Article

PixieGPT: Design and Implementation of a Generative Pre-Trained Transformer for Universities of Bangladesh

Hasan Mahmood Aminul Islam *, Mehedi Hasan *, Sumiaya Ahmed *, Ariful Islam Fardin *, Mehedi Hasan Nabil *

Posted Date: 20 February 2024

doi: 10.20944/preprints202402.1083.v1

Keywords: NLP; BeRT; PixieGPT

Article

PixieGPT: Design and Implementation of a Generative Pre-Trained Transformer for Universities of Bangladesh

Hasan M A Islam*, Mehedi Hasan, Sumiaya Ahmed, Ariful Fardin, Mehedi Nabil

* Correspondence: hasan.mahmood@ewubd.edu
† Current address: east West University, Dhaka, Bangladesh
‡ These authors contributed equally to this work.

Abstract: In a densely populated country like Bangladesh, universities grapple with the challenge of efficiently addressing myriad queries from a large student body, leading to a heightened workload for university stakeholders. To tackle these challenges, we introduce PixieGPT, a tailor-made Generative Pre-Trained Transformer for Bangladeshi universities. PixieGPT significantly mitigates workload by adeptly handling common university-related queries, thereby enhancing user experience. The hierarchical structure plays a crucial role in managing diverse queries from thousands of students about the university system. The solution introduces a modular hierarchical knowledge base (KB) with simpler complexities, addressing the intricacies of efficiently managing large volumes of queries. PixieGPT is designed in a modular way so that the solution is also adaptable for the implementation of other universities worldwide based on the requirements of a particular administrative system. The modular nature facilitates easy adaptation with minor changes based on specific university requirements, ensuring a seamless integration process. This paper delves into the intricacies of PixieGPT's design, emphasizing its pivotal role in mitigating workload challenges for university stakeholders in Bangladesh. The incorporation of BERT for Natural Language Understanding(NLU) and GPT models for Natural Language Generation(NLG) enhances PixieGPT's capabilities, contributing to the scalability and efficiency of the system. The presented use case underscores the practical benefits of PixieGPT, positioning it as a promising solution for universities globally with similar operational frameworks.

Keywords: NLP; BeRT; PixieGPT

1. Introduction

In the bustling educational landscape of Bangladesh, where universities stand as beacons of knowledge, a mounting challenge looms large — efficiently addressing the myriad queries from an ever-expanding student body. Against the backdrop of Bangladesh's densely populated environment, universities face a formidable task in navigating through the increasing influx of students. As the demand for education rises, so does the complexity of handling diverse queries from this expansive student demographic. This surge in inquiries not only strains the resources of educational institutions but also prolongs the time required for students and staff to obtain vital information. Offline assistance, the conventional method for addressing student queries, is proving to be increasingly time-consuming and, consequently, amplifying the workload for university stakeholders. The sheer volume and variety of questions emanating from students about courses, university processes, and related matters pose a significant challenge. This inefficiency hampers the overall effectiveness of educational institutions and underscores the need for innovative solutions to streamline operations and enhance user experience. The intricate nature of addressing these challenges extends beyond the sheer number of queries; it encompasses the need for a dynamic and efficient system capable of meeting the diverse informational needs of a large and growing student population. As a result, there arises a pressing demand for a transformative solution that not only mitigates workload challenges but also enhances the overall responsiveness of university systems. To address these pressing challenges head-on, we introduce PixieGPT – a specialized Generative Pre-Trained Transformer meticulously crafted for the unique context of Bangladeshi universities. PixieGPT stands as a beacon of innovation, offering a tailored approach to handling common university-related queries with unparalleled efficiency. At the core of PixieGPT's mission lies the objective to significantly mitigate workload challenges and elevate the user experience within the educational sphere. This introduction marks the unveiling of a transformative solution poised to redefine the landscape of university interactions in Bangladesh and beyond. A key innovation within PixieGPT lies in its hierarchical structure, strategically designed to manage diverse queries from thousands of students about the university system. Complementing this, the solution introduces a modular hierarchical knowledge base (KB) with simpler complexities. This forward-looking design addresses the intricacies of efficiently managing large volumes of queries, promising a paradigm shift in the way universities handle information flow. Beyond its immediate application in Bangladesh, PixieGPT unfolds as a solution designed for universal impact. Its modular design facilitates seamless integration into universities worldwide, demonstrating adaptability with minor adjustments based on specific institutional requirements. PixieGPT emerges not only as a localized remedy but as a scalable and versatile tool for similar operational frameworks on a global scale. The technical prowess of PixieGPT is further accentuated by its incorporation of cutting-edge language models. BERT, specializing in natural language understanding (NLU) and GPT models and excelling in natural language generation (NLG), synergize within the framework of PixieGPT. This powerful combination enhances the system's scalability and efficiency, positioning PixieGPT at the forefront of transformative solutions for university operations. This paper embarks on a detailed exploration of PixieGPT's design intricacies, emphasizing its pivotal role in mitigating workload challenges for university stakeholders in Bangladesh. As we delve into the nuances of PixieGPT, the incorporation of BERT for NLU and GPT models for NLG underscores its significance in the broader context of advancing conversational agents in academic settings. The ensuing use case serves as a tangible demonstration of PixieGPT's practical benefits, solidifying its position as a promising and transformative solution for universities globally operating with similar frameworks. An AI chatbot for an educational institution can serve many purposes, such as enhancing conversation, supporting students, and streamlining administrative strategies. PixieGPT, designed for Bangladeshi universities, aims to streamline operations, saving valuable time for students, faculty, staff, DAO, and various stakeholders. It offers benefits such as quick access to course details, schedules, exam information, and administrative support, fostering efficiency and convenience for all users. Below, we have provided some important use cases:

- PixieGPT serves as a versatile AI chatbot, enhancing conversation and supporting students, faculty, staff, DAO, and various stakeholders. It facilitates efficient communication and engagement across the educational institution.
- PixieGPT streamlines operations by providing quick access to crucial information such as course details, schedules, exam information, and administrative support. This results in time savings for students and stakeholders, fostering efficiency and convenience.
- PixieGPT caters to the diverse needs of users, including students, faculty, staff, guardians, admission candidates, and even outsiders visiting the campus. It provides assistance ranging from academic inquiries to facility-related information, showcasing its adaptability and usefulness across various scenarios.
- The PixieGPT system can also be adapted to any other hierarchical organization (eg. banking systems, with minor changes.

