of 175 billion, it currently holds the title of the largest publicly available LLM. The GPT-3 paper introduced the concept of in-context learning (ICL) [10], which leverages LLMs in a few-shot [11] or zero-shot [12] manner. Through ICL, LLMs acquire task understanding by processing natural language text instructions. This integration of pre-training and utilization aligns LLMs with a unified language modeling paradigm. GPT-3 has showcased remarkable performance across a wide range of language tasks, encompassing text generation, translation, question-answering, and more. Its extensive size and capacity enable it to comprehend and generate text that exhibits human-like qualities on a multitude of subjects.
- *GPT-Neo* [1], a variant of the GPT model developed by EleutherAI, focuses on providing a more accessible and computationally efficient alternative to large-scale models like GPT-3. GPT-Neo comes in various sizes, such as GPT-Neo 1.3, GPT-Neo 2.7, and GPT-Neo 4.7, representing the model sizes in terms of the number of billion parameters. Although smaller in size compared to GPT-3, GPT-Neo still exhibits impressive language generation capabilities across a broad range of tasks. These models are trained using publicly available data, aiming to democratize large-scale language models by reducing computational requirements while maintaining strong performance.
- *T5* [6], developed by Google Research, is another notable LLM model that adopts a ”text-to-text” framework. It converts various language tasks into a text-to-text format, leading to remarkable results in different NLP tasks, including summarization, translation, text classification, and more. T5 is renowned for its versatility and effectiveness across multiple domains, showcasing robust performance in both supervised and unsupervised learning scenarios.
- *LaMDA* [13], developed by Google, is an innovative LLM that incorporates conversational abilities. Unlike traditional LLMs, LamDA focuses on generating dynamic and context-aware responses in a conversational setting. It excels in tasks such as chatbots, dialogue systems, and interactive conversational agents. LamDA's unique design enables it to understand and generate natural language responses, leading to more engaging and interactive conversations. While it may have a different focus compared to BERT [4], LamDA showcases impressive performance in conversational AI applications.

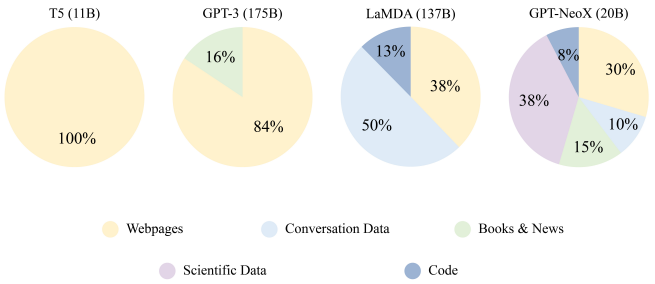

Fig. 3. The distribution of data sources in the pre-training data for existing LLMs has been examined. [14]

2) Data Usage: Large Language Models (LLMs) utilize various types of general data to enhance their linguistic knowledge and generalization capabilities. Figure 3 shows the data categorial usage of four models. A brief summary of three common types of general data:

- *Webpages*: LLMs leverage webpages to acquire diverse

linguistic knowledge and improve performance [15]. Filtering and processing are necessary to ensure data quality, as webpages can contain both high-quality sources like Wikipedia and low-quality content.

- *Conversation Text*: Including conversation data in LLM training enhances conversational competence [16] and question-answering performance [17]. Public conversation corpora subsets or data collected from online social media platforms can be used. However, caution should be exercised to prevent side effects, such as mistaking instructions for conversation starters [18].
- *Books*: Books provide valuable linguistic knowledge, model long-term dependencies, and facilitate coherent narrative generation. Open-source book datasets like Books3 and Bookcorpus2 are commonly used for LLM training.

