International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307 (Online), Volume-14 Issue-3, July 2024

Promoting Project Outcomes: A Development Approach to Generative AI and LLM-Based Software Applications' Deployment

Razieh Akbari Lalaei, Abbas Mahmoudabadi

Abstract: ***In the dynamic realm of artificial intelligence, the emergence of Generative Artificial Intelligence (GAI) has marked a revolutionary stride, particularly in the context of project execution models. This paper delves deep into the sophisticated architectures of GAI, mainly focusing on Large Language Models (LLMs) such as GPT-3 and BERT and their practical applications across varied scenarios. The intricacies of deploying these models have been effectively unraveled to ensure resonating with the specific demands of distinct cases, falling within departmental integration, medical diagnostics, or tailored training modules. Central to the proposed exposition is the innovative "Forward and Back Systematic Approach" designed for executing GAI projects. This approach is meticulously structured to enhance efficiency and ensure a harmonious alignment with the nuanced requirements of diverse applications. We dissect some strategies, including leveraging Private Generalized LLM APIs, in-context learning (ICL), and fine-tuning methodologies, to empower these models to adapt and excel. Furthermore, the proposed platform underscores the pivotal role of evaluation criteria in refining GAI project outcomes, ensuring each model's prowess. It is not strictly theoretical but yields tangible benefits in real-world applications. Under the aegis of this comprehensive exploration, the result of the study would serve as a beacon for enthusiasts and professionals navigating the GAI landscape by offering insights into optimizing robust models for specific and case-driven utilities. Standing on the brink of a modern era in AI, this paper contributes a substantial framework and critical analysis, steering the course for future innovations and applications of GAI.***

Keywords: Generative AI, Large Language Models, LLM-Based Software Applications, Private Generalization, Monitoring

I. INTRODUCTION

Generative Artificial Intelligence, abbreviated as (GAI) and sometimes also known as Generative AI, is primarily based on generative modeling that has distinctive mathematical differences from discriminative modeling [1]. Recent progress and expansion in machine learning have led to more sophisticated, innovative technology and digital content generation like GAI [2]. GAI is an unsupervised modeling or partially supervised machine learning framework that generates artificial relics via the use of statistics, probabilities, etc. [2, 3].

Manuscript received on 04 July 2024 | Revised Manuscript received on 09 July 2024 | Manuscript Accepted on 15 July 2024 | Manuscript published on 30 July 2024.
*Correspondence Author(s)
Razieh Akbari Lalaei, Generative AI Researcher and Developer, Tehran, Iran r.akbari.87@gmail.com
Dr. Abbas Mahmoudabadi*, Ph.D.; Director, Master Program in Industrial Engineering, Mehr Astan University, Gilan, Iran. E-mail: mahmoudabadi@mehrastan.ac.ir, ORCID ID: 0000-0002-1175-6730

Through advances in deep learning (DL), generative AI creates artificial relics by using existing digital content as but not limited to video, images/graphics, text, and a combination of audio and video by examining training examples, and learning their patterns and distribution as well [2-5]. Extant literature has identified two major generative AI; Generative Adversarial Network (GAN) and Generative Pre-trained Transformer (GPT) [1-3, 5-6]. Generative Pre-trained Transformer (GPT) models use great amounts of data, available in public, in the form of digital content named natural language processing (NLP) to read and produce human-like text in several languages and can exhibit creativity in writing from a paragraph to a full research article convincingly (or near convincing) on almost any topics [7]. In addition, these models are capable of engaging customers in human-like conversations like customer-service chatbots or fictional characters in video games [3, 7-8]. In this paper, we intend to introduce Generative AI and its applications and some well-known Large Language Models, and their features when they are practical for various situations. Developing a more effective forward and back systematic approach to deploy a Generative AI like GPT projects is another objective of the present study. We will focus on LLM-based software applications that are specific to a certain case. In the proposed systematic way, the primary target is to introduce the ways through specific situations like departments, medical or training cases, etc. those who want to apply these LLMs for a certain case with various alternatives of data, cost, complexity, and time. By three approaches, we simultaneously explain the applications that exist for harnessing the capabilities of LLMs in development, which encompass creating models from scratch, refining open source models through fine-tuning, or utilizing hosted APIs. In the first section, the Generative AI, its application, LLMs, and their features are discussed in more detail, followed by an introductionto the proposed systematic way in the second section. Since one of the most substantial parts of GPT projects is evaluation, it is also introduced simultaneously with its essential criteria and to mention that choosing each of them depends on the aim of the project based on the above criteria that can be considered to improve the GPT project's outcome.

II. SCIENTIFIC BACKGROUND

A. What is Generative AI?

Generative AI refers to a branch of artificial intelligence, abbreviated as AI, which focuses on creating models and algorithms capable of generating new or original content including images, text, music, and even videos.