Section 2 presents the related work of the PixieGPT. Section 3 illustrates the insight of the methodological mechanism of PixieGPT. Next, in the section 4 we will analyze the expected results before addressing the limitations of our work in section 5. In addition, we present the potential future work on this topic. Finally, we conclude our paper with a summary of key findings and contributions.

2. Related Work

In the realm of educational technology, the quest for innovative solutions to address the escalating challenges faced by universities is a dynamic field of inquiry. Several models and technologies have emerged globally, aiming to streamline operations and enhance user experience. Understanding the landscape of related work becomes imperative in appreciating the uniqueness and potential impact of PixieGPT.

In a study conducted by Lalwani et al.s (2018) [8] where to developed a chatbot for inquiry purposes on college websites. The primary objective was to improve the user experience of the college website and additionally help the students to effectively their work and also time-saving approach. PixieGPT not only excels in versatility for all university stakeholders but also harnesses the power of advanced language models, BERT and GPT. While Lalwani's inquiry chatbot relies on conventional techniques only for students, integration of the cutting-edge models of the PixieGPT ensures a more sophisticated understanding and generation of natural language, enhancing its effectiveness in addressing the diverse needs of the university community.

In a study by XINGGUANG et al. (2022), a chatbot was designed for manufacturing, specifically for equipment like boilers, pressure vessels, elevators, and more.[16]Using Pressure Vessel Manufacturing (PVM), Quora Question Pair (QQP), LCQMC, and CCKS dataset for implantation. In manufacturing chatbot uses labeled corpus that is transformed into a word vector through the BeRT pretrained language model. Then, the word vector is input into the BiLSTM module for further processing. In contrast, the domain of PixieGPT is different from this chatbot. It provides answers to various questions in university. It is designed in such a way that the answer can be provided based on the roles of every user. For implementation, it uses similar types of models on manufacturing chatbot BeRT and also uses the GPT model for better performance.

In another study conducted by Rana et al. (2019), EagleBot, a chatbot developed for addressing university-related FAQs, demonstrates effectively domain-specific functionality using SQuAD1.1 dataset.[13] This chatbot exclusively provides the answers that are connected with university related topics using BeRT based model. Conversely, PixieGPT works within a similar domain, specializing in responding to issues related to universities. PixieGPT, in contrast to EagleBot, extends its capabilities to serve other audiences both inside and outside, understanding their roles, such as students, teachers, regular users, etc.

In a study carried out by Yogi et al. (2019), a chatbot was developed with the aim of providing precise and human-like responses specifically to university admission procedures.[5]This research only used admission-related processes using the sequence-to-Sequence model with and without an attention mechanism. In contrast, PixieGpt tries to implement more inclusive and versatile since it aims to expand its deployment to include a wide range of university-related procedures across multiple categories.

Yuhao et al.'s (2021) study focuses on a domain-specific knowledge base chatbot created using university website data, employing a Feature-based approach with pre-trained BeRT model fine-tuning.[6] Their BIRD-QA framework performs better with ALBeRT-base and FITT chunking strategy for domain-specific QA. In contrast, Zhong et al.'s (2020) research introduce a chatbot for answering questions about building regulations.[18] Using IR, BeRT, and VSM models, their chatbot achieves high effectiveness, highlighting the importance of specialized models and effective dataset organization in regulatory contexts. By comparing these two papers, PixieGPT outperforms them by employing a dual-model strategy, incorporating both BeRT and GPT models. This ensures a more robust and versatile chatbot capable of handling a broader range of queries. PixieGPT excels in adaptability, catering to diverse university-related inquiries while integrating BeRT for precise question understanding and GPT for detailed and natural responses. Notably, PixieGPT surpasses the limitations of Zhong et al.'s (2020) paper, which focuses solely on construction building regulations. In contrast, PixieGPT caters to a wide range of users, including students, faculty, staff, guardians, admission candidates, outsiders, proctors, and registers, making it a versatile solution for various university-related inquiries. In summary, PixieGPT's dual-model architecture, adaptability, and advanced model integration make it superior and more versatile compared to the specific-use cases addressed by Yuhao et al. and Zhong et al. The paper "Evaluation of an Arabic Chatbot" by Alruqi and Alzahrani focuses on employing language transformers in Arabic chatbots, particularly AraBERT and AraElectra, for question-answering.[19] While their study is effective, it has limitations, such as not discussing scalability and focusing on a specific method. In contrast, PixieGPT offers a more advanced and versatile solution by using both BeRT and GPT models, catering to a broad range of queries. PixieGPT serves various users and provides detailed responses in English, overcoming the limitation of the reviewed paper, which is limited to Arabic speakers. This difference makes PixieGPT accessible to a wider audience, addressing a key limitation of the "Evaluation of an Arabic Chatbot" paper. PixieGPT performs better than the university-specific chatbots examined in "A Comparison of ChatGPT-3.5, ChatGPT-4, and Google Bard" by Vagelis Plevris et al. Unlike the authors' focus on math and logic problems,[21] PixieGPT, using both BeRT and GPT models, serves as a flexible solution for various questions. While the studied papers are good at specific tasks, on the other hand, PixieGPT, assisting students, faculty, staff, and others, provides overall help. Unlike the checked papers, PixieGPT's use of two models lets it give flexible solutions, making it better at handling a wide range of user needs.

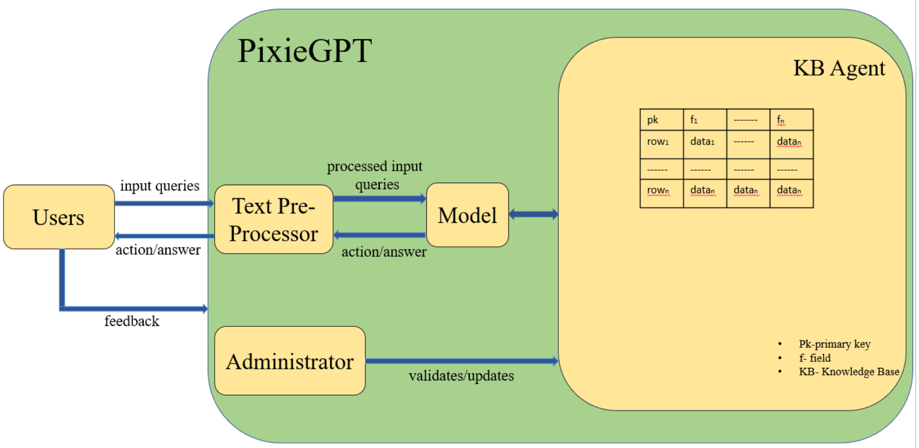
The paper "A Transformer-Based Educational Virtual Assistant Using Diacriticized Latin Script" by Loc Huu Nguy introduces a method for constructing a virtual assistant using Transformer-based models.[22] They focus on addressing challenges in the Vietnamese language, specifically handling misspelled words and words without diacritics. The authors present two Transformer-based models for diacritic restoration and mistyped word correction, tailored for the university environment. In comparison, PixieGPT stands out with its dual-model strategy employing BERT and GPT models. While the evaluated paper targets specific language challenges, PixieGPT's broader application and advanced models make it a superior solution for varied user needs in the university context. PixieGPT's dual-model architecture and adaptability position it as a more effective and versatile choice compared to the paper's specific focus on the Vietnamese language and university domain challenges.

