INNOVATION CONCEPT

TEAM MEMBERS

Kan Asvasena 103822142
Arsha Manoj   104063764
Younesh Mahaseth 103824902
Aayush Joshi 103519033

TABLE OF CONTENTS

Executive Summary 3

Part A 3

Project Overview 3

Client Requirements 3

Part B 4

Design Concept 4

Preliminary Design -Younesh Mahaseth 4

Methodology 4

Design constraints 6

Specifications: 6

Design 2 -Arsha Manoj 7

Methodology 7

Design constraints 9

Specifications 9

Design 3 - Aayush Joshi 10

Methodology 10

Design constraints 12

Specifications 12

Design 4 - Kan Asvasena 13

Methodology 13

Design constraints 15

Specifications 15

Justification of Designs 16

Justification Summary 17

References 17

Executive Summary

This project aims to develop a language model training and question-answering system with a focus on ethical considerations, collaboration, data preparation, and model fine-tuning. Four distinct design concepts have been proposed, each offering unique approaches and methodologies. The approach for each design varies. The designs are classified based on four different approaches. The first design is purely based on python. It prioritises collaboration and comprehensive data augmentation and tokenization processes. The second design focuses on implementing Langchain throughout the project. It places a strong emphasis on ethical considerations in prompt preparation and utilises various data sources and augmentation techniques. The third design is purely focused on utilising AWS resources. It emphasises implementing Domain adaptation techniques by using Amazon Sagemaker Studio. The final design employs advanced technologies such as AWS and Hugging Face Transformers, offering comprehensive data processing, fine-tuning strategies, and robust ethical considerations. Noticeable advantages and disadvantages are listed for each design respectively. Important factors like budget, technical proficiency and project requirements are thoroughly mentioned in the project.

Part A

Project Overview

The project implements the use of generative AI in the domain of education. This project consists of two integral phases: the initial phase involves the meticulous compilation of question banks in the realms of Environmental Management (ENV30003), Food Science (PEH20002), and The Microbial World (BIO20002). The objective here is to amass 120 distinct, non-overlapping questions, ideally encompassing multiple-choice and multi-answer formats within each subject area. Collaborative exchange among groups enriches the question pool. Phase 2 shifts the focus to evaluating the accuracy of LLama2. Each group employs a different model without any fine-tuning, simulating an average environmental study student's AI-assisted quest for answers. Experiments involve both prompted and unprompted AI responses, with teams exploring varied prompt designs and ensuring consistent results. Ethical considerations will guide the project's boundaries, and conventional classification matrices will assess AI-generated answers, offering valuable insights into AI's educational applications.

Client Requirements

The client requirements for this project include:

1) Question Bank collection: Collecting 120 questions from authentic sources and ensuring that the questions are non-overlapping.
2) Prompt Design: The teams are required to offer multiple approaches to create a prompt design.

3) Ethical Considerations: While prompting the design it is necessary to ensure that the models follow ethical boundaries.
4) Evaluation Metrics: The developed model should be tested in terms of accuracy and reliability by inculcating the use of multiple parameters.
5) Fine-tuning: It is a requirement to fine tune the model in order to get better results.
6) Meeting Deadlines: Teams are required to put effort in completing the work as per the flow of the project.
7) Collaboration: Teams must collaborate in order to ensure enhancement in the final output.

Part B

Design Concept

Being an innovation project, bringing up design ideas to resolve specific tasks is of paramount importance. As the field of LLM is expanding its possibilities day by day, the ways and tools to explore and train them are also being updated. Having a variety of ways of solving the requirement of the project, this documentation shall explain four different designs along with their steps, tools, process and evaluation.

This documentation would enable comparison between four different designs based on different frameworks and allow the client to choose the best option which would meet the project requirement. The documentation also provides knowledge about four different designs which can be shared among the team members to explore processes and frameworks which could be used in the future.

Preliminary Design -Younesh Mahaseth

Methodology

1. **Collaboration:**

Some of the tools such as Google Docs, Slides, Microsoft Teams and GitHub can be used for streamlined collaboration and code version control.

2. **Question Bank Assembly:**

Aggregate questions from various sources including online searches, books, journals, interviews, and collaborations can be used to build a diverse and comprehensive question bank.

3. **Data Preparation:**

Python and Pandas can be employed to pre-process and standardise the dataset, ensuring data consistency and resolving any issues. Spelling for domain-specific accuracy has to be checked and corrected.

4. **Data Augmentation:**

The dataset can be expanded using advanced text manipulation techniques such as translation, word addition, random word replacement, and text generation, enhancing model adaptability.

5. **Data Tokenization:**

Utilise Transformers and Tokenizers to break down questions, answer choices, and correct answers into smaller units, ensuring model comprehension of domain-specific language.

6. **Model Preparation:**

The Llama 2 model, "TheBloke/Llama-2-13B-chat-GGML," can be used and parameters can be configured including CPU threads, batch size, and GPU layers, based on hardware and model requirements.