Unlike traditional AI models that are trained to perform specific tasks, generative AI models aim to learn and mimic patterns from existing data to generate new (unique) outputs. In addition, Generative AI has a broad range of applications. For instance, generative models can generate realistic images, generate variations of present images, or even complete missing parts of an image in computer science. In natural language processing, generative models can be used for language translation, text synthesis, or even to develop conversational agents that produce human-like responses. Beyond these examples, Generative AI can perform art generation, data augmentation, and even generate synthetic medical images for research and diagnosis. It is an efficient and creative tool that allows us to explore the boundaries of what is possible in computer science. However, it is worth noting that generative AI also expresses ethical concerns. The ability to generate realistic and convincing fake content can be misused for malicious purposes, like creating deep fakes or spreading disinformation. As a result, there is ongoing research and development of techniques to detect and mitigate the potential negative impacts of generative AI. In summary, generative AI holds magnificent promise for various creative, practical applications and for generating creative and unique content. It continues to remain an active area of research and development, pushing the boundaries of what machines can create and augmenting human creativity in modern and exciting ways [9].

B. Domains of Generative AI

It is time to dive deeply into domains of generative AI in detail, including what it is, how it works, and some practical applications listed below as text, image, audio, and video [9].

Text Generation: Text generation involves utilizing AI models to generate humanlike text based on input prompts. Models like GPT-3 use Transformer architectures. The models are pre-trained on vast text datasets to learn grammar, context, and semantics. Given a prompt, they predict the following word or phrase based on patterns they have learned. Text generation is applied in content creation, chatbots, and code generation. Businesses can employ it for crafting blog posts, automating customer support responses, and even generating code snippets. Strategists can harness it to quickly draft marketing copy or create personalized messages for customers.

Image Generation: Image generation involves using deep learning models to construct images that look real. GANs consist of a generator (creates images) and a discriminator (determines real vs. fake). They compete in a feedback loop with the generator getting better at producing images that the discriminator can't distinguish from real ones. These models are employed in art, design, and product visualization.

Audio Generation: Audio generation involves AI creating music, sounds, or even humanlike voices. Models like Wave GAN analyze and mimic audio waveforms. They are trained on large datasets to capture nuances of sound. AI-generated music can be used in advertisements, videos, or as background tracks.

Video Generation: Video generation involves AI creating videos, often by combining existing visuals or completing missing parts. Video generating is complex due to the temporal nature of the videos.Some models use text descriptions to generate scenes, while others predict missing frames in videos. AI-generated videos can be used in personalized messages, dynamic ads, or even content marketing.

There are salient boundaries of GAI that are important limitations in real-world applications. Generative AI models may produce output errors. This is owing to the underlying nature of machine learning models relying on probabilistic algorithms for making inferences. For example, generative AI models generate the most probable response to a prompt but are not necessarily suitable. As such, challenges arise as outputs are indistinguishable from authentic content and may present misinformation or deceive users [10]. This problem in emergent behavior is addressed by hallucination [11], which refers to mistakes in the generated text that are semantically or syntactically plausible but are nonsensical or incorrect. Put differently, the generative AI model produces content that is not based on any facts or evidence, but rather on its own assumptions or biases. Moreover, the output of GAI is typically not easily verifiable.

a. Large Language Models (LLMs)

Large Language Models (LLMs) refer to a class of advanced artificial intelligence models specifically designed to process and interpret human language at an extensive scale. LLMs have emerged as cutting-edge artificial intelligence systems designed to process and generate text, aiming to communicate coherently [12]. These models are typically developed using deep learning techniques, particularly Transformer-based architectures, and are trained on vast amounts of textual data from the Internet. The key characteristic of large language models represents their ability to learn complex patterns, semantic representations, and contextual relationships in natural language. They can generate human-like text, translate between languages, answer questions, perform sentiment analysis, and accomplish a broad range of natural language processing tasks. One of the most well-known examples of large language models is OpenAI's GPT (Generative Pre-trained Transformer) series, which includes such models as GPT-3. These models are pre-trained on massive datasets and can be fine-tuned for specific applications, allowing them to adapt and excel in various language-related tasks. The capabilities of large language models have promoted significant advancements in natural language processing, making them instrumental in various industries, including customer support, content generation, language translation, and more applications. However, they also raise crucial concerns regarding ethics, bias, and misuse due to their potential to generate humanlike text and spread misinformation if not used responsibly. Some notable examples of LLMs are listed in Table 1 [9].