A study shows a machine designed for human-like responses in real Question Answer scenarios for Chinese MRC(Machine Reading Comprehension) by Chen et al.s (2023). The primary objective of this work is to empower correctly answer queries based on the given passages or paragraphs, which is an active field in NLP. Uses two existing modes like BART and T5 with predetermined and inflexible answers on the above benchmarks. In contrast, the domain of PixieGPT is different from this machine. It provides answers to various questions in university. It is designed in such a way that the answer can provide a role-based approach. In a study shows that build a system which emulate human like intelligence by Gupta et al.s (2023). This system uses machine learning to extract knowledge from from pictures, text, etc. This extracted knowledge is presented as logically predicated.It also uses CASP to check the consistency of this extracted knowledge and reason over it in a manner very similar to how a human would do it. Additionally, it uses LLM and GPT3. In contrast, PixieGpt is able to answer users' queries, specializing in responding to related universities. Uses same kind of model GPT and BeRT for better performance.

3. Methodology

The methodology section describes the Data Collection, Hierarchical Knowledgebase Design, Overview of PixieGPT System Architecture, ML model training process, Evaluation of the model and Integration With University Systems.

3.1. *Overview of System Architecture*



pk	f_1	-------	f_n
row_1	$data_1$	------	$data_n$
------	------	------	------
row_n	$data_n$	$data_n$	$data_n$

Figure 1. Primary architecture of PixieGPT

This section describes the core design of PixieGPT architecture. The architecture has several components.
The components of the PixieGPT are as follows:

3.1.1. Knowledge Base (KB):

Knowledge base refers to a centralized repository where all information, data, and knowledge are stored for the use of the chatbot. Here all kinds of data will be stored, these data can be structured, unstructured, or semi-structured. By retrieving these data, the AI program will make decisions and answers. The knowledgebase design and implementation is described more in the next section.

3.1.2. Actors:

There are two primary actors who will be involved with PixieGPT system. Though the Administrator is a user but he is the specialized one and all the others are normal Users.

- **Administrator:** Admins are individuals or teams who are responsible for managing and overseeing the PixieGPT. This group might comprise IT professionals, administrative staff, and other relevant stakeholders within the educational institution. They will look after the technical oversight, content management, and policies.
- **Users:** Users refer to students, faculty, staff, or any other individuals interacting with the PixieGPT seeking information related to the educational institution's services and resources.

  - **Students:**

    Students can ask PixieGPT about their course details, class schedules, and information on upcoming exams. If students have questions about enrollment, fees, or any administrative processes, they can seek assistance from PixieGPT as well. It can inform students about campus facilities, events, and extracurricular activities and many more.
  - **Faculty:**

    Faculty members can inquire about class rosters, schedules, and any changes in the academic calendar. PixieGPT can assist faculty members with administrative tasks, such as submitting grades, accessing resources, and providing information on institutional policies. They can use the PixieGPT to coordinate with other departments or colleagues, facilitating smoother communication.
  - **Staff:**

    Staff members can seek information on internal processes, policies, and guidelines through PixieGPT. It can provide information related to HR policies, leave requests, and other employee-related queries. Staff can use it to report facility issues, schedule maintenance, and access information about campus facilities.

- **Guardians:**

  Guardians can inquire about their child's academic progress, grades, and attendance through PixieGPT. It can provide details about school events, parent-teacher meetings, and other relevant activities.
- **Admission Candidates:**

  Admission candidates can seek information on the application process, required documents, and key deadlines through PixieGPT. It can provide details about various academic programs, entry requirements, and potential career paths.
- **Outsiders:**

  Individuals visiting the campus can use PixieGPT to obtain information about directions, parking, and visitor guidelines. Outsiders looking for general information about the institution, such as its history, mission, or notable achievements, can use PixieGPT as a quick resource.
- **Others:** Users like proctors, registers, and librarians can use PixieGPT. Proctors access exam details, schedules, and instructions. Registrars manage student records, enrollment, and documents. Librarians inquire about resource availability, hours, and events.
  It will be confirmed that every person can get help from the system.

3.1.3. Text Preprocessor:

NLP (Natural Language Processing) in the conversational chatbot job, is the text processor which is like a language expert. It helps PixieGPT understand what people are saying, figure out what they want, and respond in a way that people understand. It's like having a conversation in a way that feels natural to the people, making it easier for users to get the help they need.

- **Query Processing:**To turn user questions into a format suitable for database searches, first preprocess and tokenize the query, ensuring lowercase consistency and removing unnecessary words. Identify key terms, handle variations, and recognize entities.
- **Lookup KB:**To search in a knowledge base for a specific query, the system processes the user's question, breaks it down into key terms, and matches those terms with information stored in the database. Using NLP techniques like Name entity Recognition and sentiment analysis, it identifies the most relevant entries and retrieves them as potential answers to the user's inquiry.
- **output processing:** The output of every database query is then processed using NLG to generate human like answers.

3.2. Knowledge Base(KB) Design and Implementation

3.2.1. Database Creation:

The database architecture for the university system comprises multiple collections meticulously organized to store diverse sets of information systematically. These collections include essential categories such as admission, users, student, teacher, staff, scholarship, department, courses, and more. Each collection is designed to serve as a repository for specific types of data, ensuring efficient data management and retrieval.

During the database design phase, careful consideration was given to the storage and organization of data to facilitate seamless access to information. Information was sourced from various reliable sources including the university website and administrative offices to ensure comprehensive coverage of relevant data points. This comprehensive database serves as a valuable resource not only for system operations but also for training machine learning models.

