International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 16 (2024) pp. 596–612

Article

Optimizing Automated Conversational Large Language Models for Higher Educational Institution

Mohammed Varaliya *, Mahendra Kanojia and Subhashish Nabajja

Sheth L.U.J and Sir M.V. College,400069 Mumbai, Maharashtra, India; kgkmahendra@gmail.com (M.K); subhashishnabajja1575643@gmail.com (S.N.)
* Correspondence author: email mohammedvaraliya2661392@gmail.com

Received date: 9 April 2024; Accepted date: 26 June 2024; Published online: 10 July 2024

Abstract: Higher education institutions need to improve their query systems in order to improve communication, response times, and information access. This study investigates a novel method of combining large language models (LLMs) with knowledge bases created especially for higher education institutions in order to meet this critical requirement. We adhered with all higher education institution regulations and gathered data in both structured and unstructured formats from Sheth L. U. J. College of Arts and Sir M. V. College of Science and Commerce in Mumbai, India. Our system, employing the Llama-Index framework for text embedding and the two primary large language models, Google Gemini 1.0 and OpenAI GPT-3.5, for response generation, achieved an impressive average response time of 5 seconds for both models. Additionally, it attained an average relevancy score of 0.96 for Google Gemini 1.0 and 0.93 for OpenAI GPT-3.5 across diverse query categories. These findings clearly show how LLM-powered systems can improve communication and offer incredibly helpful, relevant information in higher education settings. We strongly encourage the deployment of these systems in order to greatly improve communication and the user experience in the context of higher education. In order to further enhance the efficiency and coverage of these systems, we plan to expand our knowledge sources and host them on cloud platforms, making them easily accessible.

Keywords: large language models; chatbots; generative pretrained transformer; generative AI; natural language processing; question answering; retrieval augmented generation

1. Introduction

Intelligent human-computer interaction has entered a new era as a result of the significant evolution of natural language processing (NLP) and artificial intelligence (AI) over the past few decades. At the forefront of this revolution are chatbots—simple yet powerful AI systems that have become universal in

our digital lives. These chatbots are just one aspect of the enormous potential of AI systems, as demonstrated by their capacity to hold meaningful conversations with users [1].

Large Language Models (LLMs) have become very popular in addition to chatbots. These LLMs are examples of sophisticated artificial intelligence (AI) structures with never-before-seen capacities for large-scale human language production, understanding, and modification [2]. Long limited to being simply instruments of communication, LLMs are currently being widely applied in a wide range of sectors and domains, such as education [3], finance [4], and public health [5].

In this ever-evolving educational landscape, effective communication and seamless sharing of knowledge are crucial for higher education institutions. Consider the variety of people who are looking for information: teachers, staff, students who are currently enrolled, and potential students. Their questions are varied and range from simple ones concerning campus services to complex ones concerning academic programs, financial aid, and extracurricular activities. It is essential that these inquiries be given quick, precise responses.

International students find it extremely difficult to navigate the complex information landscapes of their host institutions because they are frequently located in distant time zones and face language barriers. These difficulties include following strange administrative protocols, maintaining restricted office hours, and possibly having to pay for communications. This can require overcoming a lot of obstacles, like making appointments, getting necessary information, and asking questions, which can make the process difficult and even intimidating. But what if there's a way to make the whole process run more smoothly? The chatbot is a digital assistant that can help in real time, anywhere in the world, at any time of day or night. These chatbots have the potential to completely change how higher education institutions communicate with their stakeholders by utilizing the power of LLMs.

This study has the potential to greatly empower inquirers by providing them with continuous access to important details about admissions, scholarships, and college life. By providing immediate and personalized responses to inquiries, regardless of time zone differences, the proposed LLM-based chatbots can bridge communication gaps and simplify administrative procedures. This can significantly reduce the challenges faced by international students navigating unfamiliar information landscapes within their visiting institutions.

The benefits don't stop there. These chatbots can do more than just respond to inquiries by utilizing LLMs; they can also learn from interactions, improve their responses, and eventually adjust to the changing needs of their users. Our proposed LLM-based chatbot that not only answers questions from users and delivers information, but also proactively offers advice and assistance. There has been an important change in the way we view the sharing of knowledge within educational institutions.

And that's precisely the focus of this research—to develop a responsive LLM-based chatbot tailored specifically for higher education institutions. Our objective? to improve the quickness and relevance of responses, opening up channels of communication and guaranteeing that important information is available to all individuals.

2. Literature Review

Large language models (LLMs) have become increasingly popular in recent years as researchers investigate their potential applications and implications in a variety of fields, including natural language processing (NLP). In this study we explore how integrating LLMs with custom knowledge bases can transform query systems in higher education institutions. We start with a thorough analysis of the relevant literature to set the scene for this research, emphasizing the developments and difficulties faced by LLMs, their use in education, and the fusion of chatbots and knowledge graphs. LLMs are a type of artificial intelligence (AI) trained on massive amounts of text data, enabling them to generate human-quality text, translate languages, write different kinds of creative content, and answer your questions in an informative way. Recent studies have provided light on the internal workings and limitations of LLMs. Ref. [6] started looking into the security concerns related to LLMs, with a particular emphasis on training data extraction—which is crucial for guaranteeing the robustness and dependability of these systems, their study laid the foundation for more research in this field by highlighting the need for techniques to improve the security and openness of LLMs. Similar to that, Ref. [7] addressed the effectiveness and adaptation of LLMs, investigating techniques to optimize these models for a wide range of NLP assignments, their research demonstrated how crucial it is to optimize LLMs in order to guarantee their effectiveness in a variety of applications. Ref. [2] made an important contribution to the field by focusing