A. **Data Extraction and Organization:**

Extraction questions and answer options from the dataset spanning Environmental Management, Food Science, and The Microbial World will be carried out, forming the basis for interactions with the Llama 2 model.

B. **JSON File Creation:**

The questions and options have to be organised into a structured JSON file, serving as a convenient repository for prompts and associated data.

C. **Prompt Template Creation:**

Clear prompt templates are to be crafted to instruct the Llama 2 model, specifying the expected response format for effective interactions.

D. **Interaction with Llama 2 Model:**

The Llama 2 model will then be engaged using prompt templates, pose questions, provide options, and extract relevant matrices generated by the model in response.

7. Model Fine-Tuning:

The Llama 2 model then have to be fine-tuned using the provided dataset, converting text data into a suitable format compatible with the model.

A. **Dataset Preparation:**

The data for training will have to be structured, converting it into a format that the model can work with, enabling efficient fine-tuning.

B. **Data Splitting:**

The dataset will then have to be splitted into training, validation, and test sets to facilitate model training, hyperparameter tuning, and evaluation.

C. **Fine-Tuning Configuration:**

Adjusting parameters based on validation results, we need to Define hyperparameters, learning rate, batch size, and loss functions for fine-tuning.

D. **Model Initialization:**

The Llama 2 model then needs to be initialised with pretrained weights obtained from the Hugging Face model hub, providing a solid starting point for fine-tuning.

E. **Training Loop:**

Implementation of the training loop will be carried out, iterating over the training dataset, computing gradients, and updating model weights. The model learns to predict correct answers based on input questions and options.

F. **Validation:**

Periodically we need to evaluate the model's performance on the validation set to monitor training progress and prevent overfitting by early stopping if necessary.

G. **Evaluation Metrics:**

Evaluation metrics need to be defined and measured including accuracy, precision, recall, F1-score, and top-k accuracy to assess model performance.

H. **Hyperparameter Tuning:**

Fine-tuning hyperparameters needs to be done like learning rate and batch size based on validation results, iteratively optimising model performance.

I. **Model Checkpoints:**

Model checkpoints will be saved at regular intervals during training for resuming training from specific points and avoiding progress loss due to interruptions.

J. **Early Stopping:**

Early stopping will be implemented based on validation results to prevent overfitting and maintain model generalisation.

K. **Final Evaluation:**

Evaluation of the fine-tuned model on the test set to assess its performance on unseen data, utilising predefined evaluation metrics will be carried out.

8. **Relevant Matrices for Model Evaluation:**

Precision, recall, F1-Score, top-k accuracy, and confidence score analysis will be implemented to comprehensively evaluate model responses and identify areas of model confidence and correctness.

Importance of Quantized Model: Using the quantized version of the Llama 2 model is critical for efficient execution on consumer hardware. It optimises resource usage, allowing the model to run smoothly on a variety of devices. Quantization reduces precision while maintaining model performance, making it a practical choice for real-world applications. This choice ensures that your project remains accessible and performs effectively on a wide range of hardware configurations, contributing to its success.

Design constraints

- **Resource Availability:** Adequate computational resources are necessary for effective data processing and model training.
- **Model Compatibility**: The Llama 2 model must be compatible with the selected hardware and software environment.
- **Data Quality:** The success of data-driven processes depends on the quality and accuracy of the collected data.
- **Validation Reliability:** The validation process must be reliable to ensure accurate model assessment.
- **Quantization Suitability:** The chosen hardware should support the usage of selected quantized models.

Specifications:

1. **Hardware and Software:**

Adequate computing power with GPUs, Python environment with Pandas and Transformers, Collaboration tools: Google Docs, Microsoft Teams, GitHub.

2. **Data Sources and Quality:**

Diverse sources for question bank assembly, High-quality data for targeted domains.

3. **Model Configuration:**

Llama 2 model selection with hardware-specific parameters.

4. **Training and Evaluation:**

Structured dataset preparation, fine-tuning with hyperparameter tuning, Metrics: accuracy, precision, recall, F1-score, top-k accuracy.

5. **Quantization:**

Use of a pre quantized model.

6. **Domain Focus:**

Domain-specific accuracy for targeted domains.

7. **Collaboration and Documentation:**

Efficient use of collaboration tools, Detailed documentation of workflows and configurations.

Design 2 -Arsha Manoj

Methodology

1. **Collaboration**

Throughout the completion of the project, our team communication shall be characterised by conducting regular team meetings and open discussions at least twice a week. The allocated session for the project on Monday from 6:30-8:30 shall be utilised for decision making and conflict resolution which would help to establish rapport and maintain a professional relationship.

Apart from these, a project management platform named Trello Board can be used which offers a variety of features to share ideas, add checklists and move work forwards.

For document management Google Workspace will be used where Google Drive, Docs and sheets will be used to store the data, highlight meeting notes and organise project related documents.