Within the faculty collection, pertinent details such as room numbers, email addresses, office phone numbers, and other faculty-related information are readily accessible. Additionally, the database encompasses all vital university-related information crucial for students and other users. This includes procedures for course enrollment, fee payments, scholarship applications, as well as penalties and regulations.

One of the key features of the database design is the establishment of relationships between collections. These relationships enable the system to navigate seamlessly through the data, facilitating efficient retrieval of information tailored to the user's specific queries. This relational structure enhances the system's ability to provide accurate and relevant responses to user inquiries, contributing to an enhanced user experience.

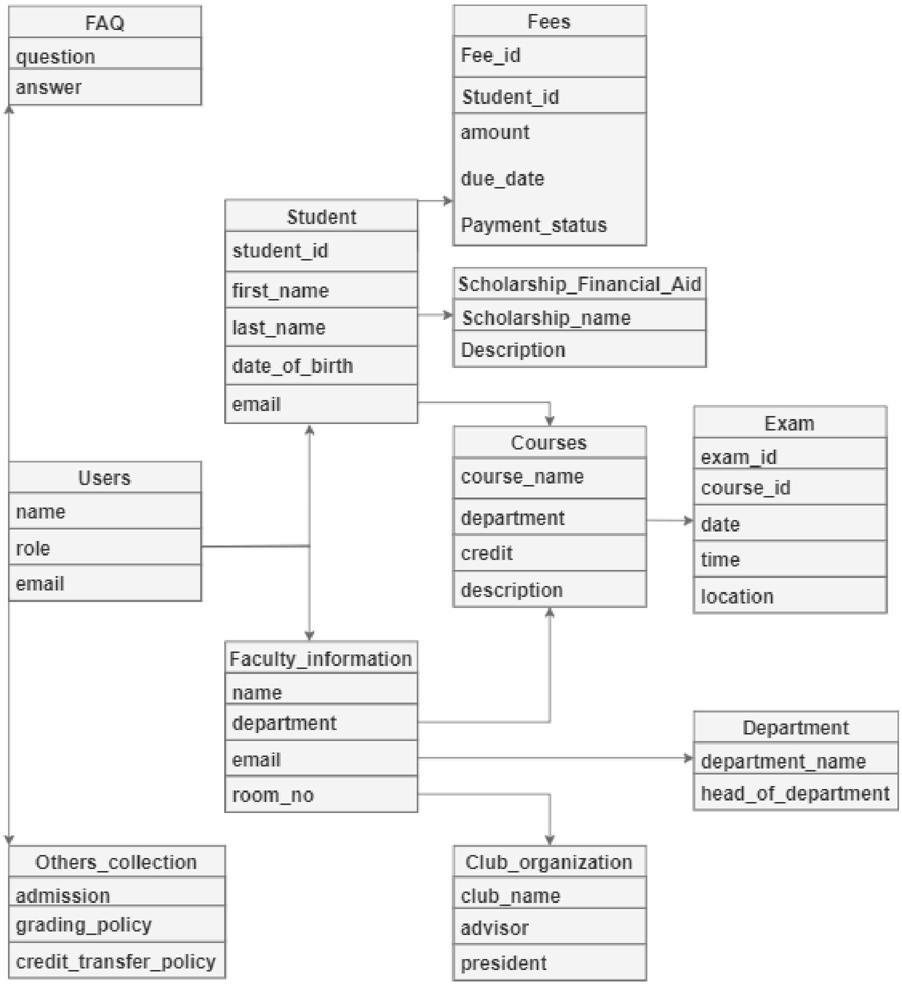


Figure 2. Overview of Database Design

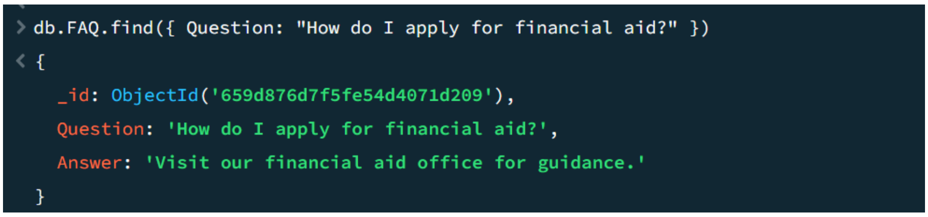
3.2.2. Hierarchical Knowledgebase:

A hierarchical knowledge base is like a well-organized tree of information. It categorizes data into levels or tiers, making it easy for a PixieGPT to find and provide specific information. In simpler terms, it helps the PixieGPT quickly and efficiently access to provide specific details based on the user's queries. Combining a hierarchical knowledge base with categorized queries in PixieGPT improves efficiency and user experience. The hierarchy organizes information on administrative, financial, academic, and about the institution enabling PixieGPT to offer targeted and specialized responses within each category. The approach of using a hierarchical knowledge base with categorized queries will potentially benefits the time complexity of PixieGPT system. The hierarchical structure allows for a more organized and efficient navigation through information. By categorizing queries, PixieGPT can quickly narrow down the scope of relevant information, reducing the search space and enabling faster retrieval of responses.

Moreover,in the database every collection are hierarchical.Every collection works in hierarchically.Suppose,if the users wants to know about a courses, then the system goes courses collection and find the course related info such as course code, name, department,objective, credit ,duration etc and provide the desire answer to the users.

