School of Computing
FACULTY OF ENGINEERING AND
PHYSICAL SCIENCES

UNIVERSITY OF LEEDS

Final Report

Enhancing University Learning with Retrieval-Augmented Generation and GPT-3.5 Fine-Tuning

Abdul Karim Abbas

Submitted in accordance with the requirements for the degree of
BSc, MEng Computer Science with Artificial Intelligence

2023/24

COMP3931 Individual Project

The candidate confirms that the following have been submitted:

Items	**Format**	**Recipient(s) and Date**
Final Report	*PDF file*	*Uploaded to Minerva (30/04/2024)*
Scanned participant consent forms	*PDF file*	*Uploaded to Minerva (25/04/2024)*
Link to online code repository	*URL*	*Sent to supervisor and assessor (30/04/2024)*

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student) ____________________

Summary

This dissertation explores the integration of Retrieval-Augmented Generation (RAG) and fine-tuning techniques with GPT-3.5 to enhance digital learning in higher education. Given the rapid adoption of digital platforms in educational settings, there is a developing need to enhance the interactive capabilities of educational chatbots to support students effectively. Current AI models, while robust, often fall short in responding to domain-specific queries that require up-to-date knowledge and often need continuous specialised training.

This research aimed to refine GPT-3.5s responsiveness and adaptability to the dynamic nature of educational content by implementing retrieval-augmented generation to enhance the chatbot's response accuracy and relevance; ensuring that the chatbot could handle a wide array of student inquiries on a specialised domain with precision.

The methodology involved the integration of RAG to supplement GPT-3.5's knowledge base, enabling it to retrieve and utilise the most relevant information from a curated dataset of university lecture materials. Additionally, the project explored fine-tuning techniques to tailor GPT-3.5's responses to the specific language and needs of university-level material.

The project achieved the development of an enhanced chatbot that successfully leverages retrieval-augmented generation to access and integrate targeted information into its responses, thereby significantly improving the specificity and accuracy of its assistance. Fine-tuning was applied to better understand and process complex academic queries specific to university courses, achieving a high degree of contextual alignment to the source lecture and response relevance. The model was rigorously tested and validated within an educational setting, demonstrating improved performance in generating accurate responses compared to the standard GPT model. Feedback from real university students confirmed that the enhancements could substantially improve their learning experience by providing precise and informative responses.

Acknowledgements

I want to express my gratitude to my project supervisor, Professor Eric Atwell, for his wisdom and guidance throughout this project. His expertise in this field was truly invaluable.

I am also grateful to Dr. Nishant Ravikumar for his essential intermediate feedback as project assessor. Their insight always helped ensure I was on the right path.

Finally, I thank my mom, dad, and sister for their unconditional love and support throughout this project and university as a whole.

Table of Contents

Chapter 1
Introduction and Background Research

1.1 Introduction

In education, particularly within the context of digital learning, chatbots represent a promising opportunity for supplementary student support. Large language models such as GPT-4 (Open AI, 2023) and BARD (Google) are already emerging as valuable tools for student support. These advanced AI models are increasingly recognised for their ability to understand and generate human-like text, making them ideal for interactive learning environments. They offer real-time responses to inquiries, can guide through complex concepts, and provide easy access to a vast array of information. By leveraging these technologies, educational institutions can deliver more responsive and adaptive learning experiences, catering to the diverse and unique needs of students across different learning settings. This integration not only improves accessibility and engagement but also allows educators to focus on other more demanding tasks, enhancing overall educational quality.

However, a significant challenge remains in the domain-specific limitations of chatbots such as GPT-3.5. When tasked with answering specific questions—such as those about new developments beyond its last training data or detailed queries about a particular university course—GPT-3.5 frequently encounters difficulties in providing accurate responses and often resorts to generating incorrect information that mimics what a correct response may look like. This issue highlights the necessity for ongoing updates and specialised training for AI models to maintain their relevance and effectiveness.

This project proposes an intuitive approach to overcome these limitations by integrating the capabilities of GPT technology, Retrieval-Augmented Generation (RAG), and fine-tuning techniques to enhance the chatbot's knowledge base with a tailored, supplementary dataset; thereby allowing the chatbot to gain the ability to access the most pertinent and up-to-date information across various domains without the need for constant retraining.

Additionally, the project will undertake a series of experiments to both quantitatively and qualitatively assess the effects of this framework on the chatbot's performance, focusing particularly on enhancing the coherence, relevance, and accuracy of its responses within educational settings.

Ultimately, the success of this project will be determined by its ability to effectively integrate advanced AI functionalities, making educational chatbots more responsive and adept at handling the specificities of academic discourse.

1.2 Digital Learning and AI

The advancement of digital learning is evident through the growing interaction with online technological platforms. Massive Open Online Courses (MOOCs) such as Coursera, introduced in 2018, have played a pivotal role in access to digital education, allowing learners worldwide to access courses from leading universities at minimal or no cost. Similarly, Learning Management Systems (LMS) such as Blackboard, launched with new features in 2023, have become integral to organising and delivering educational content, facilitating communication between educators and students, and tracking student progress in both traditional and online learning environments. Additionally, the rise of educational apps, for instance, Duolingo's language learning platform in 2019, highlights the shift towards accessible and interactive learning experiences, enabling users to study new languages on-the-go with engaging, gamified content.

Digital learning's evolution continues as it embraces increasingly complex data analytics and feedback mechanisms. By leveraging big data, educators and institutions can uncover insights into learning patterns, predict student performance, and offer targeted interventions to support learners at risk of falling behind. The potential for data-driven customisation in digital education paves the way for more nuanced and effective learning pathways, marking a significant leap forward in educational technology's ability to cater to diverse learning styles and needs.

AI chatbots can significantly contribute to the scalability of digital learning, enabling the management of large volumes of student interactions without the need for extensive human resources. This is particularly beneficial in scenarios where an instructor’s attention is spread across many students. Additionally, according to Kooli (2023), chatbots have the potential to personalise the learning pathway by analysing the students' learning abilities, suggesting additional learning resources, and providing revision on challenging topics, thereby enhancing overall learning outcomes.

To further understand how these AI systems can be optimised for educational purposes, the following chapter delves into the GPT transformer architecture, which underpins models like GPT-3.5. Outlining the basic mechanisms of how GPT models process and generate language will provide a foundation for their improvement.

1.3 Transformer Architecture

Introduced in the paper "Attention is All You Need" by Vaswani et al. in 2017, the Transformer has since become the foundation neural network architecture upon which language models, including the Generative Pre-trained Transformer (GPT) series, are built.

Transformers were developed to solve the problem of sequence transduction: The task of transforming an input sequence to an output sequence (Graves, 2012). Examples of these tasks include speech recognition, text-to-speech, and language translation.

The Transformer architecture marks a major shift in how machine learning models handle data, particularly when compared to older models that processed data sequentially. This model utilises self-attention mechanisms and is crucial for understanding the context of each element within the sequence. The architecture itself is divided into two primary components: the encoder and the decoder. Both of these components are comprised of several identical layers that feature self-attention mechanisms. This design enables the Transformer to process sequences of data in parallel, significantly enhancing efficiency and scalability compared to traditional models that process sequences in series.

1.3.1 Encoder/Decoder

The encoder's aim is to process the input text and encode it into context-aware representation. Each layer in the encoder stack includes two main sub-layers: a feed forward neural network and the self-attention layer.

The decoder, on the other hand, is tasked with generating the output text from the encoded representation. Each layer in the decoder stack has three main sub-layers: a self-attention layer, an encoder-decoder attention layer, and the feed forward neural network.

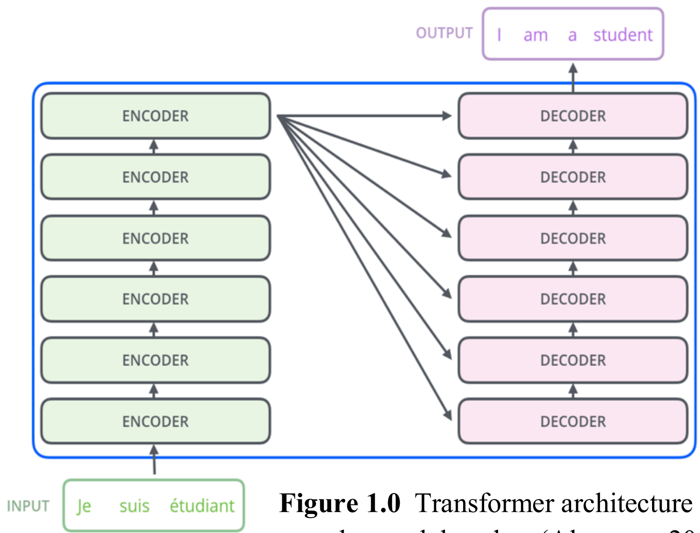

Figure 1.0 Transformer architecture showing encoders and decoders (Alammar, 2018)

1.3.2 Self-Attention

Self-attention allows a model to weigh the contexts of different parts of an input sequence. Beginning by converting each input word into a vector using an embedding algorithm, such as word2vec, the computation of three other vectors for each given word is determined: a Query vector, a Key vector, and a Value vector (Collis, 2017). For any given word, the encoder can then calculate the dot products of its Query vector with the Key vectors of all words in the sequence. These dot products give a set of scores that measure the relevance of other words to the given word.