2. **Question Bank Assembly**

a. Sources of Questions

As the topic for question collection was different from our domain, the topic can be categorised to three main areas for proper key word identification and diverse online sources will be relied on to ensure that each person collected non overlapping questions.

Questions posted by journalists and researchers in relevant publications and questionnaires conducted earlier can be explored to ensure credibility and access the data in the required format.

b. Collection Methods

Throughout the collection process, ethical considerations will be given paramount importance and keyword analysis along with web scraping shall be utilised. The scraped data can then be stored into required formats such as CSV or database according to the requirement of the model.

3. **Data Preparation**

a. Data Cleaning

Cleaning and pre-processing of the dataset involves removing irrelevant characters, handling special tokens, and formatting the data into a suitable structure. Spelling check could be conducted manually, and outlet detections and missing data can be handled with the use of python libraries like NumPy, Scikit-learn.

b. Data Augmentation

Retrieval Augmented generation (RAG) can be used to address this challenge as we need to retrieve only those pieces of content that are most relevant to us and it is divided into three subsystems Index, Retrieval and Augment, each playing a crucial role in the overall process.

Langchain is a powerful open-source framework which contains multiple components and multiple modules to train a large language model [4].

- The LangChain CSVLoader allows the user to load the data and split the csv file into rows [6].
- The RecursiveCharacterTextSplitter from LangChain allows chunking and splitting the data into smaller pieces [6].

c. Data Tokenization

TikToken is a tokenizer offered by open AI which can be used to merge splits and measure chunk size.

4. Prompt Preparation

a. Design Approaches

LangChain's response schema can be used to generate prompt and read output from the model.
The textual chunks will be converted to embeddings using OpenAI embeddings or by hugging face embeddings. This can be stored in vector store by using FAISS (Facebook AI Similarity Search) or by using Chroma which is lightweight.
The augment system in RAG can be used to create the initial prompt and which would merge with the context retrieved from the vector store and create an input for the model. This prompt is then fed into the LLM and after processing the prompt, the LLM generates and returns the response [5].

b. Preparation Tools

The initial parser can be defined with the help of response schema, after which the format for instructions can be generated with the help of. get_format_instructions () command from LangChain and define final prompt using ChatPromtTemplate.
Once the customer prompts are designed, Conversational Retrieval Chain can be initialised which loads context from then memory [5].

5. Ethics Consideration Preparation

As there are no dedicated tools for ethical considerations in prompt preparation, a combination of existing tools can be used to generate contents that are not harmful and offensive. Libraries like Fairness Indicators and AI Fairness 360 can be used to address fairness concerns and transparency.

6. Model Preparation

The open-source Llama-7b-chat model can be used both in hugging face transformers and Langchain. However, access must be requested through the Meta Website.
Parameters like GPU and CPU must be configured as ~8GB of GPU is required to run the model.

7. Input to Llama2

All the necessary libraries must be imported which includes streamlit, open AI, document loaders, FAISS etc. The collected data must be converted into CSV format which can be loaded with the help of a CSV loader in LangChain.

8. Computational Resources for inference

Understanding the GPU requirements for each model is the hardest part as the 7b or 13 b models run on most GPUs while the 70b may require higher quantisation.
The trained model can be made to make predictions or draw conclusions on unseen data. Similarity search using metadata can be carried out during this.

9. Retrieval from Llama2

Retrieval is the most important step in the RAG process. The retrieval system captures the user's question or statement and transforms it into vector format similar to the ones already stored in vector format while training. It then searches the vector store for embeddings that are similar to the query [10].
The system would then return the top matches ensuring the apt response. Goal of retriever is to efficiently find information searched by the user. LangChain supports this easily by simple lines of code.

10. Llama2 fine tuning

Even Though RAG is easy to prototype, it has many failures, and it often lies in the retrieval stage. But retrieval can be made better with the help of fine tuning. We need to have both positive and negative examples as pairs of questions that should be close to each other and far from each other for embeddings.

With LlamaIndex by using the SimpleNodeParser module, we can automatically generate these sets of examples which can be used as training signals.

11. Evaluation of Results

The fine-tuned model can be compared and evaluated with the help of two main metrics,

- Hit Rate Metric where we retrieve top k documents with the query for each pair.
- InformationRetrievalEvaluator from sentence_transformers which provides a suite of metrics such as accuracy, precision, recall at different top k values.

12. Comparative Analysis

Different versions of Llama2 models available from open sources can be used and can be compared for performing the specific task required for the project. Strength and weakness of each model can be explored with the help of evaluation and check how they align with the project goals.

13. Visualisation and comparative analysis

LangChain interactions can be viewed with good AI with the help of Ought's ICE Visualizer which tells from the colouring which part of the prompt are hardcoded and which part are templated substitutions.

Visualisations like heat maps could be used to compare the performance of models across different metrics in which the colour intensity can be used to indicate the level of performance.