3.2.3. Database Result:

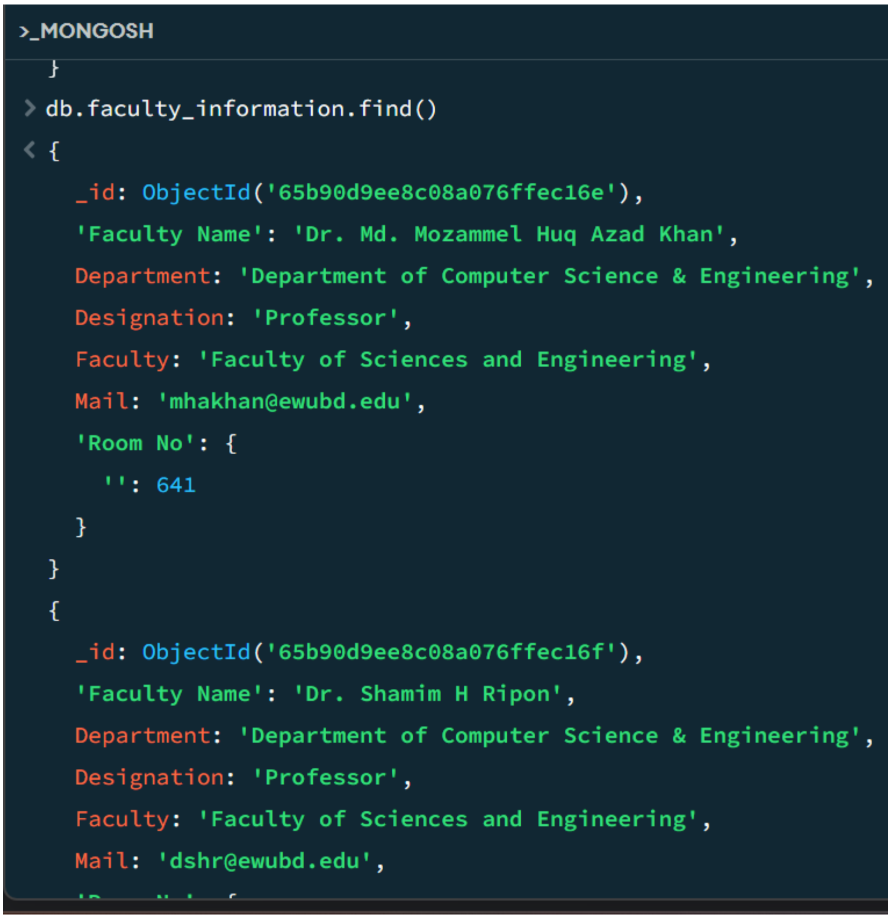
We added MongoDB to our project to help manage data better and make things scalable. MongoDB is flexible and can handle changing data needs. Now, let's take a look at what we achieved with it. Here is the output of MongoDB, which we have implemented.



```
> db.FAQ.find({ Question: "How do I apply for financial aid?" })
< {
    _id: ObjectId('659d876d7f5fe54d4071d209'),
    Question: 'How do I apply for financial aid?',
    Answer: 'Visit our financial aid office for guidance.'
  }
```

Figure 3. Database query result

The MongoDB query "db.Faculty_information.Find()" is a command that retrieves and shows all of the files within the "faculty_information" collection in the MongoDB database. It essentially fetches all of the data to be had for all faculties stored in that specific collection, supplying a complete view of the faculty information.



```
>_MONGOSH
  }
> db.faculty_information.find()
< {
    _id: ObjectId('65b90d9ee8c08a076ffec16e'),
    'Faculty Name': 'Dr. Md. Mozammel Huq Azad Khan',
    Department: 'Department of Computer Science & Engineering',
    Designation: 'Professor',
    Faculty: 'Faculty of Sciences and Engineering',
    Mail: 'mhakhan@ewubd.edu',
    'Room No': {
      '': 641
    }
  }
  {
    _id: ObjectId('65b90d9ee8c08a076ffec16f'),
    'Faculty Name': 'Dr. Shamim H Ripon',
    Department: 'Department of Computer Science & Engineering',
    Designation: 'Professor',
    Faculty: 'Faculty of Sciences and Engineering',
    Mail: 'dshr@ewubd.edu',
    'Room No': {
```

Figure 4. Database query result

The MongoDB query db.FAQ.Locate(Question: "How do I observe for financial aid?") is a command designed to retrieve records specifically related to the query "How do I apply for financial aid?" from the "FAQ" collection. This question facilitates find and display the file in the series that corresponds to the required question, supplying a focused response to that unique inquiry.

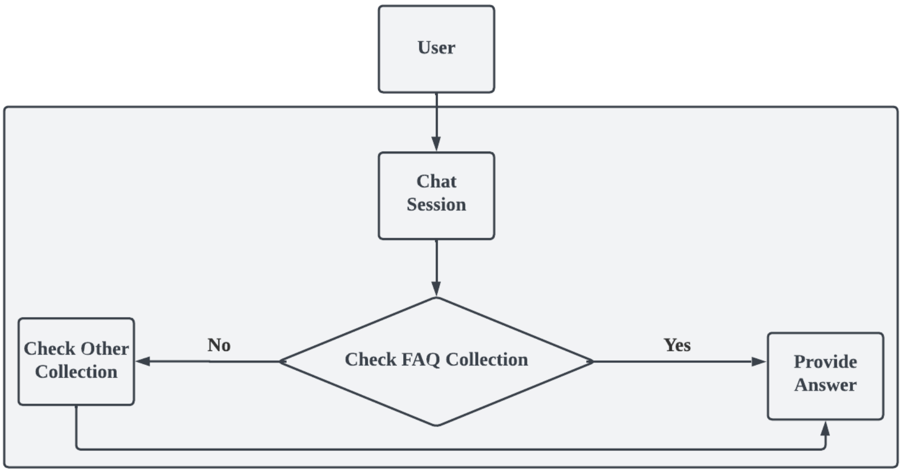


Figure 5. System hiearchy of PixieGPT

3.3. System Hierarchy

First of all, A user ask a question in the System. System check in the database for compare between the user's question and the database's stored question and answer. If the users query and database question are same then system return the desired output. If not found then it goes to others table in the database for check these types of answers.

Algorithm 1: BeRT-based Knowledge Base Query

```
Input   :User input (user_input)
Output:Generated answer (answer)
KnowledgeBaseQuery(user_input)
Step 1: Load BeRT model and knowledge base
  load_bert_model()
  load_knowledge_base()
Step 2: Preprocess user query
  query = preprocess_query(user_input)
Step 3: Retrieve relevant knowledge base entries
  relevant_entries = retrieve_entries_from_kb(query)
Step 4: If relevant entries found
  combined_input = combine_query_and_entries(query, relevant_entries)
  bert_output = bert_model(combined_input)
  answer = generate_answer_from_bert_output(bert_output)
Step 5: Else
  answer = generate_no_answer_found_response()
Step 6: Return the generated answer
return answer
```

3.4. Training Process

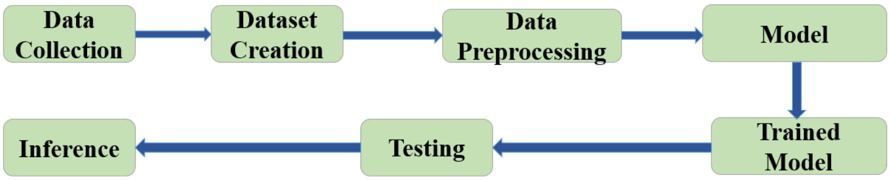


Figure 6. Methodology of PixieGPT

3.4.1. **Data Preparation**

The structured,unstructured and FAQ data collection process was different. The processes are as follows:

- Structured data collection: Structured data has been collected from the KB of the PixieGPT.
- FAQ collection: Frequently asked questions and answers have been collected from the DAO (Department Administrative Officer), students, staff etc.
- Unstructured data collection: Unstructured or textual data have been collected from the university website. Our dataset will have a question and answering pair on structured, unstructured and FAQ data.