on LLM conversion for biomedical writing, a field with particular cultural requirements, their work focused on improving techniques for optimizing LLMs such as BERT and GPT-3, providing insightful information about the details of biomedical natural language processing. Additionally, Ref. [8] investigated the security aspects of chatbots based on LLM and presented a technique known as "JAILBREAKER" to address potential security flaws, their research emphasized the necessity of strong security measures in AI-driven applications and the significance of defending LLM-based systems against malicious attacks. In a detailed analysis, Ref. [5] explored the ethical and data privacy concerns associated with using conversational AI powered by LLMs in public health interventions. Their study shed light on the moral considerations surrounding the use of LLMs in sensitive areas, prompting discussions on the responsible use of AI. The use of LLMs in education presents both opportunities and challenges, Ref. [3] addressed privacy concerns and emphasized the importance of personalized learning, their research provided important insights into the future of AI-driven learning environments and opened the door for further investigation into the possible uses of LLMs in educational settings. Furthermore, Ref. [9] presented "InternGPT", a unique framework that streamlines vision-centric tasks by integrating computer vision and natural language conversation, this creates new opportunities for research across disciplines, their research demonstrated how adaptable LLMs are at supporting multimodal interactions and bridging the gap between computer vision and language processing. As well, Ref. [10] carried out a comprehensive study on the evaluation of LLMs, offering insightful information on analyzing their effectiveness and moral implications in a variety of applications. In order to gain a greater understanding of the advantages and disadvantages of these three well-known LLMs for natural language processing tasks, Ref. [11] conducted a thorough performance comparison of OpenAI ChatGPT, Microsoft Bing Chat, and Google Bard, their research provided useful benchmarks for assessing LLM performance, assisting researchers in the selection of appropriate models for particular applications. In addition, Ref. [12] addressed the need for models with better accuracy and knowledge by improving in-context learning with answer feedback for multi-span question answering. "Language Models are Few-Shot Learners" introduces GPT-3, a 175 billion parameter language model that excels in various tasks with minimal prompting, often matching or surpassing fine-tuned state-of-the-art models, trained on a diverse dataset, GPT-3 showcases strong generalization across tasks like translation and question-answering, its performance and scalability underscore the potential and risks of large-scale models, emphasizing the need for responsible deployment [13]. Many studies have been conducted on the increasing number of transformer-based models in NLP, with studies such as [14] exploring the difficulties of knowledge graph matching, their research helped us understand transformer-based techniques better, which opened the door to the development of more reliable and effective NLP systems. Furthermore, an in-depth review of machine translation research was carried out by [15], with a particular emphasis on transformer-based models such as GPT, their analysis provides important light on these models' functionality and helped shape the course of future machine translation research. Ref. [16] research into text embedding optimization for a range of language comprehension applications offered fresh perspectives on enhancing the efficiency and effectiveness of language models, with the help of their research, LLMs are now more capable to understand and analyze human language. In order to improve long-distance relationship understanding and get around limitations in the context window of large language models, Ref. [17] looked into positional interpolation techniques, their research improved LLMs' capacity to gather contextual information from various document sections, addressing a crucial problem. In addition to these studies, there have been significant advances in specific domains, such as academic record monitoring in higher education institutions [18], and the advantages and disadvantages of using AI chatbots in higher education [19], these studies highlight the potential advantages and difficulties of using AI-driven systems in educational settings and offer insightful information about their practical applications. The effectiveness of reinforcement learning from human feedback (RLHF) in enhancing LLMs' responses to prompts related to mental health was demonstrated by [20], demonstrating the potential advantages and ethical concerns of AI in psychological treatment, their work highlights the significance of ethical and responsible AI deployment and creates new opportunities for utilizing AI-driven systems in mental health interventions. This paper [21] introduced innovative Gemini models, highlighting their versatility in processing diverse data types like images, text, and audio, thus enhancing AI systems' comprehension of various data modalities. The study contributed valuable insights to multimodal AI research, but further exploration is needed to address practical applications and potential limitations. In total, these studies contribute to a deeper understanding of LLMs and their various

applications, clearing the way for advancements in AI-driven technologies. Researchers are pushing the limits of what is possible with LLMs and influencing the direction of natural language processing and AI-driven applications by addressing important issues and investigating innovative techniques.

3. ETL (Extract Transform Load) of Data

The collection of data that we employed for our investigations was carefully selected from a variety of academic documents, with a particular focus on the academic documents of Sheth L. U. J. College of Arts and Sir M. V. College of Science and Commerce, located in Mumbai, India.

We made sure that ethical guidelines were followed throughout the data obtaining process in accordance with the rules that were put forth by the relevant higher education authorities. We used a thorough process to compile both structured and unstructured data formats in our dataset. This involved the manual transcription of relevant data from physical hard copies in addition to the integration of digital soft copies that were sourced from multiple online platforms and institutional databases.

We used a systematic approach to data collection, employing various scraping techniques such as Web scraping with Python libraries like Beautiful Soup [22], and Selenium [23], to obtain relevant information from a variety of educational sources from official institution website. Additional academic documents and reliable institutional databases such as the Academic course catalogs, schedules catalog, administrative procedures catalog, student guides catalog, and campus facility details from institution's library. A great effort was ensured to maintain the diversity and broadness by gathering a wide range of data.

However, there were challenges with the task. One of the main challenges we faced was the inconsistent way that data was presented in various sources. For example, the PDF format for the Academic Program details was selectable, while the PDF format for the Events detail was not selectable. In selectable PDFs, we used PDF text extraction tools to allow text to be highlighted and copied. By directly parsing the text from the PDF files, these tools enable us to programmatically extract the required data. For this, Python libraries such as PyPDF2 [24] or PDFMiner [25] were frequently utilized. When working with scanned documents or non-selectable PDFs, Optical Character Recognition (OCR) technology comes in handy. Due to the time-consuming and inconsistent nature of manual transcription, extracting data from physical hard copies proved to be a significant challenge. We used OCR software Online OCR [26] to get over with this obstacle. We were able to extract text from scanned images by converting them into machine-readable text using Online OCR [26]. Additionally, we extracted the text from actual hard copy catalogues such as Course catalogs, departmental brochures, information booklets, and handbooks, using the same OCR software. We also had to deal with data that was redundant across several resources. For instance, the college's events page and the official course catalog both provided information regarding academic schedules. In order to solve this problem, we employed a manual data merging procedure that helped us find and eliminate unnecessary data while maintaining the concision and manageability of our dataset. Ensuring data consistency presented a significant challenge due to variations in terminology, formatting, and even accuracy across different sources. In order to solve this, we kept in constant communication with college representatives. After reviewing and validating the extracted data, they clarified any confusion or gaps that appeared during the data collection and processing stages and confirmed that it was in line with official information. The reliability and precision of the final dataset were ensured in large part by collaboration with these officials, which included subject-matter expertise from college officials, data verification, and access permissions.

Implementing this comprehensive strategy, our goal was to build a solid repository that faithfully captures the complex character of university thorough approach to data collection, strict validation, and cooperation with college administrators. Our thorough work produced a dataset that accurately reflects the complex nature of our higher education institution, despite the difficulties presented by a variety of document types, overlapping data, and consistency issues. This dataset will be a valuable tool for academic research and investigations.

After gathering the data and carrying out certain preliminary preprocessing, the next critical step was to transform the unprocessed data into a structured format that could be used for computational analysis. This stage, where the source documents were divided into easily readable text sections.

Figure 1 illustrate the data ETL (Extract, Transform, Load) process, which provides a systematic framework for handling the collected information.

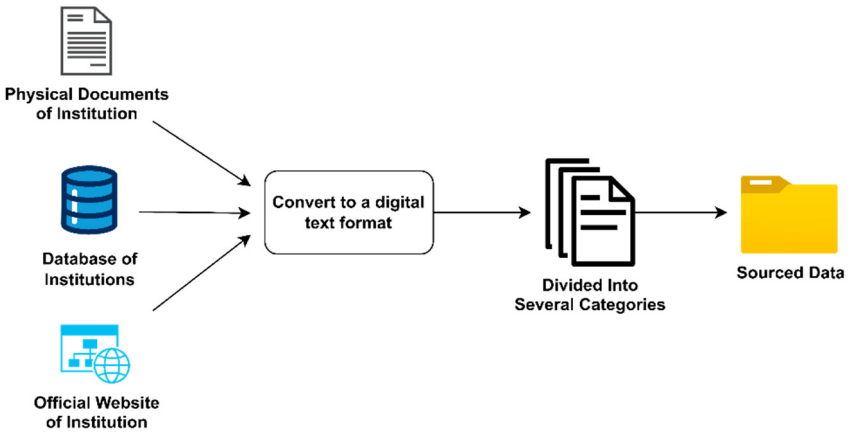

Figure 1. Data Staging.