14. Feedback Loop

There are websites which offer built in feedback widgets or plugins which can be integrated into the model interface for collecting feedback directly from users.

Design constraints

1. Data and Computational resource requirements: Effective working of LangChain demands substantial data and computational sources [8].
2. Chance of bugs: As the LangChain is a new library, there may be chances of issues which have not been found and resolved [9].
3. Limited support for all languages: As the LangChain is a python library, it does not support the model making if other languages are preferred [8].
4. Lack of Expertise: It requires time and effort to learn the components in LangChain effectively.
5. Outdated foundations: The prompt techniques and workflow of LangChain are based on early LLMs and so its capabilities may lag behind the new powerful models [9].

Specifications

1. Collaboration tools
   Google Drive, Google Docs, Google Sheets, Trello Board
2. Software Requirements
   Installed LangChain python package, API keys for accessing API.
3. Hardware Requirements
   Suitable GPU for running the selected version of model.
4. Data Requirements
   Collected data from reliable sources converted into required format, Document Loaders, Text Splitters, Retrievers, Vector Stores

5. Performance Metrics
   HitRate Metric, InformationRetrievalEvaluator
6. Documentation
   Proper codes for each module which includes data connection module, chains module, Agents module, Memory module, Callback module.
7. Security requirements
   Ethical Considerations while designing prompts, Privacy protection for input data.

Design 3 - Aayush Joshi

Methodology

1.Collaboration:

We will meet face-to-face every Monday from 6:30-8:30pm to discuss projects, brainstorm ideas, and build relationships. The primary communication platform utilised will be Microsoft Teams. Conversation and file sharing is simpler compared to other platforms. The team will be sharing word documents and excel sheets on it. Code will be handled using Google Colab.

2.Question Bank Assembly:

a. Sources for Questions:

An abundance of **open-source datasets** pertaining to environmental science are readily accessible. The utilisation of these datasets can facilitate the generation of multiple-choice questions (MCQs) or enhance the overall quality of pre-existing MCQs.

b. Collection Methods:

While data scraping could be a way to collect the required information from the website, it is important to understand the terms and conditions of the source from where the data is scraped. For the research purpose, python libraries like **Beautiful Soup** can be implemented to scrap and convert the questions into a standard format.

3. Data Preparation:

a. Data Cleaning:

One way to enhance the cleanliness of text-based data is by identifying and eliminating duplicate entries in the dataset and by performing spell checks to verify the accuracy of the input provided to the model. For smaller datasets, Python libraries like **pandas** and **NumPy** can be utilised to filter and refine the data.

b. Data Augmentation:

Data augmentation involves flipping, deleting or inserting text to make the content more robust to the model. **TensorFlow** library can be used to invoke strings related queries.

c. Data Tokenization:

Tokenization helps in eliminating noise in the data and infers key phrases by classifying texts into categories. **Amazon Comprehend** processes the text stored in the S3 data lake and extracts key phrases from a text script.

4. Prompt Preparation:

a. Design Approaches:

Since, the data for this task is limited, we will utilise domain adaptation design approach to prompt the model. **Domain adaptation** is the method of using a general prompt that can be used for a number of different jobs.

b. Preparation Tools:

The tools required for prompt preparation includes **Amazon S3 bucket** where the domain dataset will be stored, as well as importing **boto3** library which helps in accessing and managing resources in S3 bucket. The dataset must be in json format.

5. Ethical Consideration Preparation

Based on extensive research, it is observed that Llama2 models violation ratio is minimum as compared to other models. The violation ratio for LLama2 7b and 13b models is less than 5 percent as compared to other models averaging 20 to 25 percent.[13]