Afterwards, different kinds of data have been organized into passage, questions, and answers to create the dataset, Where the **'passage'** field contains the information of a topic. The **'question'** and **'answer'** field of the dataset contains the Q & A generated from the **'passage'**.The process of creating **'passage'**,**'question'** and **'answer'** has been done through the template-based and rule-based method.

3.4.2. **Experimental Setup**

Our experimental setup utilizes Google Colab's T4 GPU engine to power the development and training of a question answering system. Leveraging the Colab environment's free access to T4 GPUs, we employed popular deep learning frameworks such as TensorFlow or PyTorch to construct and train our question answering model. The Colab interface allows seamless integration of code development, data processing, and model training within a Jupyter notebook environment, optimizing the utilization of T4's computational capabilities. By harnessing the parallel processing power of the T4 GPU, we aim to expedite the training process, enhance model performance, and achieve efficient inference for our question answering system, thereby showcasing the benefits of utilizing cloud-based GPU resources for machine learning tasks.

3.4.3. **Preprocessing**

Data preprocessing is an important step in preparing raw data for machine learning models. In this context, it involves tasks such as cleaning and organizing text data for sentiment analysis, text cleaning, tokenization, and addressing class imbalance.

BERT and GPT both are pre-trained models. These models can be fine-tuned to do specific tasks in a domain. Before fine-tuning the model, the datasets we have created must be pre-processed. Preprocessing like. Then the model will perform feature extraction and answer generation tasks. Preprocessing includes,

- Feature Selection: We will carefully pick the most relevant features from our data.
- Text Cleaning: Will define functions to clean text data, removing punctuations, new line characters.
- Stopword Removal: Remove common English stop words to focus on meaningful content.

- Hashtag Cleaning: Clean hashtags, removing the ’#’ symbol at the end of sentences and retaining it in the middle.
- Filtering Special Characters: Filter specific characters like ’&’ and ’$’ from words in the text.
- Multiple Cleaning Steps: Iterate through each text, applying the defined cleaning functions in sequence to create a cleaner version of the text.
- Add Cleaned Text Column: Create a new column (’text_clean’) to store the cleaned text in the dataset.
- Tokenization : Tokenization is the process of breaking down text by words or by sentences. Our dataset contains unstructured data which has a wide range of topics courses, academic calendar, scholarship, and other common inquiries. These data will be tokenized to understand by the model.
- Data Augmentation: Data augmentation involves creating diverse variations of the existing text to enhance the training dataset to increase the accuracy of the model inference. Text data augmentation like synonym replacement, random word insertion, random swap, generating alternative phrasing and use of contractions and expansions might be useful for our project.
- N-gram Analysis: Uncovering Patterns and Phrases: We delve into N-gram Analysis, a method to uncover meaningful patterns and phrases within the text data.
- Deep Dive into Tokenization: We employ the BERT tokenizer to further clean and analyze the training data, validation data and also testing data. The tokenizer is configured with the ’bart-base-uncased’ model.
- Shuffling Data: In the train-validation-test split process, we utilized the StratifiedShuffleSplit method to ensure a representative distribution of all classes in each subset. For the training set, we performed a stratified shuffle split with 80% for training and 20% for validation, maintaining the class proportions. The same split configuration was applied to the validation set.

These preprocessing steps collectively prepare the dataset for model training, ensuring that the text data is cleaned, tokenized, and appropriately formatted for subsequent modeling steps.

3.4.4. **Deep Learning Model Training:**

The forthcoming training phase of our project will involve the utilization of cutting-edge deep learning techniques to fine-tune pre-trained transformer models for specific tasks within our university’s chatbot system. Leveraging Python’s NLTK framework for natural language processing tasks and TensorFlow for deep learning, we will embark on a journey to refine our models to achieve optimal performance.

Our primary focus will rest on two key transformer models: BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer). BERT, renowned for its versatility and effectiveness in natural language understanding tasks, will serve as the cornerstone for tasks such as text classification, named entity recognition, and question-answering. By fine-tuning BERT on our university-specific datasets, we aim to equip our PixieGPT with the ability to comprehend and respond to user queries with exceptional accuracy and contextual understanding. In tandem with BERT, we will employ the GPT model for natural language generation (NLG) tasks. GPT excels in generating human-like responses and maintaining coherent conversation flows, making it an invaluable component of our PixieGPT system. Through extensive training on diverse conversational datasets, GPT learns to produce responses that are not only contextually relevant but also exhibit fluency and coherence akin to human communication.

The training process will involve iterative refinement, where the models will be exposed to annotated datasets and fine-tuned through numerous epochs to optimize performance metrics such as accuracy, precision, and recall. We will employ sophisticated optimization techniques and loss functions to guide the training process and ensure convergence towards the desired objectives.

3.5. *Evaluation Metrics*

When evaluating a chatbot designed to answer questions for students, the choice of evaluation metrics is critical. These metrics like precision, recall, F1-score, MRR and BLeU help us understand how well the PixieGPT is performing. They rate his ability to offer the right solutions, how regularly he avoids mistakes, how quickly he responds, and the first-class solutions he provides. These factors are crucial to ensure that the chatbot is useful and effective. These metrics help us understand various factors of a PixieGPT's performance, including the accuracy, completeness, speed of response, and pleasantness of its responses. These matters are vital to ensure that the PixieGPT is undoubtedly useful for students.