Table 1. Comparison of LLMs

LLM Type	Description	Model Size	Source Type
GPT	GPT is the fourth version of OpenAI's Generative Pre-trained Transformer series. It is known for its ability to generate humanlike text and has demonstrated proficiency in answering questions, creating poetry, and even writing code.	175 billion parameters	Closed Source
BERT	BERT (Bidirectional Encoder Representations from Transformers), Developed by Google, BERT is a pivotal LLM that captures context from both directions of the input text, making it adept at understanding language nuances and relationships. It has become a foundational model for a wide range of NLP tasks.	Ranges from 110 million to 340 million parameters (depending on the version)	Open Source
T5	T5 (Text-to-Text Transfer Transformer) Also developed by Google, T5 approaches all NLP tasks as text-to-text problems. This unifying framework has shown outstanding performance in tasks like translations, summarization, and question-answering.	Up to 11 billion parameters	Open Source
RoBERTa	Facebook's RoBERTa is an optimized version of BERT that has achieved state-of-the-art results across various NLP benchmarks. It builds upon BERT's architecture and training process, further improving language understanding capabilities.	Varied configurations, ranging up to 355 million parameters	Open Source

Moreover, it is sometimes useful to distinguish between two types of LLM-based applications. One is examples like brainstorming, where it could be quite natural for everyone to type a prompt into ChatGPT, Bard, or Bing chat. The other free or paid large language models on the Internet and get a result back. Both are interface-based applications on the web. In contrast, in the example of recognizing, if an email is a customer complaint, this fits more into a company's email routing workflow. As a result, it doesn't really make sense for anyone to cut and paste customer emails at a time into a web interface to get back answers as to which ones are actually complaining emails.This is an example of an LLM that would make sense when it is built into a larger software automation that helps a company in automated email routing. The second example is writing by answering HR questions, both are LLM-based applications.It will also make more sense as a software-based LLM application because it needs access to information about a specific company's parking policy for employees, whereas a general large language model on the Internet probably doesn't have suchinformation. Numerous approaches exist for harnessing the capabilities of LLMs in development, which encompass creating models from scratch, refining open source models through fine-tuning, or utilizing hosted APIs. Here are three ways you can enable LLMs in different cases [9]:

Private Generalized LLM API:A private generalized LLM API is a way for enterprises or cases to access a large language model (LLM) that has been trained on a massive dataset of text and code. The API is private. It means that the enterprise is the only one who can operate it.This ensures that the enterprise's data is kept private. There are several benefits of using a private generalized LLM API [9] such as 1) it allows to customize the LLM to their specific needs, 2) it is more secure than using a public LLM API and 3) it is more scalable than using a public LLM API.

Design Strategy to Enable LLMs for Different Cases(ICL): With the increasing ability of LLMs, ICL has become a new paradigm for natural language processing (NLP), where LLMs provide predictions based on contexts augmented with a few training examples. It has been a new trend exploring ICL to evaluate and extrapolate the ability of LLMs [13]. At its core, ICL involves employing off-the-shelf LLMs (without fine-tuning) and manipulating their behavior via astute prompts and conditioning based on private "contextual" data. Consider the scenario of crafting a chatbot to address queries related to a collection of genuine documents. A straightforward approach might involve inserting all documents into a ChatGPT or GPT-4 prompt, followed by posing questions about them. While this might suffice for minute datasets, it isn't scalable.As this context window limit is approached, the largest GPT-4 model can exclusively handle around 50 pages of input text and its performance degrades significantly in terms of inference time and accuracy. ICL tackles this quandary ingeniously by adopting a stratagem: instead of supplying all documents with each LLM prompt, it dispatches only a select set of the most pertinent ones. These pertinent documents are determined with the aid of LLMs. In broad strokes, the workflow can be partitioned into three phases 1) Data Preprocessing/Embedding, 2) Prompt Construction/Retrieval, and 3) Prompt Execution/Inference [9]. Though this may appear intricate, it is often easier than the alternatives training or fine-tuning. ICL does not necessitate a dedicated team of machine learning engineers. Additionally, you are not compelled to manage your own infrastructure or invest in costly dedicated instances of Open AI. This approach essentially transforms an AI challenge into a data engineering task, a domain that many startups and established companies are already familiar with. Given that specific information needs to be presented in the training set multiple times for an LLM to retain the model via fine-tuning, it generally surpasses fine-tuning for moderately small datasets, and it can swiftly incorporate new data in almost real-time.

Fine-Tuning: In specialized domains like biomedicine and finance, LLMs often require fine-tuning of training data to acquire domain-specific knowledge and expressive capabilities, enabling them to effectively address domain-specific queries [14, 15].Fine-tuning with transfer learning is a technique that uses a pre-trained LLM as a starting point for training a new model on a specific task or domain. This can be done by freezing some of the layers of the pre-trained LLM and only training the remaining layers. This helps to prevent the model from over-fitting to the new data and ensures that it still retains the general knowledge that it learned from the pre-trained LLM. The steps involved in fine-tuning with transfer learning can be summarized as 1) Choose a pre-trained LLM, 2) Collect a dataset of text and code that is specific to the task or domain, 3) Prepare the dataset for fine-tuning, 4) Freeze some of the layers of the pre-trained LLM, 5) Train the remaining layers of the LLM on the training set, and eventually, get an idea of how the model has learned to perform the task [2]. Table 2 summarizes the complexity and cost ofthe three approaches discussed above.