6. Model Preparation:

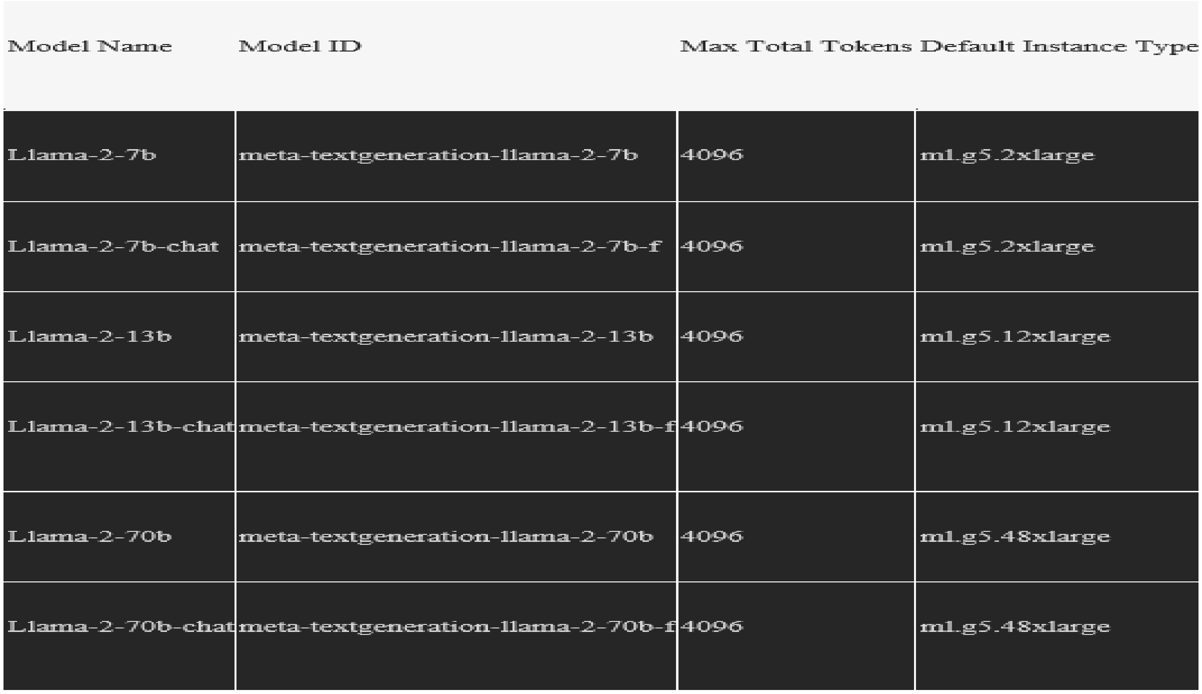
Due to the limited amount of data, **Llama2 7b and Llama2 7b** chat models are preferred because it requires less computational power. We will be utilising the **jumpstart** model provided by Amazon sagemaker studio. [14]

7. Input to Llama 2:

The data will be converted into json format and stored in the **S3 bucket**. Before fine tuning the model, we will invoke the endpoints by providing inputs, parameters and response to the model.

8. Computational Resources for Inference:

The computational resources required for each model are presented in the table below. Hence, the ideal instance type for the project would be **ml.g5.2xlarge**.



Model Name	Model ID	Max Total Tokens	Default Instance Type
Llama-2-7b	meta-textgeneration-llama-2-7b	4096	ml.g5.2xlarge
Llama-2-7b-chat	meta-textgeneration-llama-2-7b-f	4096	ml.g5.2xlarge
Llama-2-13b	meta-textgeneration-llama-2-13b	4096	ml.g5.12xlarge
Llama-2-13b-chat	meta-textgeneration-llama-2-13b-f	4096	ml.g5.12xlarge
Llama-2-70b	meta-textgeneration-llama-2-70b	4096	ml.g5.48xlarge
Llama-2-70b-chat	meta-textgeneration-llama-2-70b-f	4096	ml.g5.48xlarge

Figure 1: Model comparison with default instance type

Source: Adapted from [15]

9. Retrieval from Llama 2:

We can create an API using **AWS Lambda**, which will provide a platform to retrieve output by inserting prompts.

10. Llama 2 Fine Tuning:

To fine tune the dataset present in the S3 bucket, we will use **JumpStart Estimator** to fit the training data. In this step, we will fine tune hyperparameters involved in fitting the model. Based on the model requirement numerous parameters like number of epochs, learning rate, batch size, and permissible input length can be altered based on specifications.

11. Evaluation of Results:

The evaluation metric that can be used to check the accuracy between the result is **BLEU score** which helps in determining accuracy before the model was fine-tuned as compared to after the model was fine-tuned.

12. Comparative Analysis:

A detailed analysis about multiple Llama 2 models will be executed based on parameters like time and storage consumption and accuracy of the results. Moreover, different values of hyperparameters will be tested to figure out the optimum solution to meet the scope of this project.

13. Visualisation of the results and Comparative Analysis:

BLEU score can be visualised by using a bar chart in Tableau. We can also display a regressive line displaying the tendency of increase in accuracy with increase in number of epochs.

14. Feedback Loop:

RLHF or reinforcement learning from human feedback can be done by using AWS Lambda which will help in evaluating the output provided by the Llama2 model. [12]

Design constraints

1. Access to Amazon Web Services: While implementing the design, it is necessary to have full access to certain policies of AWS.
2. Pricing: AWS is a paid service. Since, the deployment of generative AI models require high end equipment, pricing could be a matter of concern.
3. Resource Limitations: Performance - resource tradoff is always a concern when it comes to performing high computational tasks.
4. Debugging Error: The code utilised in this design, might require alignment with AWS resources, while debugging code would be a challenge since not many resources are available to troubleshoot a problem in the code.
5. Data authenticity: While scraping data is mandatory to fact check the data through certain experts in that field.
6. Limited Database: While training a dataset, it is important to feed more information to the model, to get better accuracy.
7. Time-Accuracy tradeoff: Based on the available computing resources, the dataset might take time training on a higher number of iterations.