For example, in the sentence:

`“The plant wilted because it was not watered.”`

When processing the word "it," the Transformer uses self-attention to accurately link "it" to "plant." As the model examines each word, iterating each position in the input sequence, self-attention enables it to consider the scores of other positions within the sequence to improve the encoding of the current word (Alammar, 2018).

1.4 Generative pre-trained transformer

The Generative Pre-Trained Transformer (GPT) leverages the foundations of the Transformer architecture to achieve capabilities in language processing. At its core, GPTs apply the Transformer model's design, particularly its reliance on self-attention mechanisms and Feed-Forward Networks (FFNs), to process and generate text. However, GPTs introduce pre-training on a large corpus of text, followed by fine-tuning on specific tasks. This approach allows GPTs to learn a wide range of language patterns, structures, and nuances during its pre-training phase, effectively internalising a general understanding of natural language. The subsequent fine-tuning phase adapts this general knowledge to specific tasks, enhancing the model's performance on tasks such as text completion, translation, question answering, etc…

1.4.1 Architecture and Design

According to the paper published by OpenAI (Radford et al., 2018) introducing GPT-1, the GPT architecture is built upon a foundation of transformer blocks. Each transformer block in the GPT model functions as a distinct layer, and these layers are stacked on top of each other to form the complete model.

These layers allow the model to focus on different parts of the input sequence as it processed, effectively mimicking the way humans pay attention to different aspects of a sentence when understanding or generating language. The multi-headed nature of this attention mechanism enables the model to explore different subspaces of attention simultaneously in parallel, ultimately increasing its understanding and processing capabilities.

1.4.2 Unsupervised Pre-training

The GPT model uses unsupervised learning techniques to pre-train on a significant amount of text input. The model gains the ability to anticipate the incoming word in a sequence solely based on the preceding words seen during pre-training (Hendy et al., 2023).

According to Lund (2023), the use of unsupervised learning techniques, instead of supervised learning which requires labelled data to teach models how to make predictions, allows the GPT to learn patterns and underlying structures in language without explicit instruction. This approach is advantageous for natural language processing due to the availability of large unlabelled datasets.

During pre-training, the GPT is tasked with predicting the next word in a sequence based on the context provided by the preceding words. This language modelling objective encourages the model to understand syntax and context.

For example, in the sentence:

“`The dog sat on the ____` ”

The GPT learns to predict that a plausible word to complete the sentence could be "floor". This capability stems from the self-attention mechanism within the Transformer architecture, allowing GPT to evaluate and weigh the significance of all preceding words, thereby creating a nuanced understanding of language structure and context. This approach ensures the GPT's ability at predicting probable words in sentences.

1.5 Fine-Tuning

Fine tuning is the process of training a large existing model on a smaller and more specialised dataset. This process allows the model to adapt its parameters to the nuances of a particular task, greatly enhancing its performance for specialised applications. The significance of fine-tuning lies in its ability to leverage the vast, generalised knowledge acquired by the model during its initial training phase and tailor it to specific needs, resulting in improved accuracy, relevance, and efficiency in tasks such as text classification, question-answering, and sentiment analysis.

1.5.1 Supervised Fine-Tuning

Unlike unsupervised pre-training, fine-tuning is supervised. The GPT is introduced to labelled data from the target task, which could range from text classification to question answering. This step aligns the model's generalised understanding of language, acquired during pre-training, with the requirements of the specific task. The fine-tuning process affects each layer of the Transformer architecture, from the input layer to the output. The self-attention mechanisms, fundamental to the Transformer model, plays a vital role during this phase. They are adjusted to focus more on specific patterns within the new data that are crucial for performing the specific task effectively. For instance, if the task is to identify the sentiment of a sentence, the self-attention mechanism learns to give higher weight to words and phrases that are strong indicators of sentiment.

Each layer of the model, having previously learned general patterns of language during pre-training, now refines these patterns to be more task-specific (Figure 1.1). Lower layers, which initially learn basic syntactic features, begin to recognise syntax that is pertinent to the task. Middle layers adapt to capture semantic relationships important for the task, and upper layers fine-tune their understanding of complex language structures and context specific to the task requirements.

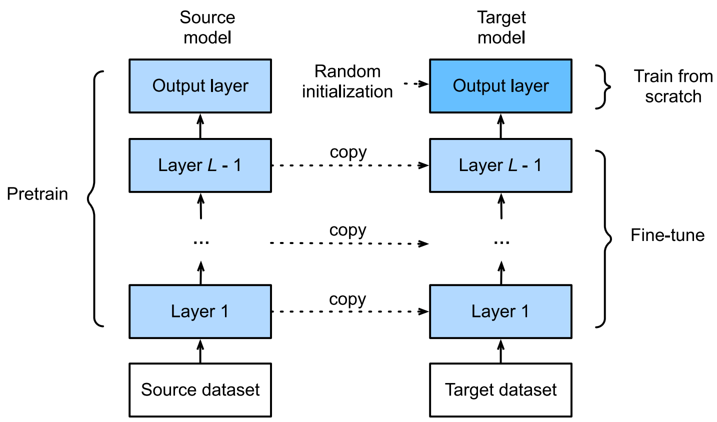

Figure 1.1 Fine tuning (Zhang et al., 2023)

Fine-tuning involves a delicate adjustment of the pre-trained parameters, where the learning rate is typically much lower than during pre-training. This ensures that the model does not forget the valuable understanding it has gained but rather build upon it to specialise in the task. The process is relatively quick compared to pre-training; due to the foundational knowledge the model already possesses.

1.5.2 Fine-Tuning in NLP Models

In the context of Natural Language Processing (NLP) and Artificial Intelligence the comparison of Retrieval Augmented Generation (RAG) and fine-tuning techniques in enhancing NLP models has been a subject of recent studies, including the work by Balaguer et al., (2024). This research highlights the complementary strengths of both approaches in improving the performance of AI systems for complex language tasks. While RAG focuses on augmenting model responses with externally retrieved information, fine-tuning adjusts the model's internal parameters to better align with the specific characteristics of the task at hand.

One notable application of fine-tuning in NLP models is in the customisation of chatbots for customer service. Highlighted by Gnatyuk (2023), fine-tuning a general-purpose model like GPT-3.5 on a dataset comprising customer service interactions and FAQs specific to a company, the chatbot can provide more accurate and contextually relevant responses to customer queries. This approach can be applied to educational dialogue for example.

A crucial aspect of fine-tuning NLP models or using RAG as an approach is the collection of high-quality, task-specific data to train on. Balaguer et al., (2024) emphasise the use of web scrapers as an effective technique for gathering such data. Web scrapers automate the extraction of relevant text and data from web pages, providing a rich and inclusive dataset that encompasses the variety of language used and contexts appropriate to the specific domain needed.

1.6 Retrieval Augmented Generation

The concept of Retrieval Augmented Generation represents a significant advancement in the field of natural language processing and artificial intelligence, particularly in how machines understand and generate human-like text. Miesle (2023) highlighted how RAG combines the capabilities of text generation models with information retrieval systems to enhance the generation of text that is not only contextually relevant but also rich in content and accuracy. For instance, in educational technologies, RAG can power systems that provide detailed explanations or answers to student inquiries based on course material. Similarly, in customer service, RAG-enabled chatbots can deliver accurate, detailed responses to complex queries by leveraging an extensive database of product information and support documentation.

At its core, RAG operates by retrieving information relevant to a query from a dataset through text embeddings and information retrieval algorithms. This information is then used as a supplementary input for the generation phase, where a language model, typically a transformer-based model like GPT or BERT (Bidirectional Encoder Representations from Transformers), produces text that incorporates the data from the retrieved documents (Jacky, 2023) through text embeddings and similarity scores. This methodology allows RAG models to produce responses that are not only coherent and contextually appropriate but also informative and precise, drawing directly from the source data.

1.6.1 Text Embeddings

Text embeddings convert textual data into high-dimensional vector spaces, allowing machines to comprehend the semantics and contexts of language. Techniques such as Word2Vec, GloVe, as well as newer contextual embeddings from models like BERT and GPT which leverage the self-attention mechanisms of transformers, as described in section 1.3.2 Self-Attention, decode the relationships among words in a sequence, as emphasised by Barnard (2024). The result is a weighted sum of these vectors, which encapsulates the contextual connections between words.

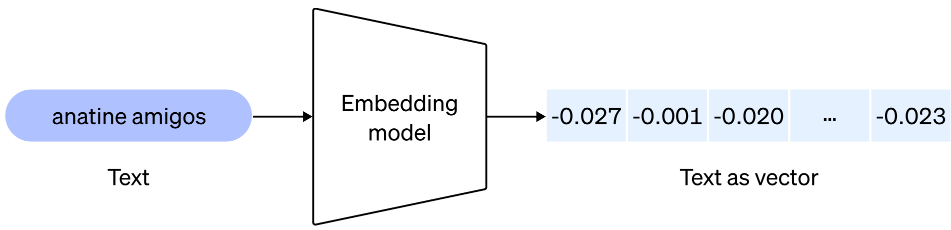

Figure 1.2 Word embedding flowchart (OpenAI, 2024)

In contrast to static models, these embeddings are dynamic and adjust based on the context of the words used. This method allows each word to related closely to the other words in the sequence, effectively capturing long-range dependencies within the text.