Table 2: The Comparison of Complexity and cost for Three Approaches of LLM

Approaches	Complexity	Cost
Private Generalized LLM API	Low	Variable (low to high)
Design Strategy to Enable LLMs for Different Cases: ICL	Medium	Medium
Fine-Tuning	High	High

C. Monitoring Generative AI Models:

Although instruction-tuned LLMs exhibit impressive capabilities, these aligned LLMs are still suffering from annotators' biases, catering to humans, hallucination, etc. To provide a comprehensive view of LLMs' alignment evaluation [16]. LLMs are increasingly being deployed in pervasive applications such as chatbots, content moderation tools, search engines, and web browsers [17, 18], which drastically increases the risk and potential harm of adverse social consequences [19, 20]. Monitoring generative AI models, especially LLMs involves tracking various dimensions to ensure their responsible and effective uses. Over here, how you can include the aspects of correctness, performance, cost, robustness, prompt monitoring, latency, transparency, bias, A/B testing, and safety monitoring in this strategy [9] with more detail in Table 3.

Following the above mentioned, it is clear that text generation would be a complicated process that needs accurate, well-defined, and customized processes. It means that each step needs attention to improve more practices. Therefore, the essential concept behind the proposed procedure here is to restrict the bound of text searching to achieve more accurate results. The idea is discussed in the following sections focusing more on theoretical approaches. Accordingly, the depicted figures and discussions are provided to help readers on what has been proposed and what is different from the previous studies.

Table 3: Criteria for Monitoring Generative AI Models

Criterion	Definition
Correctness	Correctness refers to the accuracy of the generated content and whether it aligns with the desired outcomes
Performance	Performance relates to the quality of generated content in terms of fluency, coherence, and relevance
Cost	Cost monitoring involves tracking the computational resources and infrastructure expenses associated with running the AI model
Robustness	Robustness assesses the AI model's ability to handle diverse inputs and adapt to different contexts
Prompt Monitoring	Prompt monitoring involves examining the prompts or inputs provided to the AI model and ensuring they align withethical guidelines
Latency	Latency measures the response time of the AI model, ensuring it meets user expectations for timely interactions
Transparency	Transparency involves providing insights into how the AI model operates and makes decisions
Bias	Bias monitoring focuses on identifying and mitigating biases in the model's outputs, such as gender, race, or cultural biases
A/B Testing	A/B testing involves comparing the performance of different model versions or configurations
Safety Monitoring	Safety monitoring aims to prevent harmful actions or outputs from the AI model

III. IDEA CREATION AND PROPOSED PROCEDURE

Since the concept behind the present study is to advance a procedure enhancing the accuracy of text generation, looking at deeply previous works, would help readers to better recognize the main difference between what has been done so far and what is currently proposed. The method of the text generation here lies in the formal presentation of the forward and back systematic approach doing a Generative AI such as GPT projects, so before introducing the model, we review the previous well-known structures. To have an effective Generative AI project, some methods are defined followed by reviewing them to make a clear understanding to introduce our proposed Method.

A. What is Present as Basic?

Basic Structure: The basic project lifecycle of a Generative AI deals with 4 core principlesthat sequentially happen as shown in Figure 1. [21][22]:

1. **Scope:** define the problem statement you want to solve using LLM to determine how it should work
2. **Select:** Choose a model pre-train your — existing one or train from scratch
3. **Adapt and Align Model:** develop and align model to design with Prompts, fine-tune and evaluate for best output
4. **Application Integration:** optimize and deploy models for inference, and build LLM-powered applications.

Gen AI/LLM Testbed: To harness the maximum potential of LLMs and ensure their responsible development, it is crucial to establish a dedicated LLM testbed. This testbed serves as a controlled environment for researching, testing, and evaluating LLMs, facilitating innovation while addressing ethical, safety, and performance concerns [9].

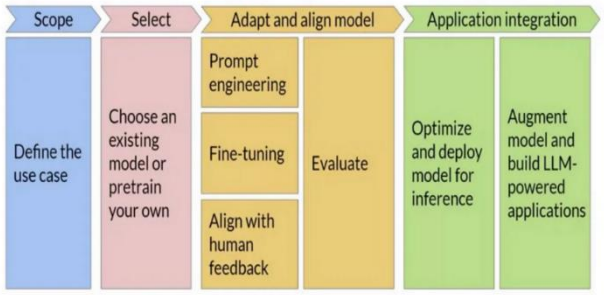

Figure 1: Generative AI Project Life Cycle

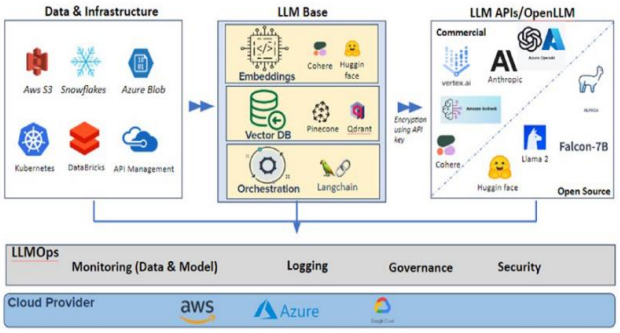

Figure 2: Gen AI/LLM Testbed

Retrieval Number: 100.1/ijsce.D363614040924
DOI: 10.35940/ijsce.D3636.14030724
Journal Website: www.ijsce.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