After obtaining the unprocessed data, we turned our attention to preparing it for computer analysis. This involved transforming unprocessed data into sections of legible text, setting the stage for further data processing steps. To ensure accuracy and detail in our dataset, each text section was carefully chosen to represent a single coherent piece of information taken from the original documents. In addition to making data processing and analysis easier, this segmentation laid the groundwork for later stages of dataset improvement and refinement.

Our next step was to classify the data into distinct domains in order to make analysis and organization easier after the data had been converted into computational text format. The categories covered a wide range of topics, including location, facilities provided, academics, course offerings, extracurricular activities, and academic support. This structured approach to data categorization empowered us to efficiently analyze and retrieve relevant information, enhancing both the depth and broadness of our dataset.

The creation of a Sourced Data directory, which acts as the main storage location for all processed data files, was essential to our data management plan. The result of our struggles was gathered into this directory, which made it simple to manage and access the data we collected and processed over the course of the investigation. The careful attention to detail and precision, evident in the structure of each text part as an ordered unit of information extracted from the source documents, underscored the reliability and robustness of our dataset. This grouping made it easier to process and analyze data efficiently, laying foundations for later stages of dataset embedding.

4. Research Methodology

In order to address the challenges of revolutionizing query systems in higher education institutions, we carefully organized our research methodology by utilizing a combination of cutting-edge models. To be more precise, we used one framework and two main large language models to streamline different parts of our research procedure. An important part of embedding the text dataset and turning it into a numerical vector representation was done by the first framework, llama-index [27,28]. Through the process, unstructured textual data was converted into structured numerical embeddings, facilitating effective computational analysis and retrieval. Additionally, we utilized the OpenAI-GPT-3.5 model, known as “text-davinci-003” [29] as another foundation of our methodology. This powerful large-language-models enabled us to generate responses based on user queries with an excellent level of consistency and relevance. To further improve our approach’s capacity to generate responses in regards to user queries, we also integrated the Google Gemini 1.0 model [21]. Through utilizing all of these three models, our goal was to create a solid and comprehensive strategy for transforming query systems in higher education institutions. Throughout the following parts, we provide detailed descriptions of each model’s application and their respective contributions to our research framework.

4.1. LlamaIndex

Text embeddings are a natural language processing (NLP) technique that converts text into numerical vectors [30]. Texts with similar meanings have closer embeddings because embeddings capture semantic meaning and context. For instance, since they both describe similar contexts, the sentences “I took my dog to the vet” and “I took my cat to the vet” would have embeddings that are close to each other in the vector space [30].

LlamaIndex [27,28] uses embeddings to provide your documents with an advanced numerical representation. Text is fed into embedding models, and they produce an extensive set of numbers that represent the text’s semantics. There are numerous techniques to utilize when determining how similar two embeddings are (dot product, cosine similarity, etc.). LlamaIndex compares embeddings using cosine similarity by default [27].

However, we employed two embedding models to advance our findings. LlamaIndex [27] by default uses “text-embedding-ada-002" from OpenAI [29]; we used this embedding model for the openai-gpt-3.5 LLM Model because this model was specifically designed and optimized for compatibility with the OpenAI GPT-3 architecture, ensuring optimal performance and accurate representation of the text data within the LLM. Additionally, we used “models/embedding-001” [31] designed and optimized for the Google Gemini 1.0 LLM architecture for the Google Gemini 1.0 LLM Model [21], this strategic selection ensured compatibility and facilitated efficient embedding of text data, allowing the Gemini model to effectively understand and process the information.

4.1.1. List Index

The list index, a simple data structure with nodes arranged sequentially, is utilized during index construction. This process involves breaking down document texts into chunks, which are then transformed into nodes and organized into a list, as nodes are seen in Figure 2. When no additional query parameters are provided during query time, the LlamaIndex [27,28] loads every node in the list into the Response Synthesis module.

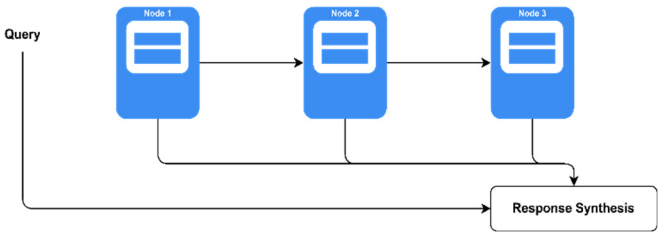

Figure 2. Response Synthesis Module [27].

The list index offers multiple querying methods, providing versatility in data retrieval. For instance, one can employ an embedding-based query to fetch the top k neighbors or incorporate a keyword filter, as illustrated in Figure 3 [27,28]. The lighter shade of the node in Figure 3 signifies its lack of relevance to the provided keyword, “population”. This feature of the list index proves beneficial when compiling a combined response from various data sources, facilitating comprehensive information synthesis.

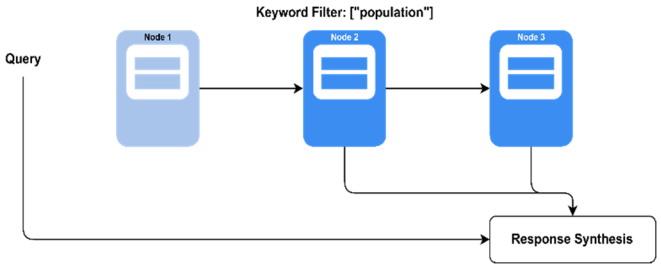

Figure 3. Keyword Filter [27].

4.1.2. Vector Store Index

An in-memory SimpleVectorStore [27,28] that is initialized as part of the default storage context is used by GPTVectorStoreIndex by default. Vector-store based indices create embeddings during index construction, in contrast to list indexes. In other words, in order to produce embeddings data, the LLM endpoint will be called during index construction. Finding the top k most similar Nodes and feeding them into our Response Synthesis module is the process of querying a vector store index.

When generating the overall response, this method makes sure that all potentially significant indices are taken into account Figure 4 [27].

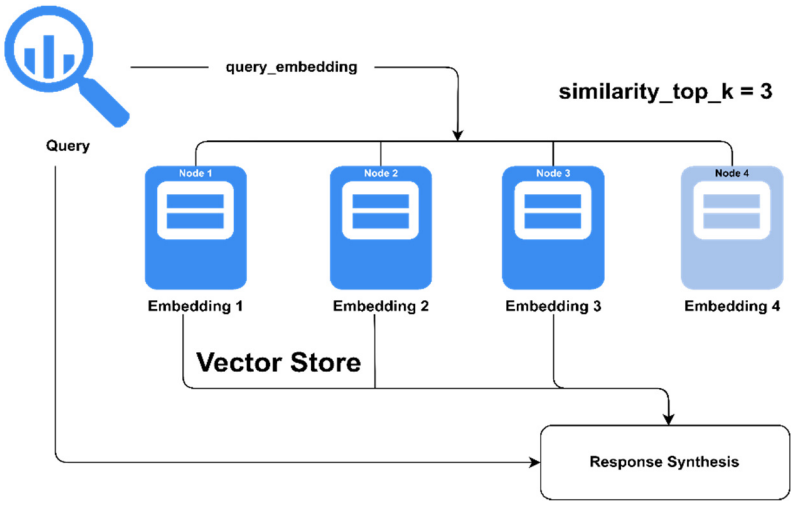

Figure 4. Querying a Vector Store Index [27].