1.6.2 Information Retrieval

Information retrieval (IR) plays an essential role in linking the vector representations of text embeddings with their real-world use in applications such as question answering. The foundation for creating semantic connections between different texts lies in the cosine similarity measure. This allows IR algorithms to quantitatively assess the semantic similarity between two pieces of text, thus linking a user's query with a suitable answer source. Cosine similarity is mathematically determined by the cosine of the angle between two vectors. Considering the word embedding of a query as vector A, and the embedding of a potential answer source as vector B:

$$Similarity(\boldsymbol{A}, \boldsymbol{B}) = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{||\boldsymbol{A}|| \cdot ||\boldsymbol{B}||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sum_{i=1}^{n} A^2 \cdot \sum_{i=1}^{n} B^2}$$

where, A_i and B_i are the ith word embedding features of vectors A and B.

The resulting similarity score ranges from -1, suggesting exactly opposite, to 1 suggesting exactly the same.

1.7 Previous Approaches

1.7.1 Fine-tuning large neural language models for biomedical natural language processing

The research paper "Fine-Tuning Large Neural Language Models for Biomedical Natural Language Processing" by Robert Tinn et al. offers a comprehensive study on the intricacies of adapting large neural language models to the biomedical NLP domain.

It places emphasis on domain-specific pre-training and vocabulary as key to enhancing model robustness especially in the medical field. By pretraining models like PubMedBERT on biomedical corpora, the research achieves improvements across a range of NLP tasks. Exploring the balance between specialised and general vocabulary could inform more effective pretraining strategies, not just in biomedicine but across any specialised NLP domains.

The paper also delves into the importance of data collection and processing. The research underscores the importance of compiling a comprehensive and relevant biomedical corpus that includes a wide range of medical literature, clinical reports, patient records, and other domain-specific texts. The meticulous process of data collection ensures that the model has access to a rich repository of biomedical knowledge, which is essential for the effective pre-training and fine-tuning.

Furthermore, the paper discusses the sophisticated methods employed for processing this biomedical data, emphasising the need for cleaning, normalisation, and structuring of the datasets to make them suitable for training purposes. By highlighting the critical role of data collection and processing, the study reinforces the notion that the foundation of any successful NLP application lies in the careful and methodical preparation of its underlying data.

Lastly, it highlights a systematic exploration of fine-tuning techniques and addresses a critical challenge in applying large models to low-resource datasets. The paper identifies that in those larger models, while potentially more powerful, pose greater challenges for fine-tuning stability. This insight prompts further questions about the optimal model size for specific biomedical NLP tasks and whether there exists a "sweet spot" where the benefits of increased model capacity outweigh the stabilisation challenges. Future research could focus on developing guidelines for selecting model sizes based on the characteristics of specific NLP tasks and datasets.

1.7.2 RAG vs Fine-tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture

This paper titled "RAG VS FINE-TUNING: Pipelines Tradeoffs and a Case Study on Agriculture" by Aman Gupta et al. presents an in-depth analysis of the advantages and challenges associated with the use of Retrieval-Augmented Generation and fine-tuning techniques in the development of Large Language Models for domain-specific applications, with a focus on agriculture.

The paper effectively contrasts the RAG and fine-tuning approaches, highlighting their respective strengths and limitations. RAG, which augments the model's prompt with external data, is shown to be beneficial for its ability to incorporate vast amounts of domain-specific information without permanently altering the model's parameters. This approach is particularly valuable in fields like agriculture, where real-time data and geographic-specific knowledge can significantly enhance the model's responses. However, the study also notes RAG's dependency on the quality and relevance of the retrieved information, which can sometimes lead to inaccuracies or irrelevant content being introduced into the model's outputs.

Fine-tuning, on the other hand, involves directly incorporating additional knowledge into the model itself by adjusting its parameters based on a specific dataset. The paper underscores fine-tuning's strength in significantly improving model accuracy and its ability to tailor the model more closely to domain-specific needs. Nonetheless, the challenges associated with fine-tuning, such as the potential for model overfitting and the high computational costs involved, are highlighted and accentuated. The study's findings demonstrate an accuracy increase of over 6% through fine-tuning, further enhanced by combining it with RAG, which adds another 5% to accuracy. This cumulative effect showcases the potential between these two approaches.

While the paper provides compelling evidence of the benefits of RAG and fine-tuning, it also opens possibility for further research. One area for exploration is the development of more efficient fine-tuning techniques that minimise the risks of overfitting and reduce computational demands. Additionally, the paper's focus on agriculture as a case study is highly insightful but also calls for research into the feasibility of applying these methodologies across different knowledge domains.

Chapter 2
Methodology

In the previous chapter, relevant technologies as well as their use cases were identified. Based on the research gathered, the current chapter aims to highlight the methodology pipeline (Figure 2.0) that was used to prepare, process, and use the data to create the chatbot.

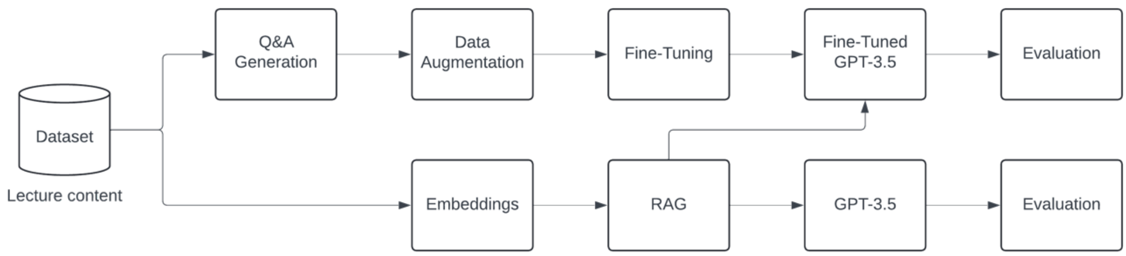

Figure 2.0 Methodology pipeline: Data is collected. Content and structure of the documents is extracted. The data is fed to Q&A generation. The generated Q&A pairs are used to fine-tune the model. Model is evaluated with and without RAG under different metrics.

2.1 Version control

To manage the project, Git was used to keep track of changes made. The code repository was hosted on GitHub and Figure 2.1 below shows a sample of the git commit history while working on the project. Each commit is labelled with a message defining what changes were made.

Following each commit, the project was tested to ensure it still ran as expected. An advantage of using Git for this project is the ability to "rollback" to previous changes. Using the commit ID, if the project fails to run as expected after a change, the old version can easily be restored.

```
* 1de9df8 added web scraped files
* 2fa05c4 fixed file name issue
* ba90e1f implemented downloads
* 7505aad correctly fetches unit urls
* ff61215 added GPT tune
* 923c5f8 added back translation and synonym replacement
```

Figure 2.1 Sample Git commit history

2.2 Data Collection

The focus of this chatbot is to provide accurate lecture specific answers to all queries. To achieve this, it is essential to compile a large and well-organised dataset. The COMP2121 Data Mining course was selected based on previous personal experience with the module. It is known that the lecturer consistently publishes transcripts and lecture PowerPoints online, all of which are available on the Minerva Blackboard module page. This accessibility of materials makes it an ideal choice.

To collect the material, a web scraper is developed using Selenium (selenium.dev), Chrome webDriver (chromedriver.org), and Python to fetch all files needed automatically. The scraper identified files with either "lecture" or "transcript" in the title to filter out other material posted by the module leader such as further readings or videos. Ultimately, the scraper retrieved and downloaded all lecture slides and transcripts as both pptx and pdf files respectively.

2.3 Data Extraction

Correctly processing the data is crucial. By automating the data extraction process, this methodology can minimise the time and effort traditionally required to manually extract text from PowerPoint and PDF files.

2.3.1 PDFs

Using the *PyMuPDF* library, text is extracted from PDF documents. Given the complex nature of PDF formatting, which can range from simple text documents to files containing images, tables, and various graphical elements, the python library effectively handles the text extraction process. By iterating through each page of the PDF document, it captures the textual content and consolidates it into a singular text file.

2.3.2 PPTX

Similarly for the collected PowerPoint slides, the *python-pptx* library is used to access and navigate the structure of PowerPoint files. This process involves iterating through each slide within a presentation and systematically extracting the textual content embedded within. The extracted text is also saved to a text file corresponding to the original presentation file, thus converting the visual and structured content of PowerPoint files into the accessible and manipulable text format needed.