By leveraging these diverse sources of general text data, LLMs can broaden their linguistic knowledge, improve their ability to understand and generate text and enhance performance across various NLP tasks. **GPT-Neo**, in particular, offers distinct advantages when it comes to the data it utilizes. Its training data is more diverse and includes a significant amount of scientific information, making it particularly suitable for writing tasks that require accurate and factual information, such as the IELTS Task 2. Compared to other LLMs like GPT-3 and T5, GPT-Neo excels in its ability to deliver high performance across a wide range of tasks while maintaining a smaller parameter count. This versatility makes GPT-Neo well-suited for various NLP applications. Moreover, the availability of publicly accessible training data and the user-friendly nature of GPT-Neo models facilitate reproducibility, enabling researchers and developers to build upon existing work and drive further advancements in the field.

E. Model Development

1) Data Preparation: For our dataset, we employed the Huggingface Tokenizer to convert the text data into tensors for computational purposes. Each tensor was constrained to a dimension of 256. As per Huggingface guidelines, these tensors served as both the input and label for the fine-tuning process. However, each essay in our dataset comprised approximately 250-300 words, exceeding the 256-dimensional limit for a single tensor representation. The most basic approach was to divide the text into blocks, with each block containing 256 tokens. However, this approach led to fragmented paragraphs and a loss of coherence between sentences within the same essay. See figure 4

In Block Divide we can see straight, between blocks there is no connection between each other. But for the Sliding Window, we can see that the following blocks can completely carry part of the information of the previous blocks. So we opted to utilize the Sliding Window algorithm with two customizable parameters: window_size and stride. See the pseudo-code of the sliding window algorithm 1.

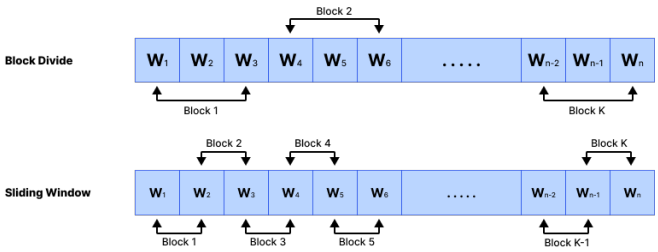

Fig. 4. Example of Block Divide and Sliding Window algorithms. Block Divide is shown with block_size = 3, Sliding Window is shown with window_size=2, and stride=1.

Algorithm 1: Sliding Window Algorithm

Data:
paragraph: *An essay in a dataset*
tokenizer: *Hugging Face's Tokenizer*
stride > 0
window_size > 0

Result: List of sequences token

$A \leftarrow []$;
$start \leftarrow 0$;
$end \leftarrow 0$;
$tokens \leftarrow tokenizer.tokenize(paragraph)$;
$N \leftarrow len(tokens)$;
while $start \leq (N - window\_size)$ **do**
  $end \leftarrow start + window\_size$;
  $sub\_token \leftarrow tokens[start : end]$;
  $A.append(sub\_token)$;
  $start \leftarrow start + stride$;
end

TABLE I
HYPERPARAMETERS

Hyperparameter	Value
Update Steps	7020
Batch Size	32
Warmup Steps	50
Optimizer	AdamW
β_1	0.9
β_2	0.999
ϵ	1×10^{-6}
Learning Rate	3×10^{-4}
Learning Rate Scheduler	Linear Decay
Loss	Cross Entropy
Weight Decay	0

2) Training Details: In this paper, we conducted fine-tuning on our custom dataset using Hugging Face's Trainer. The complete set of parameters used for fine-tuning is outlined in Table III-E1. To optimize the model's parameters, we employed the AdamW optimizer [19] with a learning rate