4.2. Large Language Models

An algorithm for deep learning that can handle a wide range of natural language processing (NLP) tasks is called a large language model (LLM) [32]. Huge language models are large because they are trained on enormous datasets and employ transformer models. They can now recognize, translate, forecast, or create text or other content as a result.

The most popular architecture for a large language model is the transformer model. It is made up of a decoder and an encoder. Tokenizing the input and simultaneously solving mathematical equations to find relationships between tokens are how a transformer model processes data. This allows the computer to recognize patterns that a human would recognize if it were asked the same question [32].

4.2.1. The Transformer Architecture

The Transformer architecture uses an encoder-decoder structure, it produces outputs without the use of transformations or a future occurrence [33,34].

To put it simply, an input sequence is mapped by the encoder, located on the left half of the Transformer architecture, to a sequence of continuous representations, which is subsequently fed into a decoder see Figure 5. In order to produce an output sequence, the decoder, located on the right half of the architecture, receives both the encoder's and the decoder's output from the previous time step [33,34].

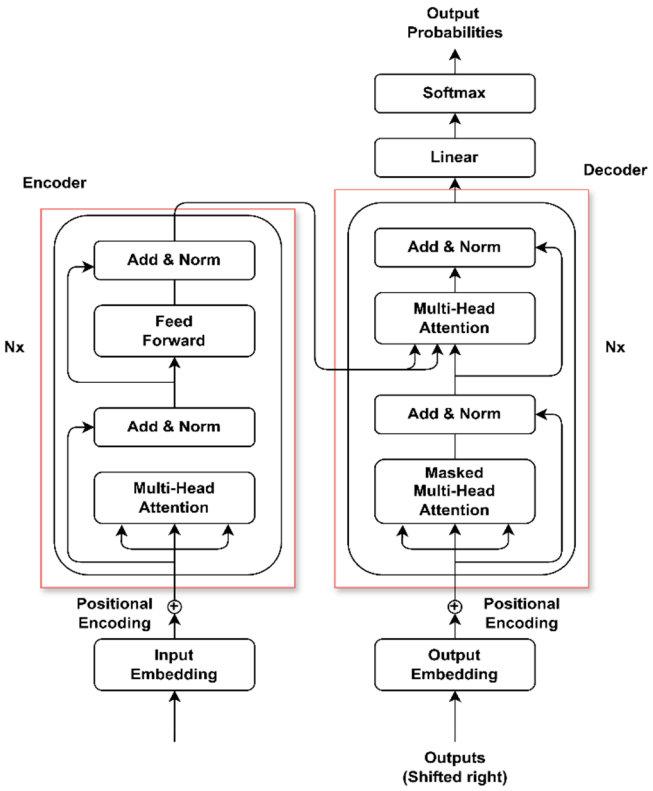

Figure 5. The Transformer-model architecture [33].

4.2.2. The Encoder

The encoder is made up of a stack of $N = 6$ identical layers, with two sublayers in each layer:

1. A multi-head self-attention mechanism is implemented by the first sublayer. The Transformer attention mechanism employs a multi-head implementation wherein h heads receive separate linearly projected versions of the queries, keys, and values. These heads h then produce parallel outputs, each of which is utilized to produce a final result [33,34].
2. The second sublayer is a feed-forward network that is fully connected and consists of two linear transformations with the activation of Rectified Linear Units (ReLU) in between

$$\mathrm{FFN}(x) = \mathrm{ReLU}(\mathbf{W}_1 x + b_1)\mathbf{W}_2 + b_2 \quad (1)$$

Each of the six Transformer encoder layers uses a different set of weight $\mathrm{W}_1, \mathrm{W}_2$ and bias b_1, b_2 parameters to apply the same linear transformations to every word in the input sequence [33,34]. Furthermore, there is a remaining connection surrounding each of these two sublayers. A normalization layer called layernorm(.), follows each sublayer and attempts to normalize the sum of the values obtained from the sublayer's input, x, and its own output, sublayer(x):

$$\mathrm{layernorm}(x + \mathrm{sublayer}(x)) \quad (2)$$

It is crucial to remember that because the Transformer architecture does not use repeated events, it is unable to automatically record any information regarding the relative positions of the words in the sequence. The input embeddings need to have positional encodings added in order to introduce this information. The sine and cosine functions of various frequencies are used to create the positional encoding vectors, which have the same dimension as the input embeddings. The positional information is subsequently added by simply adding their sum to the input embeddings.

The Decoder

The decoder shares a number of characteristics with the encoder. In addition, a stack of $N = 6$ identical layers, each made up of three sublayers, makes up the decoder.

1. The first sublayer gets the previous output of the decoder stack, improves it with positioning information, and executes multi-head self-attention over it. The decoder is altered to only focus on the words that come before it, whereas the encoder is made to process all of the words in the input sequence, regardless of where they are in the sequence. Therefore, the prediction for a word at position i can only rely on the outputs that are known for the words that are in front of it in the sequence. This is accomplished in the multi-head attention mechanism (which implements multiple, single attention functions in parallel) by covering the values obtained from the scaled multiplication of matrices Q and K with a mask. The matrix values that would typically correspond to unauthorized connections are removed in order to implement this masking [33,34].

$$\text{mask}(\mathbf{QK}^T) = \text{mask}\left(\begin{bmatrix} e_{11} & e_{12} & \dots & e_{1n} \\ e_{21} & e_{22} & \dots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \dots & e_{mn} \end{bmatrix}\right) = \begin{bmatrix} e_{11} & -\infty & \dots & -\infty \\ e_{21} & e_{22} & \dots & -\infty \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \dots & e_{mn} \end{bmatrix} \quad (3)$$

2. A multi-head self-attention mechanism similar to the one found in the encoder's first sublayer is implemented in the second layer. This multi-head mechanism on the decoder side gets the keys and values from the encoder's output as well as the queries from the previous decoder sublayer. This enables the decoder to process every word in the input string.
3. A fully connected feed-forward network, similar to the one used in the encoder's second sublayer, is used in the third layer of the decoder. The second layer's output is processed by this network, which also refines it further to help in the overall decoding process [33,34].

Moreover, a normalization layer succeeds the three sublayers on the decoder side, which also have remaining connections surrounding them.

4.2.3. OpenAI GPT-3 Architecture

The OpenAI GPT-3 model "text-davinci-003" [29] is a large language model trained on an extensive dataset of text and code. With 175 billion parameters, this transformer-based model is among the most substantial and powerful language models ever developed. Because of its versatility, GPT-3 can be used for a number of tasks, such as question answering, translation, and text generation [35]. The ability of transformers to identify long-range dependencies in sequential data is well known. The GPT-3 language model has a lot of information. Given some input text, it can probabilistically predict which letters from a known vocabulary will make an appearance next [36].