2.4 Prompt Generation

Leveraging the files from the previous step, the aim is now to generate contextually accurate and high-quality question and answer prompts. These Q&A prompts are essential for fine-tuning the chatbot as well as being used to create a dataset for evaluation. OpenAI provides a specific *.jsonl* format for fine-tuning models which requires the prompts and responses to be structured for supervised model training. Adhering to OpenAI's formatting guidelines is crucial for the successful fine-tuning of the model. Figure 2.2 below shows an example pair:

```
{"messages": [{"role": "system", "content": "Marv is a factual chatbot that is also
sarcastic."}, {"role": "user", "content": "What's the capital of France?"}, {"role":
"assistant", "content": "Paris, as if everyone doesn't know that already."}]}
```

Figure 2.2 Example Question/Answer pair format
(OpenAI, 2023b)

2.4.1 Text Cleaning

When extracting text from PowerPoint (PPTX) and PDF files, the output may often include a variety of non-standard characters and irregular spacing. This can occur from formatting outlier data types within the file such as special characters for bullet points, annotations, or encoding differences that are not consistent with standard ASCII text. Additionally, text extraction processes may inadvertently introduce multiple consecutive whitespace characters, including spaces, tabs, and newline characters, which can distort the natural flow of text and complicate further text processing.

To address this, two critical cleaning operations are performed on the text. First removing non-ASCII characters, which are frequently encountered in text extracted from PPTX and PDF files due to their support for a wide range of fonts and special characters beyond the basic ASCII set. By filtering out these non-ASCII characters, the text is restricted to a standard character set, enhancing compatibility, and reducing the likelihood of issues down the line.

Secondly, normalising the excess whitespace within the text. This step is particularly important because text extracted from PPTX and PDF files often contains inconsistent spacing—such as multiple spaces between words or extraneous spaces at the beginning or end of lines—reflecting the original document's layout rather than the logical structure of the text. By collapsing multiple whitespace characters into a single space and trimming leading and trailing whitespace, the final dataset is cleaner and more uniform in structure.

2.4.2 Text Splitting

The process continues with the extraction of text from the documents through segmenting the text into manageable chunks of 100 tokens. The size was chosen through trial and error to ensure that each piece of text is optimal for maintaining both contextual integrity and relevance without missing key concepts or generating redundant or repetitive questions.

2.4.3 Q&A Generation Chain

Once the text is segmented, LangChain (langchain.com) is used to integrate OpenAI's capabilities to generate Q&A prompts from the segmented text. This integration is pivotal, as it leverages OpenAI's already trained advanced language model to understand the content's nuances and context to generate relevant questions and answers. The generated Q&A pairs are directly reflective of the module content, ensuring that the fine-tuning process is grounded in the actual data extracted from the documents. GPT-3.5 is prompted with the following text:

````
"""You are a smart assistant designed to help professors examine university courses, come up with questions to test the sample content given.

You are given a segment of a lecture, you must come up with a question and answer pair that can be used to test a student's understanding of it.

All text given is a part of lecture material, so address it as such.

When coming up with this question/answer pair, you must respond in the following format:
```
{{
    "question": "$YOUR_QUESTION_HERE",
    "answer": "$THE_ANSWER_HERE"
}}
```
Everything between the ``` must be valid json.

Please come up with a question/answer pair, in the specified JSON format, for the following text:
````

Figure 2.3 GPT Question/Answer prompt

The data is requested and saved in JSON. Each Q&A pair is encapsulated as an individual JSON object, which contains a prompt and its corresponding response.

This structure allows for future data manipulation with ease and aligns with OpenAI's format requirements for fine-tuning datasets. A sample can be seen in Figure. 2.4 below:

```
{
    "question": "What will Professor Eric Atwell be talking about in the lecture?",
    "answer": "Professor Eric Atwell will be talking about n-gram language modeling."
},
```

Figure 2.4 Returned Question/Answer pair

2.5 Implementing Retrieval Augmented Generation

Incorporating external information through RAG allows the chatbot to produce more accurate and contextually relevant outputs. Through embeddings and query creation, each input query can be packaged with only relevant information from the module before it is passed on to the GPT.

2.5.1 Embeddings

Embeddings are fundamental to the RAG framework, serving as a bridge between raw text and the machine learning models that process it. The lecture embeddings are generated for set text chunks using the "`text-embedding-3-small`" model from OpenAI. This process transforms each text chunk into a high-dimensional vector that captures semantic and syntactic information.

The generation of embeddings starts with the previously split text chunks and stored along with the lecture titles. These chunks are processed in batches to efficiently generate embeddings which are stored for later retrieval in a *.csv* file (Figure 2.5) along with the respective lecture names, fetched from file names. The embeddings effectively represent the semantic relation of the text in a form that can be quantitatively analysed.

```
      lecture title            text                                          embedding
0     5.2 University ...  OCOM5204M – Data mining and ...  [-0.021362967789173126, -0.016367396339774132,...
1     5.2 University ...  is much more complicated. You ...  [-0.028354814276099205, -0.00609942339360714, ...
2     5.2 University ...  one exact particular example. ...  [0.001691731158643961, 0.043431151658296585, 0...
...          ...                              ...                                              ...
1119     lecture 3.2       Villanueva] of [ORG United] ...  [-0.0433991365134716, -0.0064753261394798756, ...
1120     lecture 3.2       Entropy Markov Models (MEMM)...  [-0.018054673448204994, 0.029924873262643814, ..
```

Figure 2.5 embeddings.csv sample

2.5.2 Query Creation

The query creation starts with the generation of a unique embedding for the user's query. This is achieved by passing the query through the same embedding model used to encode the module data chunks, ensuring consistency in the representation of textual information. This embedding is the vector that encapsulates the semantic essence of the user's question.

Using both embeddings, the semantic similarity between the query embedding and each of the embeddings associated with the stored text chunks is calculated using the cosine similarity metric. After assessing the cosine of the angle between two vectors in a multi-dimensional space, a higher cosine similarity value indicates a greater degree of similarity between the query and a given text chunk, suggesting relevance to the user's inquiry.

Sorting the returned chunks in descending order of similarity, the algorithm can prioritise text that is most likely to contain information relevant to answering the query to incorporate with the prompt.

The incorporation of relevant text into the query message is executed with careful consideration of a predefined token budget. This budget represents the maximum allowable length of the input to the GPT model, ensuring that the query message remains within the model's defined capabilities.

Finally, the prompt encapsulates selected text chunks in a structured format, clearly indicating them as sources of context for the GPT model along with the lecture titles. This structured message, which also includes the original query, is then passed to the generative model. A sample of the final prompt is shown below.

```
Use the below lecture content to answer the question. Ensure that you mention the lecture title in your
answer. Never give a response without mentioning which lecture it has come from. It is critical that you
cite your source lecture title. If the answer cannot be found in the contents, write 'I could not find an
answer :(' do not use your own knowledge.


Lecture title: 1–1 Background practical applications Transcript

Lecture content:

"""

data mining? So if you've done the machine learning course, or maybe you haven't yet but you will do
sometime soon, this is mainly about learning the algorithms for doing various different sorts of machine
learning, where you input some data and some classes if it's a classifier. And then you have various
different ways of predicting the classes of unseen data. And you want to aim to have optimal accuracy, or
to measure how good it is, how many of it you got right in your test set and how many are wrong. ©
University of Leeds 4 of 10 For data mining, it's also machine learning, but it's applied and there's much
more of a focus on using the machine learning as a
"""


Lecture title: 1–1 Background practical applications Transcript

Lecture content:

"""
```

Figure 2.6 Sample prompt for the query: “What is Data Mining about?

2.6 Data Augmentation

A critical step in enhancing the diversity and volume of training data available for machine learning models, particularly in the domain of natural language processing, is data augmentation. This process involves artificially increasing the dataset's size by generating new data points from existing ones, providing the model with a higher number of diverse Q&A examples to learn from. This project uses two main strategies for augmenting the data: back-translation and synonym replacement.

2.6.1 Back Translation

Back-translation is a key technique for enhancing linguistic models by introducing syntactic diversity. This approach involves translating a sentence into an intermediary language before reverting it to its original form: maintaining its semantic integrity but modifying its structure. Feng et al. (2021) highlight that this technique is employed with the objective of utilising the augmented data as a regularising factor, thereby mitigating the effects of overfitting during the machine learning models' training phase.

The process begins by randomly selecting an intermediate language from a specified list with the goal of including syntactic variability and removing any bias in the outcomes. According to Ciolino et al. (2022), the three languages that resulted in the most significant movements in various NLP metrics when used for back-translation were Tatar, Danish, and Malayalam. Based on these findings, these languages were chosen. The sentence is translated into an intermediate language utilising the Google Translator API. As the text undergoes back-translation into its original language, it retains its semantic context but adopts new structural variations. Below is a sample to illustrate the impact of back-translation on sentence syntax.

Original language	Building the background knowledge for a chatbot that mimics a real teacher is harder because real teachers do more than just answer questions; they also provide additional context and engage in more complex interactions.
Intermediary language(Danish)	Det er sværere at indbygge baggrundsviden til en chatbot, der efterligner en rigtig lærer, fordi rigtige lærere gør mere end blot at svare på spørgsmål; de giver også yderligere kontekst og engagerer sig i mere komplekse interaktioner.
Back-translated	It's harder to build background knowledge into a chatbot that mimics a real teacher because real teachers do more than just answer questions; they also provide additional context and engage in more complex interactions.