Specifications

1. Collaboration Tools:
   Microsoft Teams, Google Drive, Microsoft Teams, Google Colab
2. Software Requirements:
   AWS SageMaker, Full policy access to IAM user, Jumpstart Model, S3 bucket
3. Hardware Requirements:
   Basic hardware requirement, since the code will run on AWS instances.
4. Data Requirements:
   - Question Bank sourced from authentic websites.
   - Human feedback collected through AWS Lambda API.

5. Performance Metrics:
  - BLEU
6. Documentation:
  - Described and documented code which aligns with the design idea.
7. Security Requirements:
  - Terms and conditions of the source website.

Design 4 - Kan Asvasena

Methodology

1.Collaboration:

In order to enhance teamwork, our methodology incorporates both in-person and digital collaboration tools. Our team is committed to meeting face-to-face every Monday from 6:30-8:30pm. Apart from these in-person sessions, our collaboration will primarily be conducted online. To streamline teamwork, we will utilise Google Workspace, which includes Google Calendar, Drive, Docs, and Sheets, to store, share, and collaborate amongst team members. Our primary communication tool will be Slack Business+. For the management of code, we have chosen GitLab Ultimate, which offers features such as Auto DevOps, Kubernetes integration, and extensive CI/CD pipelines.

2.Question Bank Assembly:

a. Sources for Questions:

Given that our expertise does not encompass fields such as Environmental Management, Food Science, or The Microbial World, we will enlist scholars who specialise in these domains to ensure high-accuracy and reliable databanks. These scholars will draw questions from esteemed peer-reviewed journals and will also participate in seminars to evaluate question quality.

b. Collection Methods:

To meet the demands for flexibility, scalability, high availability, and integration, this project will utilise a data lake architecture. This approach enables the aggregation, storage, and analysis of various data types—including structured, semi-structured, and unstructured data—while incorporating robust security measures to safeguard data integrity and confidentiality.

3. Data Preparation:

a. Data Cleaning:

To ensure the accuracy and reliability of our dataset, we will implement a data pipeline using Apache Kafka for real-time stream processing. The goal is to identify and rectify errors, outliers, or inconsistencies, thereby enhancing the quality of the data that will feed into subsequent processes.

b. Data Augmentation:

As larger dataset benefits model performance and more robust evaluations, data augmentation techniques, including Back Translation, Random Insertion, Random Replacement, and Text Generation, will be employed. Scholars responsible for data collection will subsequently validate the augmented data for accuracy.

c. Data Tokenization:

We will employ the Hugging Face Tokenizers library to streamline the tokenization process, offering flexibility through adjustable parameters. Renowned methods like Byte Pair Encoding (BPE) will be applied for their proficiency in managing expansive vocabularies. To enhance efficiency, we'll incorporate Hugging Face's Accelerate library, enabling swift distribution of tokenisation tasks across multiple CPUs and GPUs, thus expediting the data preparation stage.

4. Prompt Preparation:

a. Design Approaches:

To ensure a comprehensive and adaptable set of prompts, we will employ a multi-faceted design strategy grounded in academic research. Techniques such as 0-shot and few-shot learning will be incorporated to build prompts that can generalise well across various scenarios.

b. Preparation Tools:

We will utilise the same data lake architecture for storing and managing prompt designs. This will allow for efficient retrieval and modification of prompts, as well as real-time updates and collaborative editing by the team, thereby streamlining the preparation process.

5. Ethic Consideration Preparation

To rigorously test the ethical boundaries of our AI model, we will employ ethic banging tests that include ethical prompts such as Toxicity, Language Polarity, and Hurtful Sentence Completions [11]. These prompts will simulate edge-case ethical scenarios for evaluation.

6. Model Preparation:

We plan to utilise all variants of the Llama 2 model—specifically the 7B, 14B, and 70B versions, in both standard and chat configurations. We will implement quantisation techniques to reduce computational demands, in addition to fine-tuning hyperparameters. Each model version will be subjected to a comprehensive array of questions and prompt designs to evaluate its efficiency and effectiveness.

7. Input to Llama 2:

We will convert our question bank and prompt designs into the Hugging Face Datasets format, underpinned by Apache Arrow. This facilitates zero-copy reads of large datasets without memory constraints, thereby optimising speed and efficiency [2] [1]. Additionally, we'll employ Hugging Face's Accelerate to enable the same PyTorch code to be run across any distributed processing [3].

8. Computational Resources for Inference:

We will use a cluster of Amazon EC2 P5 instances, powered by the latest NVIDIA H100 Tensor Core GPUs. This will provide the computational horsepower necessary as well as ability to scale horizontally.

9. Retrieval from Llama 2:

We will extract inferences from Llama 2 using bespoke Python scripts designed to interact seamlessly with the model's API. Subsequently, the retrieved data will be securely stored in the data lake for further analysis.