A comprehensive overview of the sizes, architectures, and learning hyper-parameters of the different OpenAI-developed GPT-3 models is given in Table 1. Essential metrics such as the number of parameters (nparams), number of layers (nlayers), model dimension (dmodel), number of heads (nheads), dimension of heads (dheads), batch size, and learning rate are used to characterize each variant of the model, which ranges from GPT-3 Small to GPT-3 175B [36]. The complexity of the model is indicated by the number of parameters, and its depth and feature extraction ability are determined by the number of layers. The dimensions of the model and parameters related to the head affect how well the model recognizes complex patterns and pays attention to relevant information in the input sequence. Training efficiency and convergence are highly dependent on batch size and learning rate. By looking at these variables, we were able to decide which models and training approaches would work best for our goal.

With the exception of using alternating dense and sparse attention patterns, the OpenAI GPT-3 family of models is based on the same transformer-based architecture as the GPT-2 model, including updated initialization, pre-normalization, and reverse tokenization [36]. The largest GPT-3 175B, also known as "GPT-3", contains 175 B parameters, 96 attention layers, and a batch size of 3.2 M. The GPT family uses the decoder half, as opposed to language models like BERT [37], which use the encoder to construct embeddings from raw text that may be used in other machine learning applications.

Table 1. Sizes, architectures, and learning hyper-parameters of the GPT-3 models, [36].

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{heads}	Batch Size	Learning Rate
GPT-3 Small	125	12	768	12	64	0.5 M	6.0×10^{-4}
GPT-3 Medium	350	24	1024	16	64	0.5 M	3.0×10^{-4}
GPT-3 Large	760	24	1536	16	96	0.5 M	2.5×10^{-4}
GPT-3 XL	1.3 B	24	2048	24	128	1 M	2.0×10^{-4}
GPT-3 2.7B	2.7 B	32	2560	32	80	1 M	1.6×10^{-4}
GPT-3 6.7B	6.7 B	32	4096	32	128	2 M	1.2×10^{-4}
GPT-3 13B	13.0 B	40	5140	40	128	2 M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0 B	96	12288	96	128	3.2 M	0.6×10^{-4}

4.2.4. Google Gemini Architecture

Gemini models are based on Transformer decoders [33], which have been improved with better architecture and model optimization to allow for efficient inference on Google's Tensor Processing Units and reliable training at scale [21]. They use effective attention mechanisms, such as multi-query attention, to accommodate 32k context lengths [38].

As shown in Table 2, the initial release, Gemini 1.0, comes in three primary sizes that can handle a variety of uses.

Table 2. An overview of the Gemini 1.0 model family [21].

Model Size	Model Description
Ultra	This is the most advanced model, able to perform at the cutting edge in a variety of extremely demanding tasks, such as multimodal and reasoning tasks. The Gemini architecture makes it efficiently severable at scale using TPU accelerators.
Pro	An efficient model that offers outstanding efficiency on a variety of jobs and is optimized for both cost and latency. This model demonstrates extensive multimodal capabilities and high reasoning performance.
Nano	The most effective model, intended for on-device use. With 1.8B (Nano-1) and 3.25B (Nano-2) parameters, we trained two different Nano versions, one for low memory devices and the other for large memory devices. It learns from larger Gemini models through distillation. It offers best-in-class performance and is 4-bit quantized for deployment.

Gemini models can process a wide range of inputs, including text, images, charts, screenshots, PDFs, and videos, and can produce outputs that are both text- and image-based [21]. Drawing inspiration from earlier models such as Flamingo [39], CoCa [40], and PaLI [41], Gemini is unique because it is fundamentally multimodal, able to generate images with discrete image tokens [21]. Although the model can handle a wide range of inputs, we concentrate on its exceptional text processing and output generation capability for the purposes of this study see Figure 6. This targeted approach allows us to explore and harness the full potential of Gemini models in the context of text-based tasks.

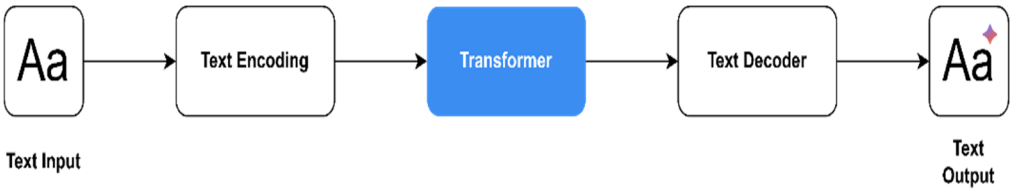

Figure 6. Extracted text model architecture from the Gemini 1.0 model architecture [21].

5. Proposed System

This article presents a proposed system that uses large language models (LLMs) and custom knowledge bases to revolutionize query systems in higher education institutions. Semantic index and strong LLM's are the two main components of this novel approach.

We discuss the prompts used in this study to guarantee response relevance. The first prompt, employed across various categories such as location, facilities provided, academics, course offerings, extracurricular activities, and academic support, which was stored in our sourced in data preprocessing phase, sets the stage for context-aware responses:

The Prompt 1: Custom Knowledge Base Chatbot Prompt Template mentioned below is a fundamental component that guarantees the applicability of the answers produced by our model. Our model is able to obtain a thorough understanding of the context by being given instructions that are specifically tailored to the content category, including location, facilities provided, academics, course offerings, extracurricular activities, and academic support. For example, when a user asks where MVLU College is located, the model uses the stored data in the sourced data that is unique to the "location" category. In a comparable manner, the model uses the relevant data to provide an accurate response when a question relates to academic support. An instance of this could take place if someone asks the model if teaching services are available, and the model replies by providing information about the academic support services that MVLU College provides.

```
The below content enclosed in triple ``` is all about our college <Category go here>, which is located in India
```
{Category Content}
```
This was all that MVLU College had to teach you about this particular {category}, and now you have to behave like custom knowledge base chatbot for MVLU college.

Now respond to the user's query.
```

Prompt 1. Custom Knowledge Base Chatbot Prompt Template.

The Prompt 2: Contextual Answering and Clarification Template, this prompt template guarantees that the responses from our model adhere with the given context, which improves accuracy and consistency. The model uses this template to ask the user for more information when there is not enough context for the question or when there is not enough data to properly answer it. For example, when a user asks, "What are the facilities available at MVLU College?", and doesn't specify which kind of facilities they're asking about, the model replies, "I'm sorry, I'm not sure what facilities you are referring to. Could you please provide more context or information?": By reducing the possibility of giving out incorrect or irrelevant information and encouraging users to submit thorough queries, this strategy raises the query system's overall efficiency.

```
Answer the question based on the context provided.

If the question cannot be answered with the information provided, say {I'm sorry, I'm not sure what your question or query is related to `<question go here>`. Could you please provide more context or information?}

Adhere to the above regulation strictly.
```

Prompt 2. Contextual Answering and Clarification Template.

We make sure that our system provides precise, relevant, and contextually appropriate answers to user inquiries by integrating these two prompt templates into the response generation process of our model. This improves user experience and increases confidence in the information provided.

5.1. Embedding Phase

After the dataset was segmented, the crucial step of embedding occurred, which involved converting textual data into numerical representations. We used state-of-the-art embedding models like "text-embedding-ada-002" from OpenAI [29], and "models/embedding-001" from the Google [31] by using framework called llama-index [27] to achieve converting textual data into numerical representations.

As illustrated in Figure 7, outlines the sequential steps involved in transforming the sourced data into embedding vector representations.