- **Precision** This metric assesses how correct the PixieGPT's solutions are. It measures the proportion of correct solutions out of all of the solutions the PixieGPT affords. A high precision score suggests that the PixieGPT is reliable in giving correct answers and avoids mistakes.
  Mathematical Representation:
  $$Precision = \frac{TP}{TP + FP}$$
  TP (True Positives): The number of correct answers the chatbot provided.
  FP (False Positives): The number of incorrect answers provided by the chatbot when it should not have.
- **Recall:** Recall evaluates how well the PixieGPT avoids missing correct solutions. It measures the proportion of questions that the PixieGPT effectively answers out of all the questions it had to answer. An overestimation of recall indicates that the PixieGPT rarely misses the questions it should deal with.
  Mathematical Representation:
  $$Recall = \frac{\text{TP}}{\text{FN} + \text{TP}}$$
  TP (True Positives): The number of correct answers the PixieGPT provided.
  FN (False Negatives): The number of questions for which the PixieGPT should have provided an answer, but it failed to do so.
- **F1-Score:** The F1 Score shifts the stability between accuracy and consideration. This is particularly useful when there may be a need to consider the trade-off between avoiding wrong answers and not missing the right ones now. A high F1 Score means that the PixieGPT effectively balances accuracy and completeness.
  Mathematical Representation:
  $$F_1 = \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
  F1-score combines the precision and recall scores into a single value.
- **Mean Reciprocal Rank (MRR)** MRR specializes in the speed of providing correct answers. It calculates how often the PixieGPT places the ideal solution at the top of its response list.It gives perfect answers quickly, which is especially important for engaging people..
  Mathematical Representation:
  $$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{Rank\_i}$$
  N: The total number of questions.
  Rank_i: The position (rank) of the first correct answer for the i-th question.
- **BLeU (Bilingual evaluation Understudy)**When the PixieGPT generates text responses, BLeU measures the best of these solutions by evaluating them as a reference solution. The generated responses appear to match the reference responses by examining the overlap of the n-grams.

A higher BLeU rating indicates higher quality answers, this is essential. BLeU is calculated by evaluating the n-grams in the generated solution with those in the reference responses. The final rating is a mixture of accuracy, where a better score indicates a higher overlap.

3.6. *Integration With University Systems*

The development of the Final PixieGPT System Evaluation will comprise several key stages following the training of the BERT model. To seamlessly integrate the trained BERT model, Django in Python will be employed to deploy the model as a web service, facilitating efficient interactions with the system. The front-end design, crafted using HTML, Bootstrap CSS, and JavaScript, will provide users with an intuitive interface for posing questions and interacting with PixieGPT. Meanwhile, the back-end operations managed through PHP within the Laravel framework, will ensure effective communication between the user interface and the BERT model, employing API endpoints to send user queries and retrieve responses. MongoDB will serve as the database solution, enabling the storage of user queries and model responses, thereby facilitating the analysis of user interactions for future enhancements. The deployment will be orchestrated through hosting services like Heroku, AWS, or Google Cloud Platform, with meticulous attention given to security measures, including input validation and user authentication, considering potential public deployment.

Extensive testing and evaluation will be conducted to verify the seamless integration of the front-end, back-end, and BERT model, evaluating PixieGPT's responses to diverse inquiries. Throughout the development process, thorough documentation and reference to pertinent tutorials and best practices will be pivotal in ensuring the system's robustness, usability, and security. Thorough testing and evaluation will be pivotal in validating the seamless integration and performance of the PixieGPT system. A comprehensive testing approach will be employed, encompassing various methodologies to ensure the system's robustness, accuracy, and usability across different dimensions. Unit testing, employed to verify individual system components in isolation, will ensure the correctness of functionalities within the front-end, back-end, and model integration layers. Meanwhile, integration testing will facilitate the examination of interactions between modules, guaranteeing their cohesive operation. End-to-end testing, conducted to assess the entire system workflow from user input to model response, will provide insights into the overall functionality and user experience. Performance testing will offer a deeper understanding of system behavior under different loads and conditions, affirming its stability and scalability. Security testing will play a crucial role in identifying and addressing potential vulnerabilities, ensuring the system's resilience against security threats. Moreover, usability testing involving user interactions will aid in refining the interface and enhancing user experience. Data quality evaluation and benchmarking exercises will further contribute to evaluating PixieGPT's effectiveness, ensuring its alignment with expectations and assessing its superiority against comparable models or systems.

The meticulous application of these testing methodologies and evaluation strategies will not only affirm the system's readiness for deployment but also contribute significantly to its robustness, accuracy, security, and user satisfaction.

4. Expected Result

In integrating PixieGPT into an educational institution, we anticipate multifaceted and promising effects. Primarily, the model is expected to exhibit an excessive degree of expertise in generating contextually appropriate responses tailored to the specific inquiries and needs of the university's students, parents, and staff groups. In addition, PixieGPT is expected to excel in information dissemination and utilize its huge record storage to always offer correct and up-to-date answers to users. In addition, PixieGPT scalability is expected to play a key role in handling various queries ranging from admission information to faculty information, schedules, and institutional rules. This scalability eases the workload of the administrative group of workers, and it also guarantees that users get stable and reliable information for various queries, which contributes to continuous and effective use. Essentially, the integration of PixieGPT into an academic institution is expected to lead to advanced user experience through well-timed and correct responses, and scalability to satisfy a large selection of queries. Together, these results contribute to the effective support and operational performance of educational institutions and position the PixieGPT as a valuable tool for facilitating communication and disseminating information in the academic environment.

5. Limitations

The limitations of this paper are as follows:

- Collecting sensitive information of the different stakeholders due to ethical and privacy concerns.
- Handling diversified queries of the different languages: Presently, PixieGPT considers English as its communication medium.

6. Future Work

This paper introduces a smart institutional helpline for students and other user's queries efficiently. In the future development of this educational helpline, we envision several areas to enhance its capabilities and effectiveness in terms of robustness and scalability of the hierarchical KB handling versatile queries of the different stakeholders. The potential improvements of the PixieGPT are the following:

- **Multimodal capabilities:** The system takes only the text as input. In future work, there is a space to enhance the input mechanism by integrating multimedia support such as image, audio, and video recognition.
- **Multi language engagement** PixieGPT can be extended to handle different languages e.g., Bangla, Finish, French, Spanish etc.
- **Implement Adaptive Learning:** By implementing adaptive learning mechanisms, PixieGPT responses are based on what users like and how they used it before. This will help to chat with users better.

7. Conclusions

PixieGPT emerges as a tailored solution aimed at revolutionizing the educational landscape in Bangladesh by addressing the intricate challenges faced by universities in managing the inquiries of the stakeholders. To do so, PixieGPT integrates the advanced NLP techniques such as BERT and GPT models within a hierarchical knowledge base framework, PixieGPT introduces a robust platform for the efficient handling of the diverse queries of the stakeholders while experiencing the overall user expectations. The meticulous development process of the PixieGPT encompasses the database design, efficient KB lookup mechanism in O(nlgn), model training, and rigorous testing, underscores our commitment to delivering a reliable and scalable solution to the evolving needs of the university stakeholders of Bangladesh. In the future work of the PixieGPT, we will focus on the continuous improvement and its expansion to the broader community of the global educational system. Besides, we envision refining the system through the model training and performance optimization of the PixieGPT. Moreover, PixieGPT is designed and implemented in a way that can easily be adapted to other hierarchical organizations with minimal change. In addition, PixieGPT is dedicated to leveraging user feedback and data-driven insights to drive iterative enhancements through the PixieGPT.