Figure 2.7 Sample back-translation set

2.6.2 Synonym Replacement

The synonym replacement strategy revolves around selecting specific words within a sentence and swapping them with their synonyms. Despite the change in word choice, the essence of the sentence remains intact. Utilising the WordNet database via the Natural Language Toolkit (NLTK) for this purpose ensures that the replacements are both contextually and semantically fitting.

Employing synonym replacement brings forth several advantages similar to back translation. It introduces a degree of linguistic diversity to the training dataset. This enrichment with varied expressions of the same notion aims to enable the model to comprehend and respond to a wider array of user inputs, thereby enhancing its robustness and adaptability.

Initially, the process involves identifying suitable synonyms for a given token, with consideration of the token's part-of-speech tag to ensure functional interchangeability within the sentence context. This is facilitated by converting spaCy's POS tags to WordNet's format, enabling the synonym retrieval from WordNet. Subsequently, the algorithm can identify tokens within the sentence eligible for replacement—those that are not stop words or proper nouns and those that have a corresponding POS tag in WordNet. A subset of these words is randomly selected for synonym substitution.

When crafting the new sentence, for each chosen token, a synonym is selected from the set of available synonyms and used to replace the original word in the sentence. This process results in the construction of a new sentence where each selected word has been replaced with a synonym. Given that synonym replacement might inadvertently lead to grammatical discrepancies, the final step involves applying a grammar correction tool to identify and fix any grammatical errors in the new sentence. This ensures the output is not only semantically consistent but also grammatically sound and can be seen in the table below:

Original sentence	What is a limitation that university researchers face when using Sketch Engine?
Synonym-Replacement	What is a restriction that university research workers face when using Sketch Engine ?

Figure 2.8 Sample Synonym-Replacement set

2.7 Fine-Tuning

2.7.1 Model Training

The fine-tuning job is established through the OpenAI API that passes the dataset, the base model (gpt-3.5-turbo) type, and hyperparameters such as epochs, learning rate, and batch size.

Through trial and error, the ideal hyperparameters were identified and the final model was tuned on a dataset of 4446 lines, on 8 epochs, with a learning rate multiplier of 2 and a batch size of 23.

2.7.2 Model Testing

Testing involves periodically assessing the model's performance while tuning using a validation dataset to ensure effective learning without overfitting. The validation set comprises 20% of the entire dataset, with the remaining 80% divided between training (60%) and testing (20%). The training process is monitored using training and validation loss metrics. OpenAI use the cross entropy loss function to measure how well the model's predictions align with the actual data.

The cross-entropy loss function allows the optimisation of the weights of a machine learning model throughout training. The goal is to reduce the discrepancy between the actual and predicted outcomes. Consequently, a value approaching 0 indicates a well-performing model, while a value closer to 1 suggests a model that performs poorly.

These training metrics, along with the final evaluation, help guide adjustments to hyperparameters or training data to optimise learning.

2.8 Evaluation

The evaluation of NLP models is a critical process demanding a comprehensive approach to accurately assess their performance across a range of tasks and outputs. Given the nuanced nature of language and the complexity of tasks, it is essential to utilise a diverse set of evaluation metrics.

In preparation for a detailed assessment, five distinct metrics have been selected, each chosen for its unique ability to highlight a specific aspect of model performance. Following the quantitative analysis provided by these metrics, a user evaluation is incorporated to add a qualitative aspect to the evaluation process.

This diversified evaluation framework aims to provide a holistic understanding of an NLP model's capabilities, ensuring that the assessment is not only comprehensive but also aligned with the subtilise of language processing tasks.

2.8.1 Metrics

Since evaluating NLP models often requires a multifaceted approach, five metrics have been chosen to quantify different aspects of model performance. They are ROUGE, METEOR, BERTScore, F1 score, and semantic similarity scores. Each of these metrics highlights different aspects of model performance and will be used to give a wholistic evaluation.

2.8.1.1 ROUGE Score

The ROUGE Score (Recall-Oriented Understudy for Gisting Evaluation) is a suite of metrics used predominantly in evaluating automatic text summarisation. It calculates the quality of output by measuring the overlap between the generated text and a set of reference text. The metric focuses on the overlap of n-grams, word sequences, and the longest common subsequence (LCS), (Lin, 2004). LCS is particularly noted for its effectiveness in capturing sentence-level structural similarities, which plays a crucial role in assessing the coherence and cohesion of the generated texts. Despite this, ROUGE has limitations, including a potential bias towards longer sentences due to its emphasis on recall, and it may not adequately capture all semantic nuances between the generated and reference texts.

2.8.1.2 METEOR Score

The METEOR Score (Metric for Evaluation of Translation with Explicit ORdering) is a metric designed for evaluating machine translation quality by comparing generated translations with one or more reference translations. It seeks to address some of the shortcomings of earlier metrics like BLEU by incorporating synonymy and stemming into its evaluation process, thereby providing a deeper understanding of semantic accuracy (Banerjee and Lavie, 2005). The METEOR Score calculates alignment between the generated and reference texts, factoring in precision and recall and applying a harmonic mean to balance the two. It includes penalties for mismatches such as word order and grammatical errors, making it highly sensitive to small inaccuracies in large texts.

2.8.1.3 BERTScore

BERTScore leverages the deep contextual embeddings generated by models such as BERT (Bidirectional Encoder Representations from Transformers) to evaluate the semantic similarity between generated and reference texts (Zhang et al., 2020). By using contextual embeddings, BERTScore can accurately assess the semantic coherence and relevance of the text, making it highly aligned with human judgments. However, the computational intensity of generating embeddings and the dependency on the specific pre-trained BERT model selected are notable limitations. The performance of BERTScore can significantly vary based on the characteristics of the underlying model.

2.8.1.4 F1 Score

The F1 Score is a widely used metric in natural language processing for classification tasks, offering a harmonised measure that combines precision and recall. By calculating the harmonic mean of precision and recall, the F1 Score provides a single metric that balances these two aspects, making it particularly valuable for evaluating the performance of models in identifying relevant information (Varoquaux et al., 2015). However, the F1 Score may not fully encapsulate the complex nuances of linguistic tasks that extend beyond the realm of binary or multiclass classification.

2.8.1.5 Semantic Similarity Score

The Semantic Similarity Score assesses the degree of semantic equivalence between two text snippets by employing advanced embedding techniques to capture the underlying semantic information of the texts (Hugging Face, n.d.). This metric is crucial for applications requiring a deep understanding of textual content, such as document summarisation, information retrieval, and question-answering systems. By leveraging semantic embeddings, it can align closely with human perceptions of textual similarity, particularly for distinct comparisons. However, challenges arise in dealing with syntactic variations and there is also an inherent difficulty in establishing a universal benchmark for measuring "similarity," alongside the dependency on the quality of the utilised embeddings for accurate evaluations.

2.8.2 User Evaluation

Human evaluation is ultimately crucial for validating the relevance and coherence of generated texts. Although metrics such as BERTScore and Semantic Similarity Scores provide insights into specific attributes of text quality, gathering user feedback delivers a holistic assessment that captures both the quantitative metrics and the qualitative dimensions of the generated text.

Given the technical nature of the inquiries and the tendency of generative models like GPT to fabricate details, it's crucial that the participants possess a foundation in the topic to accurately assess the responses' validity. The methodology for user evaluation is designed with a focus on gathering the responses of individuals who have an academic background relevant to the subject matter, particularly students who have previously completed the module COMP2121 Data Mining. This study involves presenting these informed participants with ten pre-set chosen questions, categorised across three levels of difficulty to comprehensively assess the models' performance across a broad range of possible scenarios:

- Easy difficulty general questions about data mining.
    - Ex. *Who founded Sketch Engine?*

- Medium difficulty questions that are more course specific content but may be found outside the lecture content.
    - Ex. *What is data wrangling in the context of data mining?*

- Challenging Questions that are highly lecture specific.
    - Ex. *In what lecture does Eric Atwell talk about multi-word expressions?*

The questions are displayed in sequence, accompanied by responses from three distinct models:

- Fine-Tuned GPT-3.5-Turbo with RAG
- Unmodified GPT-3.5-Turbo with RAG
- Unmodified GPT-3.5-Turbo

In an effort to reduce bias, the origin of each answer is not disclosed to the participants and they are randomly shuffled for each question. After reviewing the questions and answers, participants are asked to select the response they believe best addresses the query. All participant responses are collected anonymously, ensuring the integrity and impartiality of the feedback.

Chapter 3
Results

The results of the fine-tuning process, as detailed in Section 2.7, are presented below through a series of graphs (Figure 3.0 and 3.1) that visualise the performance metrics of the fine-tuned model. The final metrics show a training loss of 0.753, a validation loss of 1.687 with an accuracy rate of 0.792.

Figure 3.0 Training and validation loss over 1600 steps.