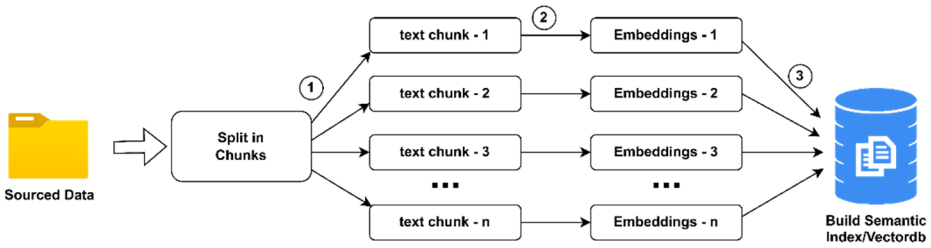

Figure 7. Data Embedding Process [42].

1. Data Chunk Segmentation:
   a. Segmenting the data into small text chunks is the first step in the embedding process.
   b. Every chunk is a complete informational unit taken directly out of the original source documents.
   c. The data is divided into more manageable chunks by this segmentation process, which makes processing and analysis more effective.
2. Conversion to Embedding Vector Representation:
   a. The individual text chunks are transformed into embedding vector representations after chunk segmentation.
   b. For this, cutting-edge embedding models like "text-embedding-ada-002" from OpenAI [22], and "models/embedding-001" from the Google [31] were utilized.
   c. The textual content's semantic meaning and context are captured by the numerical embeddings produced by these models.
   d. We improve the query-based retrieval processes' computational effectiveness and efficiency by transforming textual data into numerical vectors.
3. Storage in Vector Database (Semantic Index):
   a. After that, the generated embedding vectors are kept in a vector database, sometimes referred to as a semantic index.
   b. Our query system is built around this vector database, which allows semantic searches based on user queries.
   c. Retrieving relevant details quickly and effectively is made easier by storing the data in an organized vector format.
   d. Users can easily access the desired data because the semantic index simplifies the retrieval process.

Through data segmentation, numerical embedding, and vector database storage, we guarantee that the dataset is optimized for effective query-based retrieval. This stage is essential to improving our dataset's computational power and allowing users to efficiently extract relevant insights and data.

5.2. Response Generation Phase

LLMs used in our proposed system, which consists of Google's Gemini 1.0 model [21] and OpenAI's GPT-3 "text-davinci-003" model [29]. These LLMs are well known for their proficiency in a range of language-related tasks, including question answering, translation, and text creation [35].

Figure 8 illustrates the answer retrieval process, which clarifies our proposed system's operational workflow. The steps involved are broken down as follows:

1. First inquirer uses the LLM graphical user interface to ask a query.

2. A query embedding process using the "llama_index" model class GPTSimpleVectorIndex [27]. is initiated when an enquirer poses a question. Through this process, the query text is transformed into readily comprehensible numerical vector representations for the system.
3. Next, the system searches the knowledge base/vectordb for the query.
4. A list of results that are ranked is returned by the knowledge base/vectordb.
5. Next, the system transmits the ranked results and the processed query to the LLM Generative AI, which in our case consists of the Google Gemini 1.0 model [21] and the OpenAI GPT-3 "text-davinci-003" model [29].
6. Based on the previously ranked results, the LLM Generative AI responds to the enquirer's query with a comprehensive response that is sent back to the LLM Graphical user interface.
7. The LLM Graphical user interface then shows the generated response to the inquirer, satisfying their request for information.

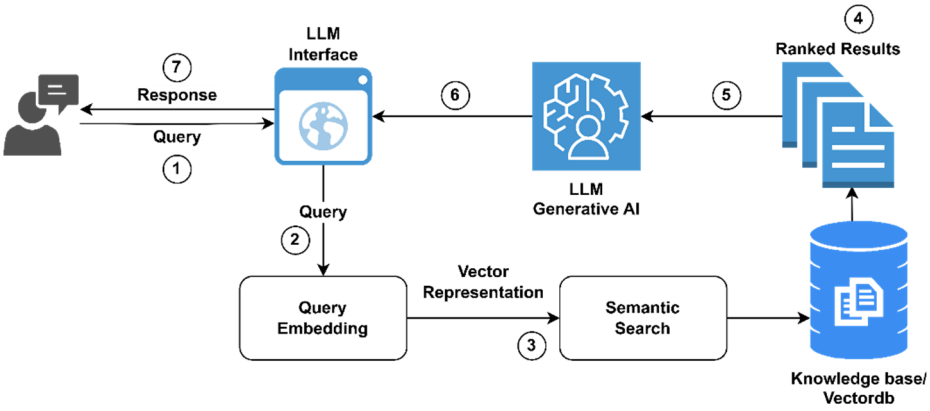

Figure 8. Answer Retrieval Process [42].

Our system's main objective is to provide users with highly relevant and educational responses to their inquiries, thereby improving the effectiveness and accessibility of query systems used by higher education institutions. Our goal is to make information retrieval more efficient by integrating state-of-the-art LLMs with customized knowledge bases, which will in turn encourage an informed and active educational community.

The graphical user interface (GUI) of our higher education institution's query system is shown in Figure 9 above. Through this GUI, students, staff, and other inquirers can communicate with chatbots and have their questions answered.

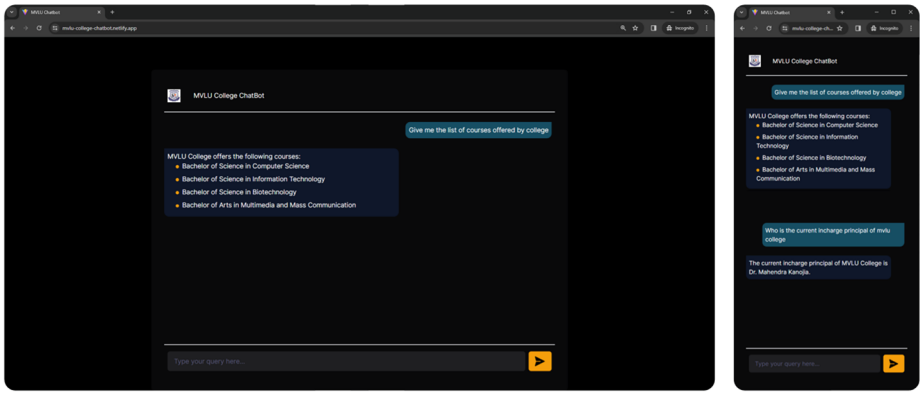

Figure 9. GUI of our Query System for Higher Education Institution.

6. Results & Discussion

The proposed approaches for integrating large language models with custom knowledge bases to transform query systems in higher education institutions went through an in-depth evaluation. Each large language model Google Gemini 1.0 [21] and the OpenAI GPT-3 "text-davinci-003" [29] received 300 questions covering a wide range of topics, such as location, facilities offered, academics, course offerings, extracurricular activities, and academic support.

The average response time for both models was 5 seconds. Furthermore, responds with an average relevancy of 0.93 were received by the OpenAI GPT-3 model "text-davinci-003" [22] and 0.96 were received by the Google Gemini 1.0 model [21].

$$X_{relevancy} = \frac{\sum_{i=1}^{n} x_i}{n}, \quad (4)$$

The average relevancy of the responses was determined using the formula above. The variable $X_{relevancy}$ denotes the overall average relevance factor, while x_i indicates the average relevancy of response *i*, and n indicates the total number of questions asked by the enquirer, which in this case is 300.