LLMOps: What's more, the LLMOps (large language model operations) platforms offer a well-defined and comprehensive workflow that covers training, optimization, deployment, and continuous monitoring of LLMs, whether they are open source or proprietary. This streamlined approach is designed to expedite the implementation of generative AI models and their applications. The cases that increasingly integrate LLMs into their operations become essential to establish robust and efficient LLMOps. This section delves into the significance of LLMOps and how it ensures the reliability and efficiency of LLMs in enterprise settings [9]. Sustaining oversight of generative AI models and applications relies ona continuous monitoring process. These processes aim to tackle challenges like data drift and other factors that might hinder their ability to generate accurate and secure results. Figure 3 represents the LLMOps workflow. This Figure outlines the workflow for developing and deploying a Large Language Model (LLM), which seems to be referenced as "LLMOps" here.

- **Data:** Training data collection is the initial step where it is collected for training the model. The collected data is then preprocessed, named the processing stage. This might involve cleaning the data, formatting it properly, and possibly transforming it to be effectively used for training the model.
- **Model Development:** The term "Open Source Foundation Model" indicates that an existing open-source model can be utilized as a starting point. This might be a pattern that has been pre-trained on a large dataset and is available for further customization. Training/Fine-Tuning, The foundation model is then trained or fine-tuned on the specific dataset prepared in the earlier step. This process adjusts the weights within the model to specialize it for the tasks and data it will be used for. Trained Model (Fine-Tuned), the outcome of the training/fine-tuning process is a model that has been customized for specific tasks or data.
- **Deployment &Usage Deploy:** The trained and fine-tuned model is deployed to an environment where it can be used. This could be on a server, cloud, environment, or any platform that supports the deployment stage. Model Deployment (self-hosted or hosted), refers to the options for where the model is deployed. It can be self-hosted, meaning hosted on the organization's own servers, or hosted, and on external servers, such as cloud service providers.
- **Data Storage (Embedding):** In the deployment environment, there is a system for embedding stores that are vector representations of input data which can be used for various purposes like similarity searching, data retrieval, etc.
- **Prompt:** The deployment system is set up to take in prompts or input data to generate responses or perform tasks by utilizing the model.
- **Monitor:** After deployment, the system is continuously monitored to ensure that it functions correctly and to perform maintenance tasks like updating the model, retraining with new data, or improving performance.

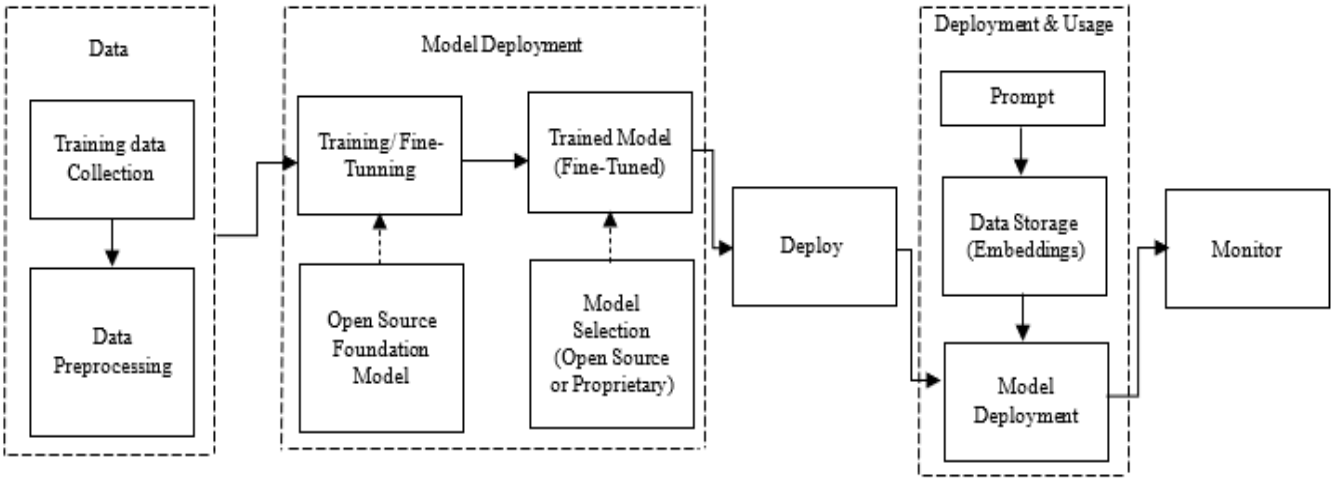

Figure 3:LLMOps Workflow [9]