In summary, PixieGPT represents a significant step forward in reshaping how universities engage with their constituents, offering a powerful tool to streamline operations and enhance user satisfaction. Due to the modular and hierarchical design of PixieGPT, it is poised to usher in a new era of efficient innovation in the educational domain, mitigating a significant amount of time for prospective students, faculties, staff, and other stakeholders.

References

1. Chen, Y. and Zulkernine, F., 2021, December. BIRD-QA: a BeRT-based information retrieval approach to domain specific question answering. In 2021 Ieee International Conference On Big Data (Big Data) (pp. 3503-3510). Ieee.
2. Rana, M., 2019. eaglebot: A Chatbot Based Multi-Tier Question Answering System For Retrieving Answers From Heterogeneous Sources Using BeRT.
3. Chandra, Y.W. and Suyanto, S., 2019. Indonesian chatbot of university admission using a question answering system based on sequence-to-sequence model. Procedia Computer Science, 157, pp.367-374.
4. Xingguang, L., Zhenbo, C., Zhengyuan, S., Haoxin, Z., Hangcheng, M., Xuesong, X. and Gang, X., 2022. Building a Question Answering System for the Manufacturing Domain. Ieee Access, 10, pp.75816-75824.
5. Adamopoulou, e. and Moussiades, L., 2020. Chatbots: History, technology, and applications. Machine Learning with Applications, 2, p.100006.
6. AbuShawar, B. and Atwell, e., 2015. ALICe chatbot: Trials and outputs. Computación y Sistemas, 19(4), pp.625-632.
7. Lund, B.D. and Wang, T., 2023. Chatting about ChatGPT: how may AI and GPT impact academia and libraries?. Library Hi Tech News, 40(3), pp.26-29.
8. Lalwani, T., Bhalotia, S., Pal, A., Rathod, V. and Bisen, S., 2018. Implementation of a Chatbot System using AI and NLP. International Journal of Innovative Research in Computer Science & Technology (IJIRCST) Volume-6, Issue-3.
9. Rajaraman, V., 2023. From eLIZA to ChatGPT: History of Human-Computer Conversation. Resonance, 28(6), pp.889-905.
10. Brill, T.M., Munoz, L. and Miller, R.J., 2022. Siri, Alexa, and other digital assistants: a study of customer satisfaction with artificial intelligence applications. In The Role of Smart Technologies in Decision Making (pp. 35-70). Routledge.
11. Yadav, S.S., Kumar, P., Kumar, S. and Singh, S., 2021, December. Google Assistant Controlled Home Automation with Voice Recognition. In 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N) (pp. 1039-1044). Ieee.
12. Kepuska, V. and Bohouta, G., 2018, January. Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home). In 2018 Ieee 8th annual computing and communication workshop and conference (CCWC) (pp. 99-103). Ieee.
13. Liu, B., Wei, H., Niu, D., Chen, H. and He, Y., 2020, April. Asking questions the human way: Scalable question-answer generation from text corpus. In Proceedings of The Web Conference 2020 (pp. 2032-2043).
14. Zhong, B., He, W., Huang, Z., Love, P.e., Tang, J. and Luo, H., 2020. A building regulation question answering system: A deep learning methodology. Advanced engineering Informatics, 46, p.101195.
15. Sadhuram, M.V. and Soni, A., 2020, July. Natural language processing based new approach to design factoid question answering system. In 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 276-281). Ieee.
16. Ahmad, N.A., Hamid, M.H.C., Zainal, A. and Baharum, Z., 2020. UNISeL Bot: Designing Simple Chatbot System for University FAQs. International Journal of Innovative Technology and exploring engineering, 9(2), pp.4689-4693.
17. Nguyen, T.T., Le, A.D., Hoang, H.T. and Nguyen, T., 2021. NeU-chatbot: Chatbot for admission of National economics University. Computers and education: Artificial Intelligence, 2, p.100036.
18. Vamsi, G.K., Rasool, A. and Hajela, G., 2020, July. Chatbot: A deep neural network based human to machine conversation model. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-7). Ieee.
19. Alruqi, T.N. and Alzahrani, S.M., 2023. Evaluation of an Arabic chatbot based on extractive question-answering transfer learning and language transformers. AI, 4(3), pp.667-691.
20. Plevris, V., Papazafeiropoulos, G. and Jiménez Rios, A., 2023. Dataset of the study:"Chatbots put to the test in math and logic problems: A preliminary comparison and assessment of ChatGPT-3.5, ChatGPT-4, and Google Bard". Zenodo.
21. Lam, K.N., Nguy, L.H. and Kalita, J., 2023. A Transformer-based Educational Virtual Assistant Using Diacriticized Latin Script. IEEE Access.

22. Pearce, K., Alghowinem, S. and Breazeal, C., 2023, June. Build-a-bot: teaching conversational ai using a transformer-based intent recognition and question answering architecture. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 37, No. 13, pp. 16025-16032).
23. Zeng, Y., Rajasekharan, A., Padalkar, P., Basu, K., Arias, J. and Gupta, G., 2024, January. Automated interactive domain-specific conversational agents that understand human dialogs. In International Symposium on Practical Aspects of Declarative Languages (pp. 204-222). Cham: Springer Nature Switzerland.
24. Hao, H., Sun, X.E. and Wei, J., 2023. A semantic union model for open domain Chinese knowledge base question answering. Scientific reports, 13(1), p.11903.
25. Yan, R., Li, J. and Yu, Z., 2022. Deep learning for dialogue systems: Chit-chat and beyond. Foundations and Trends® in Information Retrieval, 15(5), pp.417-589.
26. Chen, N., Li, H., Bao, Y., Wang, B. and Li, J., 2023. Natural Response Generation for Chinese Reading Comprehension. arXiv preprint arXiv:2302.08817.
27. Gupta, G., Zeng, Y., Rajasekaran, A., Padalkar, P., Kimbrell, K., Basu, K., Shakerin, F., Salazar, E. and Arias, J., 2023. Building Intelligent Systems by Combining Machine Learning and Automated Commonsense Reasoning. In Proceedings of the AAAI Symposium Series (Vol. 2, No. 1, pp. 272-276).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.