We also examined the models' performance on the same 300 queries that were posed to both the models was notable, with 50 queries in each category, as shown in Figure 10. This illustrates how well the model works to provide relevant and educational responses to a wide range of questions about higher educational institutions.

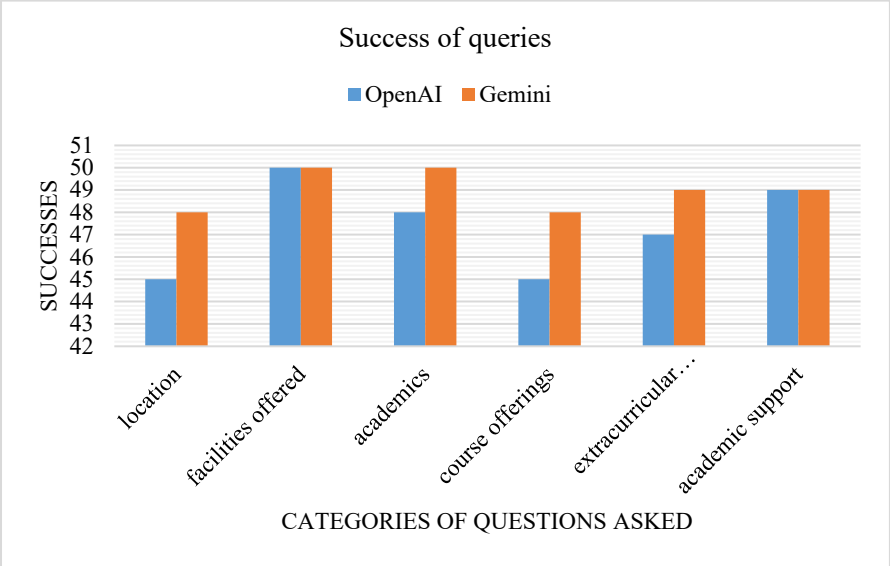

Figure 10. Success of queries.

The LLM OpenAI GPT-3 model and the Google Gemini 1.0 model are compared in Table 3. The OpenAI GPT-3 [22] has attained an average relevancy of 0.93, while the model's average success rate was 47.5. On the other hand, Google Gemini 1.0 [21] reported an average relevancy of 0.96, and the model's average success rate was 49.

Table 3. Comparison of both Large Language Models.

Model Name	Average Relevancy of LLM	Average Success Rate of LLM
OpenAI GPT-3	0.93	47.5
Google Gemini 1.0	0.96	49

Moreover, the model demonstrated the ability to process queries that were not consistent with the knowledge base customized for the specific university under review. When a query was asked that was not covered by the established knowledge base, the model responded with an appropriate message, like "I'm sorry, I'm not sure what your question or query is related to `<question go here>`. Could you please

provide more context or information?" This feature is essential because it prevents responses from being generated that are inaccurate or misleading, maintaining the accuracy and dependability of the data supplied. This demonstrates how well our models follow their intended objectives and function within limits that have been established through prompt engineering.

The online setup was used to implement both models. The cost of 1k tokens, which make up the OpenAI GPT-3 "text-davinci-003" [29] model, is $0.0200. And right now, Google Gemini 1.0 [21] model is free of cost, which means the company does not charge for it.

Upon evaluating both the models, we found that, the Google Gemini 1.0 [21] consistently outperforms the OpenAI GPT-3 model [29]. The difference performance is relatively high when the models are used with custom knowledge base. This implies that the Google Gemini 1.0 [21] model is more adept at using customized knowledge bases to produce accurate and relevant results, which improves the query system's overall efficiency in higher education.

7. Conclusions and Recommendations

In conclusion, our work presents a novel way to improve query systems in higher education by utilizing large language models (LLMs) and custom knowledge bases. Our evaluation of both the Google Gemini 1.0 [21] model and OpenAI GPT-3 "text-davinci-003" [29] model revealed promising results, with both models demonstrating an average response time of 5 seconds and relevancy score of 0.96 and 0.93. Especially when used as a custom knowledge base chatbot, the Google Gemini 1.0 model performed exceptionally well, demonstrating its improved capacity to provide relevant and accurate responses. This demonstrates how LLM-powered chatbots have the ability to completely transform how people communicate and access information in educational settings.

Moreover, combining data from different sources and storing knowledge datasets on servers might enhance the chatbot's functionality and broaden its application. Our research provides a solid foundation for the future creation of responsive LLM chatbots, which could provide users with relevant and accurate information in a range of industries, including healthcare, education, and customer service.

8. Future Scope

This research demonstrates the potential of integrating custom knowledge bases with large language models for revolutionizing query systems in various domains beyond educational institutions. Future work can explore the application of this approach to specialize chatbots in diverse fields, such as healthcare, customer service, and government services. This can significantly improve user experience and efficiency by providing targeted and accurate information even during periods of high demand or staff unavailability. For instance, in healthcare, a chatbot could answer basic medical questions at odd hours, reducing patient anxiety and directing them to appropriate resources. Additionally, incorporating reinforcement learning can further enhance the system's capabilities by enabling it to learn from user interactions and continually improve the accuracy and relevance of its responses. By continuously learning from user feedback, the model can adapt and refine its responses, leading to a more personalized and effective user experience. This research paves the way for the development of intelligent and interactive chatbots that can significantly improve information access and communication across various domains, potentially leading to benefits like reduced customer churn in service industries.

Author Contributions
Conceptualization, M.V., S.N. and M.K.; methodology, M.V.; software, M.V., S.N.; validation, M.V., S.N. and M.K.; formal analysis, M.V.; investigation, M.V.; resources, M.V.; data curation, M.V., S.N.; writing original draft preparation, M.V.; writing review and editing, M.K. and M.V.; visualization, M.V.; supervision, M.K.; project administration, M.K.; funding acquisition, M.K. All authors have read and agreed to the published version of the manuscript.

Funding
This research received no external funding.

Conflict of Interest Statement
The authors declare no conflicts of interest.

Data Availability Statement

The dataset used in this study is not publicly available due to confidentiality concerns. The data was specifically designed for our LLM chatbot to serve the needs of our higher education institution, Sheth L. U. J. College of Arts and Sir M. V. College of Science and Commerce in Mumbai, India, making the protection of this information our top priority.