B. What is Proposed to Enhance Text Generation Accuracy?

The seven-step process for effectively utilizing large language models (LLMs) in different cases encompasses a comprehensive approach. It begins by determining the specific aim, followed by identifying the best-suited LLM for individual needs. The process involves determining specific data requirements, preparing and integrating this data, and then developing strategies for training and fine-tuning the LLM to the particular needs. Each step is critical for optimizing the LLM's performance and its alignment with the case's objectives as well as culminating in a comprehensive evaluation to ensure it meets the expected standards of accuracy and efficiency. Our novelty and suggestion lie in the fourth phase in which we intend to improve the method of customization on tailoring data to achieve more accurate and impactful results. The primary concept behind the idea is that whenever we restrict the bound of search, more accurate auto-generated texts will be resulted by the models.As shown in Figure 4 in a dashed box, the customization begins to receive private or specific row data that may happen in real experiments. The boxes, depicted in black and words in white, indicate they are different from what has been previously proposed. This systematic approach is key to leveraging the maximum potential of LLMs in a specific case context. The above-mentioned steps are discussed more in detail below where Figure 4 depicts what is happening over the implementation of LLM.

1. In the first step, the focus is on defining the specific objectives you want to achieve with a large language model (LLM). This involves identifying the key areas in the specific business where LLMs can add value, such as improving customer service or enhancing data analysis capabilities. It is crucial to align the use of LLMs with your broader case's goals to ensure implementation is relevant and impactful.
2. In the second step, you evaluate and select the most suitable large language model for your specific needs. This decision is based on factors like the model's capabilities, scalability, and compatibility with your data and objectives. It is essential to consider both the technical aspects of the LLMs and how their features align with your intended applications.
3. In the third step, you identify and collect the precise data sets required to train or work with your chosen large language model (LLM). This involves selecting relevant and high-quality data that aligns with your objectives and the model's capabilities. The focus is on gathering diverse and comprehensive data to ensure the LLM can learn effectively and deliver accurate, contextually relevant outputs.
4. In the fourth step, the focus is on tailoring the chosen large language model to provide your specific requirements. Since the idea behind the present research work is developed in this phase, it is depicted differently, by dashed lines, in Figure 4. As shown in the figure, the following procedures in the customization phase depend on the type of data which is typically defined as private and specific raw data. If they are Private-Based, ICL, Prompt, and Human feedback are the following steps. Put differently, if data are in raw formats, pre-train, prompt, human feedback, and fine-tuning are implemented in customization. This customization can involve fine-tuning the model with your unique dataset, adjusting parameters to design your application, and integrating domain-specific knowledge. The goal is to ensure that the LLM performs optimally for your particular use case, delivering results that are not only accurate but also extremely relevant to your case's needs.
5. In the fifth step, the process involves launching the customized large language model into a real-world and limited environment. This crucial phase is where the LLM is integrated into existing systems or platforms and is made operationally for end-users. A successful deployment requires thorough testing to ensure the model functions correctly and efficiently in the intended application, addressing any technical challenges that arise during this transition to practical use.
6. The sixth step continuously involves overseeing the performance and outputs of your large language model (LLM) post-deployment. This monitoring is crucial to ensure the model remains accurate, relevant, and aligned with evolving cases' needs and data environments. It includes tracking model performance metrics, detecting any anomalies or drifts in output quality, and making necessary adjustments to maintain optimal operation and compliance with standards or regulations. After evaluating, if necessary it will need to shift back to the previous steps and try to refine.
7. The final step in utilizing large language models involves implementing the model in a live environment and on a broad scale. This stage is critical for seeing how it performs under real-world conditions and for providing valuable insights or services. It's essential to ensure seamless integration with existing systems and to establish protocols for ongoing support and maintenance to address any issues that may arise during its operational use.