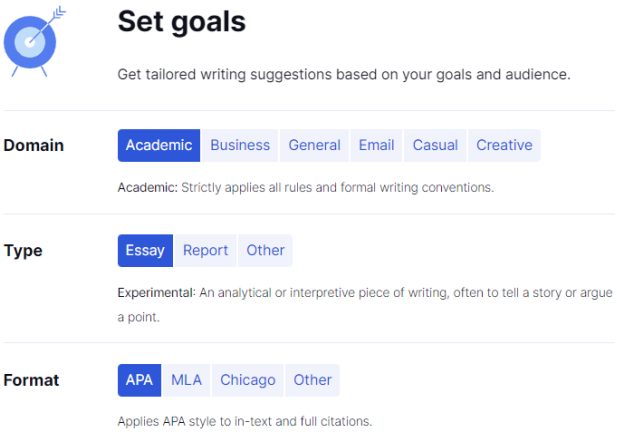

Fig. 5. Grammarly tool setting.

of 3×10^{-4}. Additionally, we applied a linear learning rate decay, gradually reducing the learning rate to zero over time. To prioritize enhancing the quality of results in a small data set, we set the weight decay equal to 0 to prevent the model from getting stuck in sharp local minimum [20] [21]. The Cross-Entropy Loss function [22] was employed to quantify the discrepancy between predictions and actual labels. During the fine-tuning process, we conducted 10 epochs, with 90% of the data allocated for training and the remaining 10% for evaluation.

Furthermore, to optimize performance and efficiency, we employed a mixed-precision approach. Specifically, we utilized a 16-bit integer format (FP16) for specific steps such as forward calculation and gradient computation, while converting back to a 32-bit integer (FP32) format to compute the loss and metric score. This method facilitated faster training, reduced memory usage, increased training batch size, and improved energy efficiency [23], all while maintaining high-quality output.

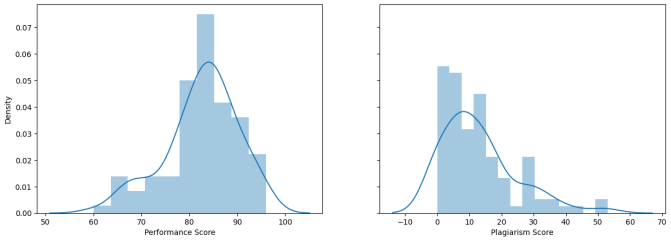

Fig. 6. Distribution of Performance Score & Plagiarism Score on Our Model

3) Evaluation: As there is a lack of publicly available academic evaluation data specifically for this task, we employ BLEU [24] score and ROUGH [25] score, supplemented with loss evaluation to measure the effectiveness of our model. Additionally, we utilize the Grammarly tools which is a digital writing assistant to calculate scores for meaningful words, considering using setting specific domain of academic writing,

TABLE II
EVALUATE SCORE

Model name	Train loss	Valid loss	Bleu score	Rouge score
GPT2-124M	1.7742	2.004801	0.231548	0.497854
GPT2-355M	0.6788	1.792658	0.370100	0.607069
GPT2-774M	0.2429	**1.336823**	0.567848	0.708003
GPT-Neo-125M	1.3602	2.307180	0.288826	0.516954
GPT-Neo-1.3B	**0.1776**	1.743037	**0.600481**	**0.720475**

with the report type and a format categorized as "other" to align with the APA style. The summary of Grammarly setting is in figure 5.

The accuracy score is calculated as

$$a_u = f(n_w, n_i) \tag{1}$$

and performance score as

$$p = c(a_u, a_o) \tag{2}$$

where a_u is user accuracy score, n_w is total number of words, n_i is number of issues, a_o is accuracy score of other text, as explained in [26].

We utilize a premium Grammarly account to evaluate based on 100 samples in the training set to determine if the model is experiencing overfitting. The evaluation on Grammarly will consist of two parts: text quality check and plagiarism check. For the text quality check, we will retrieve the overall score, which is calculated based on criteria such as readability and vocabulary usage in sentences. Moreover, the plagiarism check process is transparent and straightforward. Grammarly allows us to compare sample text passages with existing ones on various websites. Our main findings are presented in III-E2 and 8

- Although the GPT-2 model with a size of 774M exhibited lower training and validation losses, our model, GPT-Neo 1.3 B, yielded intriguing results by outperforming other models in terms of BLEU and ROUGH scores. This indicates that the sentences generated by our model exhibit a higher degree of similarity and quality compared to the reference or human-generated text.
- To enhance the integrity of the generated essays, we also evaluate several sample outputs using the Grammarly tool in our model. After obtaining results for the 100 data samples, the model has demonstrated excellent performance with a text performance score of **82.53%** and a plagiarism score of **12.95%**. The low plagiarism score indicates that the model has effectively avoided overfitting, while the text performance reflects the model's ability to learn contextual representations from preceding text portions to generate subsequent text.