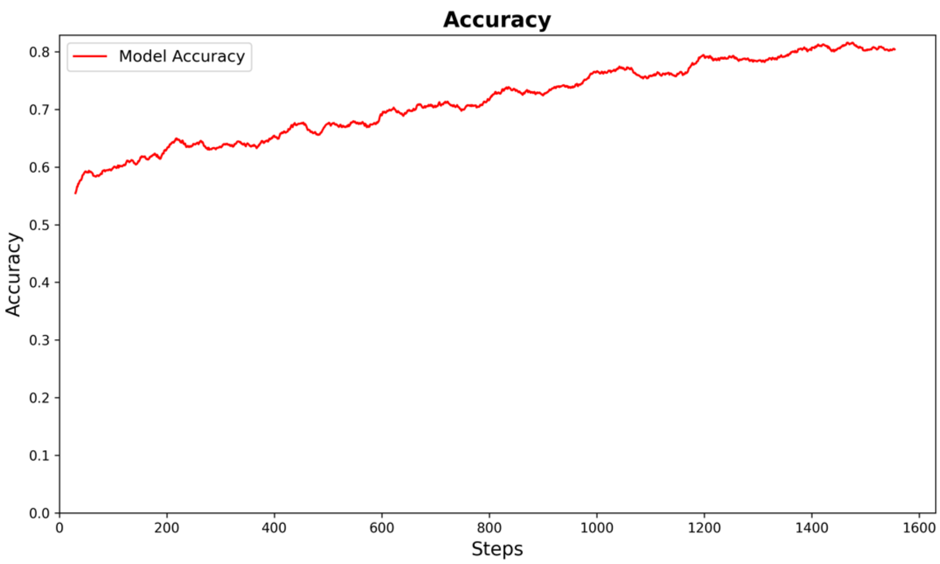

Figure 3.1 Training accuracy over 1600 steps

The results from the evaluation as described in section 2.8.1 are reported in table 1.0 with the highest scores for each metric emphasised in bold. The fine-tuned GPT-3.5-Turbo with RAG model achieved the best performance for all metrics. The second best scores on all metrics were achieved by the GPT-3.5-Turbo with RAG, which leaves the base unmodified GPT-3.5-Turbo having the lowest scores. The difference between best and worst scores were 0.576 for ROUGE, 0.246 for METEOR, 0.072 for BERTScore, 0.576 for F1 and 0.183 for Semantic similarity.

	ROUGE Score	*METEOR Score*	*BERTScore*	*F1 Score*	*Semantic Similarity*
GPT-3.5-Turbo	0.1811	0.4260	0.8882	0.2490	0.7897
GPT-3.5-Turbo with RAG	0.5890	0.5138	0.9294	0.5965	0.8361
Fine-Tuned GPT-3.5-Turbo with RAG	**0.7540**	**0.6724**	**0.9606**	**0.8246**	**0.9724**

Table 1.0 Evaluation metrics. The highest score for each metric has been marked in bold

Lastly, the results from the user evaluation are illustrated below in Figure 3.2 and 3.3. Figure 3.2 indicates a varied response for the general and easy questions with the base GPT-3.5 model performing best. As the category of questions changed, users showed a strong preference to the RAG augmented model over the other models.

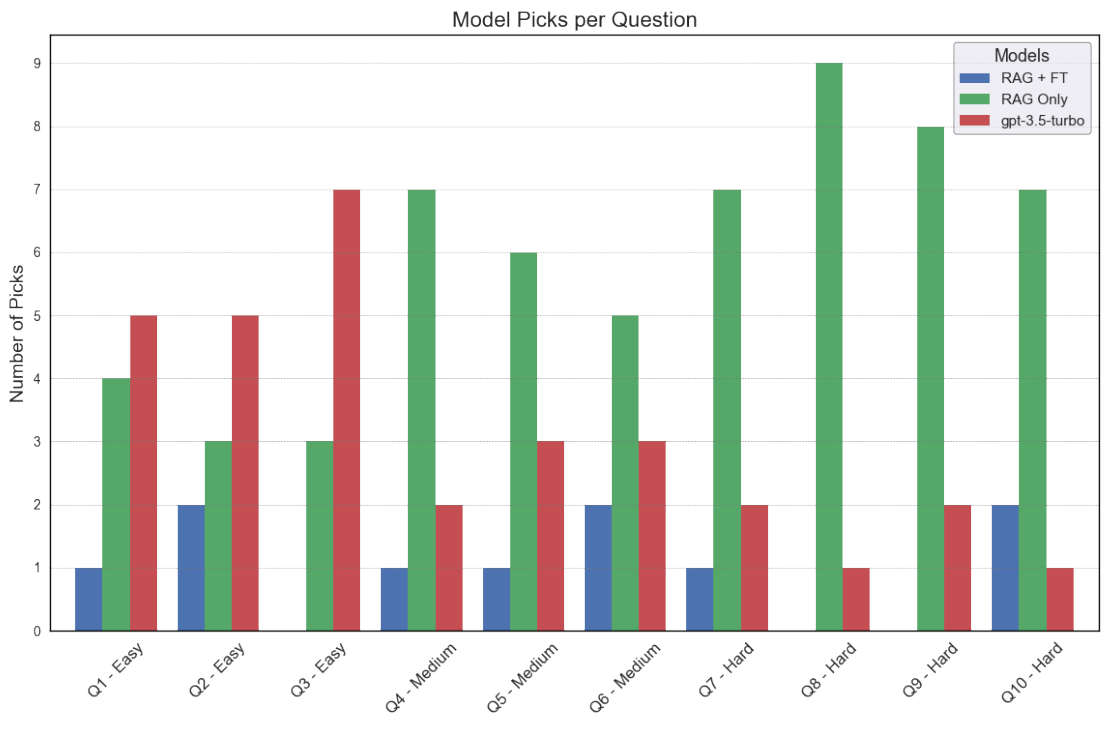

Figure 3.2 User Evaluation: Model picks per Question

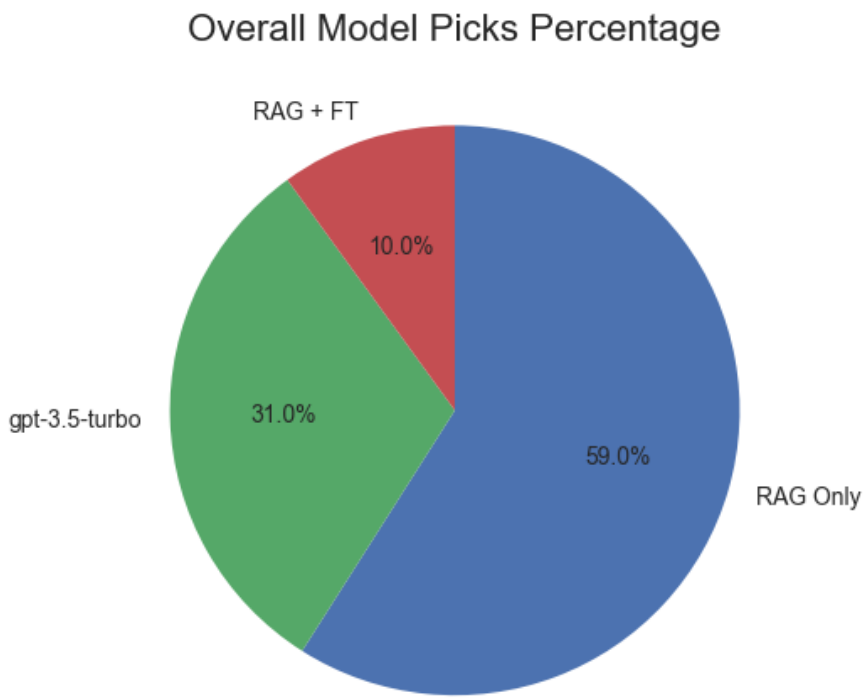

Figure 3.3 User Evaluation: Overall Model Picks

Reviewing the overall percentage results, it's evident that the RAG only model has outperformed the others, with 59% of users favouring its responses across all questions. Conversely, the fine-tuned model showed the least preference, receiving only 10% of picks. This leaves the standard, unmodified GPT-3.5 model with the remaining 31%.

Chapter 4
Discussion

4.1 Performance Evaluation and Analysis

4.1.1 Fine-Tuning Discussion

As reported in the previous section (Chapter 3, Results) the training loss for the final fine-tuned model was 0.7530 with the validation loss significantly higher at 1.6872. This observation indicates a performance gap where the model performs well on the training data but much worse on unseen validation data.

The primary factor contributing to this issue is overfitting. The model learns the details and noise within the training data to such an extent that it begins to negatively impact its performance on new data. This is evidenced by a lower training loss compared to the validation loss, suggesting that the model has memorised some aspects of the training data which do not generalise to the validation set. Although this was anticipated, the main challenge lies in preventing overfitting from escalating to a degree that significantly distorts outcomes and diminishes the model's effectiveness on new data that GPT-3.5 inherently lacks prior knowledge of. Importantly, data augmentation proved to be a significant aid in this regard.