10. Llama 2 Fine Tuning:

We will utilise the advanced PEFT (Progressive Ensemble Fine Tuning) technique to enhance the model's performance iteratively. Continuous monitoring tools like AWS CloudWatch will be employed to ensure that the system runs smoothly and maintains the desired level of performance during the fine-tuning process.

11. Evaluation of Results:

Evaluation metrics will be calculated using a custom Hugging Face Evaluate library. Key performance indicators (KPIs) will include precision, recall, F1-score, and Top-k accuracy.

12. Comparative Analysis:

A rigorous comparative analysis will be conducted to assess the efficiency of different versions, prompt designs, hyperparameter tuning strategies, and fine-tuning methods. This will enable us to discern the most suitable configurations for achieving high levels of both effectiveness and efficiency.

13. Visualisation of the results and Comparative Analysis:

Data visualisation will be executed using D3.js to create interactive dashboards, enabling stakeholders to delve into the model's performance metrics. Techniques like heatmaps and word clouds will also be utilised to provide a graphical representation of the model's strong and weak points.

14. Feedback Loop:

User feedback will be collected through an interactive web portal developed in React.js. Feedback will be categorised and analysed to detect patterns or areas that require attention. Based on this feedback, the model will undergo iterative refinement. Simultaneously, any new ethical considerations or biases that come to light will be addressed, and mitigation strategies will be updated accordingly.

Design constraints

8. Budget Limitations: The project must adhere to a predetermined budget, which could potentially limit the computational resources or third-party services we can employ.
9. Team Expertise: While our team possesses a background in machine learning, data science, and software engineering, limitations may arise in areas requiring highly specialised knowledge. This could necessitate additional training or external consultation, potentially affecting timelines and budget allocations.
10. Time Constraints: All project milestones must be achieved within the stipulated timeline, thereby necessitating efficient project management strategies.
11. Data Privacy and Security: Strict compliance with Australian Privacy Act 1988 and other privacy regulations will be mandatory, imposing constraints on data collection and storage methods.
12. Technical Limitations: The choice of frameworks, languages, and architectures may be constrained by the capabilities of the existing infrastructure.
13. Expert Availability: The need for specialised scholars in fields like Environmental Management, Food Science, and The Microbial World may introduce constraints related to availability and scheduling.
14. Scalability: The architecture and tools used must be capable of scaling to handle increased data volumes or user demands, without compromising performance.
15. Ethical Guidelines: The project must adhere to ethical standards, particularly concerning data sourcing and model training, which may introduce additional layers of complexity.

Specifications

1. Collaboration Tools:

Scheduling: Google Calendar, File Sharing: Google Drive, Document Collaboration: Google Docs, Google Sheets, Communication: Slack Business+, Code Management: GitLab Ultimate

2. Software Requirements:

Backend: Python 3.x, Apache Kafka, AWS Cloud Services, Frontend: React.js, D3.js, Data Science: Hugging Face Transformer, Hugging Face Tokenizers, Hugging Face Datasets, Hugging Face Accelerate, Apache Arrow

2. Cloud Infrastructure Requirements:

Compute: A cluster of Amazon EC2 P5 instances equipped with NVIDIA H100 Tensor Core GPUs, Data Lake: Utilising AWS Lake Formation for building and securing the data lake, backed by Amazon S3 for scalable storage. Amazon Glue for cataloguing, and AWS Athena for SQL-based queries. Optionally, Amazon Kinesis for real-time data streaming.

3. Data Requirements:

Question Bank sourced from peer-reviewed journals retrieved by hired scholars, User feedback collected through React-based web portal.

4. Performance Metrics:

Precision, Recall, F1-Score, Top-k accuracy

5. Documentation:

Comprehensive code documentation following best practices, Detailed project reports and whitepapers for stakeholders.

6. Security Requirements:

Encryption of sensitive data, Secure access controls for the data lake

Justification of Designs

Design One Younesh

- Pros: The design utilises the existing quantized version of the Llama2 model and hence can be runned easily on google collaboration platform for the entire process of the project. Also, all the questions and the options can be provided to the model at once just by programming the system to take the input and give out on a single click from the user.
- Cons: The design exports the cleaned dataset into json format which will demand the additional storage. Also, it becomes difficult for the non-programmer to design the prompt while providing the input to the model.

Design Two Arsha

- Pros: As the project requires to test the accuracy using an external dataset, a design should be chosen in such a way that its is easy to connect the LLM with the data.Langchain can simply be installed using pip install and it helps to connect the LLM to external data sources such as files or any other applications with ease.Moreover,the components of LangChain such as document loaders,Prompts,chains,vector databases and agents fasten up the training process[7].
- Cons: It requires proper expertise to implement the LangChain process which is complex and sometimes it is difficult to integrate with existing tools.