4) Model Deployment:

- To deploy, we will utilize the Client Side Render mechanism, where ReactJS will be used to build the user interface and Python Flask will serve as the backend with

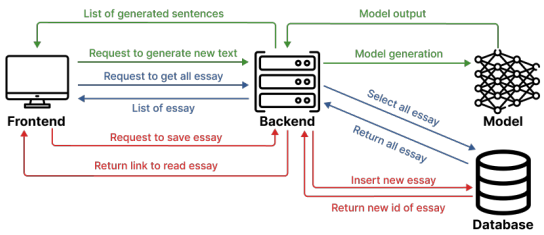

Fig. 7. Diagram of the Working Process.

main functionalities such as model loading and database interaction. View the figure 7 for an overview of the working process.

- ReactJS usage on the Frontend helps to enhance the user experience by implementing features without the need for page reload (this is an advantage of ReactJS when using the Client Side Render mechanism compared to Server Side Render). Additionally, ReactJS provides useful libraries to facilitate user interface development.

- In the special implementation approach, we can successfully deploy GPT-Neo-1.3B on a CPU with 32GB of RAM without the necessity of a GPU. By converting the data type of the weights from float32 to float16 to optimize memory usage and runtime, we can achieve this. However, this comes at the cost of sacrificing the model's accuracy, although the impact is insignificant.

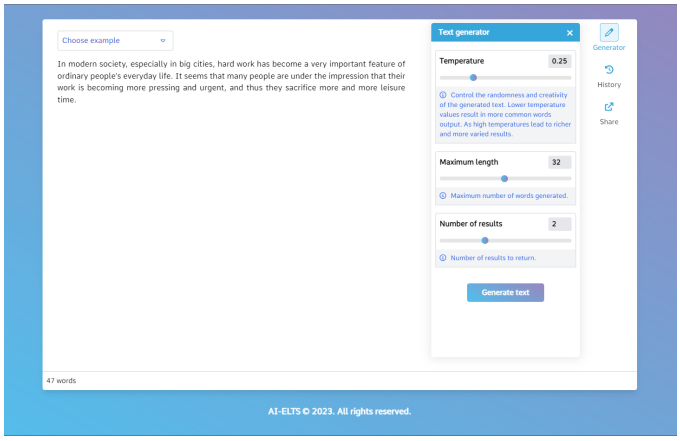

Fig. 8. Web Application for Essay generation.

IV. Limitations and Future Work

Based on our experimental results, analyses, and ablations, we have identified several limitations in our study and potential areas for future work:

- **Building a suitable evaluation metric**: Currently, there is no official metric available to evaluate the appropriateness and quality of sentences in the IELTS Writing Task 2 format. Exploring alternative approaches, such as using a different base encoder-decoder [5] or only decoder model [3], could be beneficial for assessing IELTS scoring and generating sentence examples. This could potentially involve zero-shot [12] learning techniques.
- **Choose an opening, body, or conclusion to create**: Our current focus has been on generating the next sentence based on the given context. However, there is a need to also generate appropriate closing sentences when requested by the user, or sentences suitable in the body. Adding this feature would enhance the usability and completeness of the system for users.
- **Increasing the amount of data**: The availability of IELTS Writing Task 2 data is limited, particularly since a significant portion of it is privately owned. Collecting more high-quality data remains a challenging task. However, increasing the amount of data in training, particularly with a focus on only-decoder models, can significantly improve the results.
- **Reinforcement Learning from Human Ranking**: Although our model demonstrates good results through self-supervised learning, there is still room for improvement. Incorporating reinforcement learning techniques, such as learning from human rankings [27], has the potential to further enhance the model's performance. By leveraging human input and rankings, we can fine-tune the model to prioritize necessary and frequently occurring patterns, leading to higher metrics during validation.