The tendency for overfitting can be attributed to the highly specialised nature of the training data which consisted solely of university module data and offered an arguably limited general guidance on how the model should respond to varied styles of questions, albeit enhanced through data augmentation. This issue is further exacerbated by the complexity of the GPT-3.5 model itself. Despite its vast array of parameters, it does not inherently possess knowledge of the specialised content from the Data Mining module, leading the model to result in memorising the information rather than learning to generalise from it.

To address the potential minimal guidance on how to answer questions and reduce overfitting, it may be argued that using a larger and more augmented dataset would be beneficial to expose the model to a broader and more generalised dataset. However, trials with augmentation have shown that tuning the model on an even larger and more augmented dataset was counterproductive and actually promoted higher overfitting, even with a lower learning rate and tuned hyperparameters. This underscores that both the quantity and the quality of the dataset are crucial in the training process, where a carefully curated as well as high-quality dataset is vital for achieving robust model generalisation.

Ultimately, the process helped to find the best balance between fitting the training data and generalising to new data, as evidenced by a gradual reduction in validation loss and enhanced stability in the model's predictive accuracy. The iterative process of tuning and testing underscores the nuanced nature of machine learning model development, where even minor modifications can lead to significant enhancements in model robustness and accuracy.

4.1.2 Evaluation Metrics Discussion

The evaluation of the various implementations of the project, as shown in the table 1.0, present a clear progression in performance across all metrics after fine-tuning and RAG. The fine-tuned GPT-3.5-Turbo with RAG model achieved the highest scores for all the metrics evaluated.

The ROUGE score improvements from 0.1811 in the base GPT-3.5-Turbo to 0.7540 in the fine-tuned version with RAG is substantial, indicating a notable enhancement in capturing the important points of the module data reference texts. This suggests that the fine-tuning process has significantly improved the model's ability to generate summaries that are even closer to a human-generated reference than the base GPT-3.5 model.

The METEOR score has seen an improvement, rising from 0.4260 to 0.6724 with fine-tuning. This reflects an enhanced alignment with human judgments concerning the translation quality, which in the context of the model's use-case could be related to the quality of the generated text in terms of precision and recall.

A small rise in BERTScore from 0.8882 to 0.9606 in the fine-tuned model with RAG indicates a better contextual alignment with the reference texts. The initial score of 0.8882, which is already high, reflects the base GPT-3.5-Turbo model's strong foundational understanding and processing capabilities due to its pre-training but an improved score confirms that fine-tuning has made the model's outputs more adapted to the subtleties in this context, likely by internalising patterns and relationships inherent in the dataset.

There is a noteworthy increase in the F1 score, from 0.2490 to 0.8246, indicates that there is a significant greater number of accurate responses with relevant information. Rather than simply producing a high volume of content, the model is now much more effective at narrowing down its responses to include what is truly essential. This score indicates that the model's fine-tuning has pushed it toward an optimal threshold where it is neither overly conservative nor excessively liberal in its information retrieval—it is not missing out on crucial points (thus avoiding false negatives), nor is it including too much irrelevant detail (thereby avoiding false positives). However, achieving a perfect F1 score is challenged by the inherent complexity of accurately balancing recall and precision, especially in nuanced or ambiguous contexts such as a highly technical domain such as this university module. For users, such an improvement reflects responses that are both accurate and concise, enhancing both the efficiency and utility of the model in practical applications, such as summarising lecture content or generating precise answers to complex questions.

The Semantic Similarity increasing from 0.7897 to 0.9724 indicates that the fine-tuned model generates responses that are more contextually appropriate and meaningful to the lecture data. With this level of semantic comprehension, the model is able to grasp the core concepts and ideas within the domain-specific data more effectively. Furthermore, a Semantic Similarity score nearing 0.9724 implies that the model's outputs align closely with the language of the lectures and transcripts from the module; providing responses that feel intuitive and relevant to users.

The second-best scores across all metrics were achieved by the GPT-3.5-Turbo with RAG, which indicates that the RAG component itself provides the majority significant improvement over the base model. However, the additional fine-tuning process appears to optimise it even further. This suggests that while RAG effectively enhances the base capabilities of the model by incorporating a broader knowledge domain and more detailed information retrieval, the fine-tuning process tailors these enhancements specifically and maximises performance further.

4.1.3 User Evaluation

The user evaluation data revealed a discrepancy between the expected performance from the evaluation metrics and the actual user preferences. Notably, as shown in figure 3.2, the RAG Only model was the most popular choice, gathering significant picks across all question difficulties. This preference highlights the users' appreciation for the RAG's enhanced information retrieval capabilities, especially in responding to complex queries that demanded insight directly from lecture content. For specific examples, see Appendix F.

Conversely, the RAG + FT model, despite its fine-tuning, received a remarkably lower selection rate. It was somewhat preferred for simpler questions but not for more challenging ones. This suggests that the fine-tuning, intended to specialise the model's answering, has not aligned with user expectations in practical scenarios. The user evaluations reveal a trend where the unmodified GPT model, along with its base capabilities, is preferred over the more context-trained fine-tuned variant. In some cases, users even favoured incorrect responses from the base GPT model for "hard" questions over contextually accurate answers from the fine-tuned model. This suggests that the unmodified model's style of response or perhaps its broader responses, although at times incorrect, are more favoured by users than the precision and concise answers offered by the fine-tuned version. It's a clear indication that user preferences can sometimes be counterintuitive, valuing certain qualitative aspects of the responses that may not be captured by fine-tuning.

In easier contexts, where a high knowledge of lecture material is less critical, the base GPT-3.5-Turbo model showed steady user preference. This reaffirms the inherent capabilities of the GPT-3.5 model to provide satisfactory answers without the need for additional fine-tuning or improvements on general questions.

These results are reiterated in figure 3.3. A majority (59%) preferred the RAG Only model, followed by 31% opting for the base GPT-3.5-Turbo, and 10% for the fine-tuned variant. The preference for the RAG model can be attributed to its capacity to pull information directly from lecture materials, which is particularly advantageous for generating responses that are both extensive and encompassing; unlike the fine-tuned model that, despite its tailored training, falls short of meeting user expectations in this regard by providing answers that are too concise although correct.

4.2 Conclusion

The extensive evaluation and analysis of the enhancements made to the GPT-3.5 model have provided vital insights into its performance, revealing both strengths and limitations.

From the performance analysis in section 4.1.1, it is evident that while the fine-tuned model shows excellent capability on controlled metrics and demonstrates reduced training and validation losses, although it also suffers from overfitting. This misalignment between model performance on training data versus unseen validation data underscores the critical challenge of overfitting in highly specialised datasets. Despite efforts to enhance data quality and diversity through augmentation, the results indicate a pressing need to explore more sophisticated methods to prevent overfitting and improve generalisation.

The evaluation metrics discussed in section 4.1.2 highlight significant improvements across a range of performance indicators such as ROUGE, METEOR, BERTScore, F1, and Semantic Similarity scores between GPT-3.5 and the enhanced version. These metrics reflect substantial enhancements in the model's ability to generate relevant and contextually appropriate responses. Notably, these improvements suggest that the model has the potential to achieve close alignment with the human-created reference texts and to internalise patterns within the lecture-specific data effectively.

However, as revealed in the user evaluation section 4.1.3, there is a discrepancy between the model's quantitative metric improvements and user satisfaction. While the evaluation metrics suggest high performance, the practical user experience indicates a preference for the RAG Only model over the fine-tuned version, especially for complex domain specific queries. This preference could be attributed to the fine-tuned model's responses, which, while accurate, lack the breadth and flexibility users value in real-world applications. The users' tendency to favour broader, albeit less precise responses from the base model or RAG only configurations, signals a need for a balance between precision and the ability to generate expansive and engaging content.

In conclusion, the project successfully met its aims of integrating RAG and fine tuning to enhance GPT-3.5 in educational contexts. This is shown through the quantitively evaluated metrics as well as the qualitative user evaluation. This project demonstrated how new AI technology has the potential to substantially improve educational tools, offering institutions the capability to enrich the learning experience and address a broad spectrum of educational demands, thereby bridging gaps in knowledge and accessibility.

4.3 Ideas for future work

For future ideas and improvements, investigating more sophisticated fine-tuning technique such as differential learning rates, where various layers of the neural network are adjusted at different speeds, could improve the model's capability to adapt to educational material while preventing overfitting. This approach might include applying lower learning rates to initial layers to preserve broad knowledge, while more intensively tuning the upper layers to better align with specific educational scenarios.

Additionally, the integration of vector databases can manage and retrieve text embeddings more efficiently in a RAG framework, potentially reducing retrieval times but more importantly, increasing the scalability of systems designed for real-time educational assistance. Exploring vector quantisation techniques could allow for highly efficient systems at a high scale thus allowing for multiple university modules to be considered at a time.

Based on the insights from user evaluations, it is important that future ideas focus on incorporating user feedback into the fine-tuning process of potential models. As demonstrated, fine-tuning has the potential to enhance a model's metrics significantly; however, in this instance, there was a noticeable misalignment between the improved metrics and actual user preferences. Therefore, aligning the fine-tuning strategy more closely with user expectations will be essential to maximise both the effectiveness and user satisfaction. This approach will ensure that enhancements in technical performance translate into real-world usability across all types of questions.