Design Three Aayush

- Pros: This design purely implements AWS resources, which helps in managing scalability and ease of deployment. The project demands the ability to scale computing resources up or down based on workload requirements. It provides a baseline to develop and learn Generative AI concepts using Amazon Web Services.
- Cons: The design requires full access to AWS policies. Maintaining the resources might affect the budget of the project.

Design Four Kan

- Pros: This design utilises cutting-edge technologies, offering high functionality and computational power. It also provides the student team with the opportunity to explore and learn these technologies in preparation for their future careers.

- Cons: Although the team consists of second-year data science students, all members come from non-IT backgrounds. Utilising overstacked technologies requires a steep learning curve, time, as well as a significant budget to cover costs.

Justification Summary

The four proposed designs for the Generative AI-based question and answer generation project offer a spectrum of options, each with its distinct merits and challenges. Design One prioritises accessibility and simplicity, utilising the quantized Llama 2 model on Google Colab, yet may require extra storage and pose usability challenges for non-programmers. Design Two emphasises flexibility and integration ease through LangChain, offering adaptability but requiring expertise and possible integration complexities. Design Three opts for scalability and resource management via AWS, aligning with the project's scalability needs but demanding full AWS access and budget considerations. Finally, Design Four employs advanced technologies for enhanced functionality and learning opportunities but necessitates overcoming a potential learning curve and budget constraints. The final choice should harmonise with project objectives, available resources, and team expertise, striking a balance between simplicity and scalability. Careful consideration will lead to the selection of the most appropriate approach for project success.

WORD COUNT- 4392

References

[1] “The 🤗 Datasets Library” huggingface.co. https://huggingface.co/learn/nlp-course/chapter5/1
(Accessed Aug. 20, 2023).

[2] Q. Lhoest, A. V. del Moral, Y. Jernite, A. Thakur, P. von Platen, S. Patil, J. Chaumond, M. Drame, J. Plu, L. Tunstall, J. Davison, et al., "Datasets: A community library for natural language processing," arXiv preprint arXiv:2109.02846, 2021. [Online]. Available: https://arxiv.org/abs/2109.02846.

[3] “Accelerate,” huggingface.co. https://huggingface.co/docs/accelerate/index (accessed Aug. 31, 2023).

[4] O. Mishra, “Using Langchain for question answering on own data,” Medium, https://medium.com/@onkarmishra/using-langchain-for-question-answering-on-own-data-3af0a82789ed
(Accessed Sep. 6, 2023).

[5] M. Kazmi, “Using Llama 2.0, FAISS and Langchain for question-answering on your own data,” Medium, https://medium.com/@murtuza753/using-llama-2-0-faiss-and-langchain-for-question-answering-on-your-own-data-682241488476(accessed Sep. 6, 2023).

[6] Yvann, “Build a chatbot on your CSV data with Langchain and OpenAI,” Medium, https://betterprogramming.pub/build-a-chatbot-on-your-csv-data-with-langchain-and-openai-ed121f85f0cd (accessed Sep. 6, 2023).

[7] A. Taneja, “Ajay Taneja on linkedin: Langchain – Essential Concepts – my notes,” Ajay Taneja on LinkedIn: LangChain – Essential Concepts – my notes, https://www.linkedin.com/posts/ajay-taneja-47727817_langchain-essential-concepts-my-notes-activity-7088594801058562048-Qjdz(accessed Sep. 6, 2023).

[8] “Understanding langchain - A framework for LLM Applications,” ProjectPro, https://www.projectpro.io/article/langchain/894(accessed Sep. 6, 2023).

[9] Y. H. K. (PhD), “Overview of langchain,” Medium, https://medium.com/technology-hits/overview-of-langchain-9f6362707cd0(accessed Sep. 6, 2023).

[10] H. Sahota, Deci, and H. Sahota, “Retrieval augmented generation using Langchain,” Deci, https://deci.ai/blog/retrieval-augmented-generation-using-langchain/ (accessed Sep. 7, 2023).

[11] “Evaluating Language Model Bias with 🤗 Evaluate,” huggingface.co. https://huggingface.co/blog/evaluating-llm-bias (accessed Sep. 7, 2023).

[12] O. Daniels-Koch and R. Freedman, “The expertise problem: Learning from specialised feedback,” arXiv.org, https://arxiv.org/abs/2211.06519 (accessed Sep. 8, 2023).

[13] H. Touvron et al., “Llama 2: Open Foundation and fine-tuned chat models,” arXiv.org, https://arxiv.org/abs/2307.09288 (accessed Sep. 8, 2023).

[14] MACHINE LEARNING WITH CLOUD PLATFORMS, http://lib.pnu.edu.ua:8080/bitstream/123456789/11065/1/AISTIS-2021-Kozlenko.pdf (accessed Sep. 8, 2023).

[15] S. Engdahl, “Blogs,” Amazon, https://aws.amazon.com/blogs/machine-learning/llama-2-foundation-models-from-meta-are-now-available-in-amazon-sagemaker-jumpstart/ (accessed Sep. 8, 2023).