References

1. Bansal, H., & Khan, R. A. (2018). A review paper on Human Computer interaction. *International Journal of Advanced Research in Computer Science and Software Engineering*, 8(4), 53. https://doi.org/10.23956/ijarcsse.v8i4.630
2. Tinn, R., Cheng, H., Gu, Y., Usuyama, N., Liu, X., Naumann, T., ... & Poon, H. (2023). Fine-tuning large neural language models for biomedical natural language processing. *Patterns*, 4(4).
3. Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., ... & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and individual differences*, 103, 102274.
4. Pavlyshenko, B. M. (2023). Financial News Analytics Using Fine-Tuned Llama 2 GPT Model. *arXiv* arXiv:2308.13032.
5. Jo, E., Epstein, D. A., Jung, H., & Kim, Y. H. (2023, April). Understanding the benefits and challenges of deploying conversational AI leveraging large language models for public health intervention. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (pp. 1-16).
6. Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., ... & Raffel, C. (2021). Extracting training data from large language models. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security 21)* (pp. 2633-2650).
7. Xu, R., Luo, F., Zhang, Z., Tan, C., Chang, B., Huang, S., & Huang, F. (2021). Raise a child in large language model: Towards effective and generalizable fine-tuning. *arXiv,* arXiv:2109.05687.
8. Deng, G., Liu, Y., Li, Y., Wang, K., Zhang, Y., Li, Z.et al. (2023). Jailbreaker: Au-tomated Jailbreak Across Multiple Large Language Model Chatbots. *arXiv,* arXiv:2307.08715.
9. Liu, Z., He, Y., Wang, W., Wang, W., Wang, Y., Chen, S., ... & Qiao, Y. (2023). Intern-chat: Solving vision-centric tasks by interacting with chatbots beyond language. *arXiv*, arXiv:2305.05662.
10. Chang, Y., Wang, X., Wang, J., Wu, Y., Zhu, K., Chen, H., et al. (2023). A survey on evaluation of large language models. *arXiv* arXiv:2307.03109.
11. Dao, X. Q. (2023). Performance comparison of large language models on vnhsge english dataset: Openai chatgpt, microsoft bing chat, and google bard. *arXiv* arXiv:2307.02288.
12. Huang, Z., Zhou, J., Xiao, G., & Cheng, G. (2023). Enhancing In-Context Learning with Answer Feedback for Multi-Span Question Answering. *arXiv* arXiv:2306.04508.
13. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P.,et al. (2020). Language Models are Few-Shot Learners. arXiv (Cornell University). Retrieved from https://arxiv.org/pdf/2005. 14165.pdf
14. Hertling, S., Portisch, J., & Paulheim, H. (2022). KERMIT—A Transformer-Based Ap-proach for Knowledge Graph Matching. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2204.13931
15. Hendy, A., Abdelrehim, M., Sharaf, A., Raunak, V., Gabr, M. A., Matsushita, H., Kim, Y. J., Afify, M., & Awadalla, H. H. (2023b). How good are GPT models at machine Transla-tion? A comprehensive evaluation. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2302.09210
16. Su, H., Kasai, J., Wang, Y., Hu, Y., Ostendorf, M., Yih, W. T., ... & Yu, T. (2022). One embedder, any task: Instruction-finetuned text embeddings. *arXiv* arXiv:2212.09741.
17. Chen, S., Wong, S., Chen, L., & Tian, Y. (2023). Extending context window of large lan-guage models via positional interpolation. *arXiv* arXiv:2306.15595.
18. Heryandi, A. (2020). Developing Chatbot For Academic Record Monitoring in Higher Edu-cation Institution. *IOP Conference Series*, 879(1), 012049. https://doi.org/10.1088/1757-899x/879/1/012049
19. Yang, S., & Evans, C. (2019, November). Opportunities and challenges in using AI chat-bots in higher education. In *Proceedings of the 2019 3rd International Conference on Educa-tion and E-Learning* (pp. 79-83).
20. Bill, D., & Eriksson, T. (2023). Fine-tuning a LLM using Reinforcement Learning from Human Feedback for a Therapy Chatbot Application.
21. Team, G. H. C., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., Silver, D., Petrov, S., Johnson, M., Antonoglou, I., Schrittwieser, J., Glaese, A., Chen, J., Pitler, E., et al. (2023). Gemini: a family of highly capable multimodal models. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2312.11805.
22. beautifulsoup4. PyPI. Available at: https://pypi.org/project/beautifulsoup4/ (Accessed on date: 17 January 2024).
23. selenium. PyPI. Available at: https://pypi.org/project/selenium/ (Accessed on date: 17 March 2024).
24. *PYPDF2*. PyPI. Available at: https://pypi.org/project/PyPDF2/ (Accessed on date: 31 December 2022)
25. *pdfminer*. PyPI. Available at: https://pypi.org/project/pdfminer/ (Accessed on date: 25 November 2019)
26. *Free Online OCR - Image to text and PDF to Doc converter*. (n.d.). Available at: https://www.onlineocr.net/

27. *LlaMaIndex V0.10.13.* (n.d.). Available at: https://docs.llamaindex.ai/en/stable/ (Accessed on date: 20 May 2023).
28. Zirnstein, B. Extended context for InstructGPT with LlamaIndex.
29. OpenAI. (n.d.). OpenAI Website. Available at: https://openai.com/ (Accessed on date: 25 May 2023).
30. *Embeddings guide*. (n.d.). Google AI for Developers. Available at: https://ai.google.dev/docs/embeddings_guide (Accessed on date: 10 January 2024).
31. *Google* AI for Developers. (n.d.). Google AI. Available at: https://ai.google.dev/ (Accessed on date: 10 Jan 2024).
32. *What is a Large Language Model? | A Comprehensive LLMs Guide*. (n.d.). Elastic. Available at: https://www.elastic.co/what-is/large-language-models (Accessed on date: 10 January 2024).
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *arXiv (Cornell University)*, *30*, 5998–6008. https://arxiv.org/pdf/1706.03762v5.
34. Available at: https://machinelearningmastery.com/the-transformer-model/ (Accessed on date: 20 Feb 2024).
35. Contributeurs aux projets Wikimedia. (2023). GPT-3. fr.wikipedia.org. https://fr.wikipedia.org/wiki/GPT-3
36. OpenAI GPT-3: Understanding the Architecture. (2022, May 4). The AI Dream. Available at: https://*www*.theaidream. com/post/openai-gpt-3-understanding-the-architecture (Accessed on date: 28 August 2023).
37. Srivastava, H., Varshney, V., Kumari, S., & Srivastava, S. (2020, July). A novel hierar-chical BERT architecture for sarcasm detection. In *Proceedings of the Second Workshop on Figurative Language Processing* (pp. 93-97).
38. Shazeer, N. (2019). Fast Transformer Decoding: One Write-Head is All You Need. *arXiv (Cornell University)*. https://arxiv.org/pdf/1911.02150v1.
39. Alayrac, J., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Arthur, M., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Simonyan, K. (2022). Flamingo: a Visual Language Model for Few-Shot Learning. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2204.14198.
40. Yu, J., Wang, Z., Vasudevan, V. K., Yeung, L., Seyedhosseini, M., & Wu, Y. (2022). CoCa: Contrastive Captioners are Image-Text Foundation Models. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2205.01917.
41. Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., Kolesnikov, A. I., Puigcerver, J., Ding, N., Rong, K., Akbari, H., Mishra, G., Xue, L., Thapliyal, A., Bradbury, J. T., et al. (2022). PALI: a Jointly-Scaled Multilingual Language-Image Model. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2209.06794.
42. Varaliya M, Kanojia M, Nabajja S (2023). Revolutionizing higher education institute query system by linking custom knowledge base with large language models. In *Hybrid Intelligent System*. Springer.
43. Eloundou, T., Manning, S., Mishkin, P., & Rock, D. (2023). Gpts are gpts: An early look at the labor market impact potential of large language models. *arXiv* arXiv:2303.10130.