V. Conclusion

In this report, we conducted experiments on GPT-Neo models for the task of generating the next sentence in IELTS text. The model can run without the need for strong hardware like GPU but still obtains comparable performance with the SOTA research methodology. Furthermore, the model can overcome plagiarism and achieve a comfortable score on highly trustworthy sites like Grammarly. In addition, we built a web-oriented application as a means for the user to experience the full capability of our model. However, there is always still room for improvement and we believe that the model, in the state that it is, has the potential to become the next big thing in AI products.

References

[1] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang, M. Pieler, U. S. Prashanth, S. Purohit, L. Reynolds, J. Tow, B. Wang, and S. Weinbach, "Gpt-neox-20b: An open-source autoregressive language model," 2022.
[2] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, and C. Leahy, "The pile: An 800gb dataset of diverse text for language modeling," 2020.
[3] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," 2018.
[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.
[5] T. Nayak and H. T. Ng, "Effective modeling of encoder-decoder architecture for joint entity and relation extraction," 2019.
[6] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2020.
[7] IELTS, "Ielts grows to 3.5 million a year," available: https://www.ielts.org/news/2019/ielts-grows-to-three-and-a-half-million-a-year.

[8] ——, "Test taker performance 2022," available: https://www.ielts.org/for-researchers/test-statistics/test-taker-performance.

[9] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[10] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, L. Li, and Z. Sui, "A survey on in-context learning," 2023.

[11] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[12] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning – a comprehensive evaluation of the good, the bad and the ugly," 2020.

[13] R. Thoppilan, D. D. Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du, Y. Li, H. Lee, H. S. Zheng, A. Ghafouri, M. Menegali, Y. Huang, M. Krikun, D. Lepikhin, J. Qin, D. Chen, Y. Xu, Z. Chen, A. Roberts, M. Bosma, V. Zhao, Y. Zhou, C.-C. Chang, I. Krivokon, W. Rusch, M. Pickett, P. Srinivasan, L. Man, K. Meier-Hellstern, M. R. Morris, T. Doshi, R. D. Santos, T. Duke, J. Soraker, B. Zevenbergen, V. Prabhakaran, M. Diaz, B. Hutchinson, K. Olson, A. Molina, E. Hoffman-John, J. Lee, L. Aroyo, R. Rajakumar, A. Butryna, M. Lamm, V. Kuzmina, J. Fenton, A. Cohen, R. Bernstein, R. Kurzweil, B. Aguera-Arcas, C. Cui, M. Croak, E. Chi, and Q. Le, "Lamda: Language models for dialog applications," 2022.

[14] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models," 2023.

[15] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[16] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba, "Evaluating large language models trained on code," 2021.

[17] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, "Palm: Scaling language modeling with pathways," 2022.

[18] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, "Opt: Open pre-trained transformer language models," 2022.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[20] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, "Sharp minima can generalize for deep nets," 2017.

[21] W. Wen, Y. Wang, F. Yan, C. Xu, C. Wu, Y. Chen, and H. Li, "Smoothout: Smoothing out sharp minima to improve generalization in deep learning," 2018.

[22] I. J. Good, "Rational decisions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 14, no. 1, pp. 107–114, 1952. [Online]. Available: http://www.jstor.org/stable/2984087

[23] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu, "Mixed precision training," 2018.

[24] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: https://aclanthology.org/P02-1040

[25] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

[26] Grammarly, "What is performance and how is it calculated?" available: https://support.grammarly.com/hc/en-us/articles/360007144751.

[27] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," 2022.