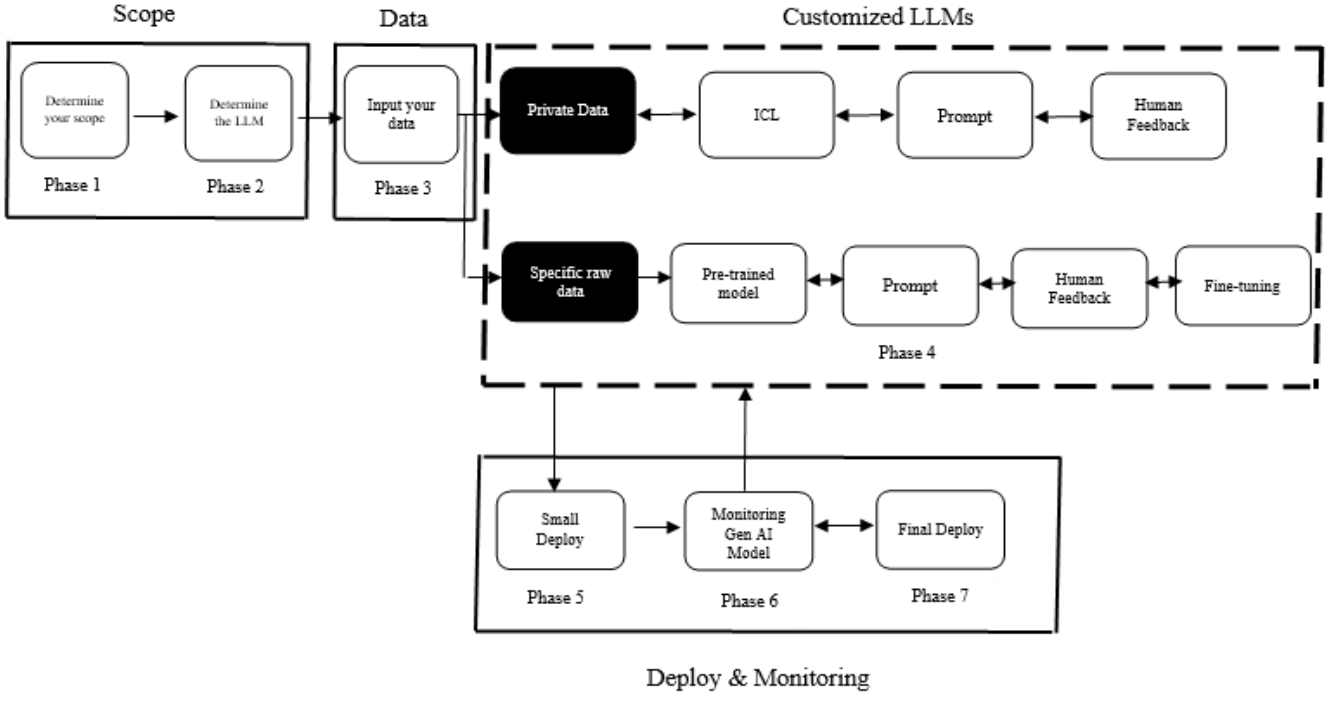

Figure 4: The Proposed Forward and Back Systematic Approach to do a Generative AI

Retrieval Number: 100.1/ijsce.D363614040924
DOI: 10.35940/ijsce.D3636.14030724
Journal Website: www.ijsce.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

IV. CONCLUSION

This exploration into Generative AI and its practical applications, particularly through the lens of Large Language Models like GPT-3 and BERT, underscores a significant leap forward in technological sophistication. Since text generation accuracy is one of the significant aspects of artificial intelligence, the primary concept behind the present research work lies in how to enhance it when data are in private or specific raw data. We focus on the process of data feeding in the customization phase in generative AI.Therefore, our proposed "Forward and Back Systematic Approach" not only harmonizes with the nuanced requirements of various applications but also ensures these models are not merely theoretical constructs. They are robust, efficient tools capable of real-world impact.By meticulously navigating through strategies like leveraging Private Generalized LLM APIs, adopting ICL, and embracing fine-tuning methodologies, the present paper illuminates a path to tailor these profound technologies to specific,case-driven needs. Ultimately, our journey through the multifaceted landscape of Generative AI reaffirms its transformative potential across sectors.As we stand at the pre-defined technological precipice, this paper not only contributes a consistent framework and critical analysis but also charts a course for responsible innovation and application to ensure that the promise of Generative AI is realized in an efficient, ethical, and impactful manner.

Further studies are suggested to perform our proposed approaches in experimental cases. One case is under the authors' new study in which the accuracy of the outputs is investigated. Focusing more on fine-tuning is another recommendation where advanced fine-tuning techniques may improve and swift this tuning stage. Integration with emerging technologies is also recommended to explore the integration of the proposed generative AI framework with emerging technologies like edge computing, federated learning, and blockchain.

ACKNOWLEDGMENTS

The authors of this paper would like to express their great appreciation and gratitude for the support received from Mr. Ali Pourjalili for his constructive comments and technical guidance.

DECLARATION STATEMENT

Funding	No, I did not receive.
Conflicts of Interest	No conflicts of interest to the best of our knowledge.
Ethical Approval and Consent to Participate	No, the article does not require ethical approval and consent to participate with evidence.
Availability of Data and Material	Not relevant.
Authors Contributions	All authors have equal participation in this article.

REFERENCE

1. Ng, A. and Jordan, M. (2001).On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes.In *Advances in Neural Information Processing Systems*, volume 14, pages 841—848.
2. Hu, L. (2023). Generative AI and Future. Retrieved on January 23 from https://pub.towardsai.net/generative-ai-and-future-c3b1695876f2.
3. Jovanović, M. (2023). Generative Artificial Intelligence: Trends and Prospects. https://www.computer.org/csdl/magazine/co/2022/10/09903869/1H0G6xvtREk.0.1109/MC.2022.3192720.
4. Abukmeil, M., Ferrari, S., Genovese, A., Piuri, V., & Scotti, F. (2021). A survey of unsupervised generativemodels for exploratory data analysis and representation learning. Acm computing surveys (csur), 54(5), https://doi.org/10.1145/3450963
5. Gui, J., Sun, Z., Wen, Y., Tao, D., & Ye, J. (2021). A review on generative adversarial networks: Algorithms, theory, and applications. IEEE Transactions on Knowledge and Data Engineering. doi: 10.1109/TKDE.2021.3130191. https://doi.org/10.1109/TKDE.2021.3130191
6. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33: 1877-1901.
7. Aydın, Ö., & Karaarslan, E. (2023). Is ChatGPT leading generative AI? What is beyond expectations? Academic Platform Journal of Engineering and Smart Systems. https://doi.org/10.2139/ssrn.4341500
8. Pavlik, J. V. (2023). Collaborating With ChatGPT: Considering the Implications of Generative Artificial Intelligence for Journalism and Media Education. Journalism & Mass Communication Educator, 0(0). https://doi.org/10.1177/10776958221149577
9. Kulkarni, A., Shivananda, A., Kulkarni, A., & Gudivada, D. (2023). *Applied Generative AI for Beginners: Practical Knowledge on Diffusion Models, ChatGPT, and Other LLMs*. https://doi.org/10.1007/978-1-4842-9994-4
10. Spitale, G., Biller-Andorno, N., and Germani, F. (2023).AI model GPT-3 (dis) informs us better than humans.*arXiv:2301.11924*. https://doi.org/10.1126/sciadv.adh1850
11. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023).Survey of hallucination in natural language generation.*ACM Computing Surveys*, 55(12):1–38. https://doi.org/10.1145/3571730
12. B. A. y Arcas, (2022). "Do large language models understand us?" *Daedalus*, vol. 151, no. 2, pp. 183–197. https://doi.org/10.1162/daed_a_01909
13. Zellers, R., Holtzman, A., Peters, M. E., Mottaghi, R., Choi, Y., Kuehlkamp, A., ... & Farhadi, A. (2023). A Survey for In-context Learning. *ar5iv*. Retrieved from https://ar5iv.labs.arxiv.org/html/2301.00234
14. Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El Showk, S., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., Kravec, S., Lovitt, L., Nanda, N., Olsson, C., Amodei, D., Brown, T. B., Clark, J., McCandlish, S., Olah, C., Mann, B., Kaplan, J. (2022). Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862.
15. Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Narang, S., Mishra, G., Yu, A., Zhao, V. Y., Huang, Y., Dai, A. M., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., & Wei, J. (2022). Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.
16. Guo, Z., Jin, R., Liu, C., Huang, Y., Shi, D., Supryadi, Linhao, Y., Liu, Y., Li, J., Xiong, B., & Xiong, D. (2023). Evaluating Large Language Models: A Comprehensive Survey. *arXiv*. https://arxiv.org/abs/2310.19736
17. Pichai, S. (2023).An important next step on our AIjourney.Accessed on 03/16/23.
18. Mehdi, Y. (2023). Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web. Official Microsoft Blog, 7.
19. Blodgett, S. L., Barocas, S., Daum'e, H., and Wallach, H. M. (2020).Language (technology) is power: A critical survey of "bias" in nlp.In *Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/2020.acl-main.485
20. Jones, E. and Steinhardt, J. (2022).Capturing failures of large language models via human cognitive biases.In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.
21. Adik, K. (2023, July 13). Generative AI Project Life Cycle. Medium. https://medium.com/@kanikaadik07/generative-ai-project-life-cycle-55ce9092e24a

22. Alenezi, M., & Almuairfi, S. (2019). Security Risks in the Software Development Lifecycle. In International Journal of Recent Technology and Engineering (IJRTE) (Vol. 8, Issue 3, pp. 7048–7055). https://doi.org/10.35940/ijrte.c5374.098319

AUTHORS PROFIE

Dr. Abbas Mahmoudabadiis a faculty member and director of the master's program in Industrial Engineering at MehrAstan University, Gilan, Iran. He received a Ph.D. in optimization in Hazmat transportation in January 2014, followed by publishing more than 90 papers in industrial engineering, public transportation, traffic safety, e-commerce, and medical statistics. Abbas teaches engineering courses in academic positionsas well asscientific cooperation with international agencies on traffic safety and industrial engineering. In his executive position, he has been working for30 years in traffic, road safety,and public transport planning in developing countries.

Razieh AkbariLalaei is a qualified professional with a bachelor's degree in Mathematics from Amir Kabir University, a master's degree in Industrial Engineering, and an MBA. Her expertise also includes a comprehensive course on Data Science, specializing in Big Data, Artificial Intelligence, Machine Learning, Deep Learning, and Information Systems. She has over six years of experience working on Digital Transformation, Industry 4.0, Data-Driven Decision Making for manufacturing companies, and recently in the applications of Generative AI. Moreover, she has served as a Teaching Assistant in Information Systems and E-Business for two semesters at the Iran University of Science & Technology.