Furthermore, developing robust API nodes and a more modular integration could significantly streamline the implementation of this framework across different online educational platforms, massive open online courses, and learning management systems. This approach not only encourages scalability but also promotes the adoption of AI-driven educational technologies by enabling easier integration with existing university learning systems.

List of References

Alammar, J. 2018. The Illustrated Transformer. *Github.io*. [Online]. Available from: http://jalammar.github.io/illustrated-transformer/.

Balaguer, A., Benara, V., Cunha, R.L. de F., Filho, R. de M.E., Hendry, T., Holstein, D., Marsman, J., Mecklenburg, N., Malvar, S., Nunes, L.O., Padilha, R., Sharp, M., Silva, B., Sharma, S., Aski, V. and Chandra, R. 2024. RAG vs Fine-tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture. *arXiv.org*. [Online]. [Accessed 31 January 2024]. Available from: https://arxiv.org/abs/2401.08406.

Banerjee, S. and Lavie, A. 2005. *METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments}*[Online]. Ann Arbor, Michigan: Association for Computational Linguistics. Available from: https://www.aclweb.org/anthology/W05-0909.

Barnard, J. 2024. What are word embeddings? | IBM. *www.ibm.com*. [Online]. Available from: https://www.ibm.com/topics/word-embeddings.

BCS 2022. *BCS, THE CHARTERED INSTITUTE FOR IT CODE OF CONDUCT FOR BCS MEMBERS* [Online]. Available from: https://www.bcs.org/media/2211/bcs-code-of-conduct.pdf.

Blackboard Inc. 2023. Educational Technology Services. *www.blackboard.com*. [Online]. Available from: https://www.blackboard.com.

Ciolino, M., Noever, D. and Kalin, J. 2022. *Back Translation Survey for Improving Text Augmentation* [Online]. [Accessed 27 March 2024]. Available from: https://arxiv.org/pdf/2102.09708.pdf.

Collis, J. 2017. Glossary of Deep Learning: Word Embedding. *Deeper Learning*. [Online]. Available from: https://medium.com/deeper-learning/glossary-of-deep-learning-word-embedding-f90c3cec34ca.

Coursera 2018. Coursera | Online Courses & Credentials by Top Educators. Join for Free. *Coursera*. [Online]. Available from: https://www.coursera.org.

Duolingo 2019. Learn a language for free. *Duolingo*. [Online]. Available from: https://www.duolingo.com.

Feng, S., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T. and Hovy, E. 2021. *A Survey of Data Augmentation Approaches for NLP* [Online]. Available from: https://arxiv.org/pdf/2105.03075.pdf.

Gnatyuk, Y. 2023. Council Post: Leveraging Fine-Tuned GPT For Fintech Customer Support. *Forbes*. [Online]. [Accessed 20 February 2024]. Available from: https://www.forbes.com/sites/forbestechcouncil/2023/11/03/leveraging-fine-tuned-gpt-for-fintech-customer-support/#.

Google n.d. Bard. *bard.google.com*. [Online]. Available from: https://bard.google.com.

Graves, A. 2012. *Sequence Transduction with Recurrent Neural Networks* [Online]. [Accessed 19 April 2024]. Available from: https://arxiv.org/pdf/1211.3711.pdf.

Hendy, A., Abdelrehim, M., Sharaf, A., Raunak, V., Gabr, M., Matsushita, H., Kim, Y.J., Afify, M. and Awadalla, H.H. 2023. How Good Are GPT Models at Machine Translation? A Comprehensive Evaluation. *arXiv.org*. [Online]. Available from: https://arxiv.org/abs/2302.09210.

Hugging Face n.d. What is Sentence Similarity? - Hugging Face. *huggingface.co*. [Online]. Available from: https://huggingface.co/tasks/sentence-similarity.

Jacky 2023. What is RAG (Retrieval-Augmented Generation)? *Medium*. [Online]. [Accessed 20 February 2024]. Available from: https://colabdoge.medium.com/what-is-rag-retrieval-augmented-generation-b0afc5dd5e79.

Kooli, C. 2023. Chatbots in Education and Research: A Critical Examination of Ethical Implications and Solutions. *Sustainability*. **15**(7), p.5614.

Lin, C.-Y. 2004. ROUGE - a Hugging Face Space by evaluate-metric. *huggingface.co*. [Online]. Available from: https://huggingface.co/spaces/evaluate-metric/rouge.

Lund, B.D. 2023. A brief review of ChatGPT: its value and the underlying GPT technology. *Preprint. University of North Texas. Project: ChatGPT and Its Impact on Academia.*. **10**.

Miesle, P. 2023. What is Retrieval Augmented Generation. *DataStax*. [Online]. [Accessed 16 October 2023]. Available from: https://www.datastax.com/guides/what-is-retrieval-augmented-generation.

OpenAI 2023a. GPT-4 Technical Report. *arXiv (Cornell University)*. **4**.

OpenAI 2024. *new-and-improved-embedding-model* [Online]. Available from: https://openai.com/blog/new-embedding-models-and-api-updates.

OpenAI 2023b. OpenAI API. *Openai.com*. [Online]. Available from: https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset.

Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I. 2018. *Improving Language Understanding by Generative Pre-Training*.[Online] OpenAI. [Accessed 21 March 2024]. Available from: https://openai.com/research/language-unsupervised.

Tinn, R., Cheng, H., Gu, Y., Usuyama, N., Liu, X., Naumann, T., Gao, J. and Poon, H. 2023. Fine-tuning large neural language models for biomedical natural language processing. *Patterns*. **4**(4), p.100729.

Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F. and Mueller, A. 2015. Scikit-learn. *GetMobile: Mobile Computing and Communications*. **19**(1), pp.29–33.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. 2017. Attention Is All You Need. *Advances in neural information processing systems*. **30**.

Zhang, A., Lipton, Z.C., Li, M. and Smola, A.J. 2023. *Dive into Deep Learning* [Online]. Cambridge University Press. [Accessed 21 March 2024]. Available from: https://d2l.ai/index.html.

Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q. and Artzi, Y. 2020. BERTSCORE: EVALUATING TEXT GENERATION WITH BERT *In*: *ICLR 2020 Conferenc*.

Appendix A
Self-appraisal

A.1 Critical self-evaluation

Overall, I am satisfied with the outcome of this project although I felt it may be highly ambitious from the start. The project was a success but that does not suggest that it was straightforward or easy by any means. Starting without any prior experience in chatbots or natural language processing meant I had much to learn in a limited timeframe. Diving into testing practical applications early on proved highly beneficial, allowing me to learn more effectively than if I had relied just on reading documentation or maintaining a high-level understanding. Through this process, I developed a deep interest in chatbot technology and NLP and I think this interest became foundational for the success of the project.

In terms of implementing my ideas, I generally did not encounter significant hurdles. However, inevitable issues did arise during software development, and pinpointing the exact problems sometimes proved challenging. Occasionally, this required me to step back and reassess the project as a whole, realising that some aspects of my methodology needed to be changed, due to my inexperience with NLP. A deeper and thorough planning and idea exploration from the beginning would have pre-empted some of these issues.

As such, project management and planning were undoubtedly my greatest weaknesses. I often found myself uncertain about the next steps, reflecting a need for a more defined and planned approach to managing the project. Additionally, the tendency to revisit ideas repeatedly had mixed effects. While it allowed me to identify weaknesses in my approach, it also took time away from being able to explore these ideas thoroughly.

In conclusion, I view this project as a personal success. I enjoyed the learning process and succeeded in achieving the set objectives of improving GPT-3.5. The challenges I faced and overcame provided invaluable experience in research, machine learning, and technical analysis. Reflecting on the entire journey, I am pleased with the final product and the skills I have developed.

A.2 Personal reflection and lessons learned

On a personal note, this project really helped me shape my future goals and ambitions. Working with natural language processing, machine learning, and chatbots as a whole sparked a genuine interest in this area of computer science, confirming that it's a field I want to continue exploring in the future.

The project also highlighted some of my weaker tendencies in project planning. I realised that my commitment to a 'I know I'll get it done' attitude wasn't sufficient enough, especially for a project with so many new concepts to learn. I've learned to value a well thought out plan that anticipates setbacks and delays. Going forward, I'll know to make sure to allocate time for potential hurdles.

As I mentioned in my critical self-evaluation, I often was revisiting and refining my work and I believe this to be down to my perfectionist tendencies. I've since come to recognise that research can be endless, there will always something that can be improved. Not everything needs to be perfected right now; some things can be saved for later or for future projects.

A.3 Legal, social, ethical and professional issues

A.3.1 Legal issues

Regarding the legal issues related to this project, it's crucial to address Intellectual Property and Copyright concerns. Third party software used in this project, such as the OpenAI API and GPT models, is permissible to use under specified conditions set forth in OpenAI's terms of use. By registering online and obtaining valid API keys, the services were used in accordance with usage policies.

Additionally, Python modules and software libraries like Langchain or NumPy are freely available and do not require licensing. This project ensured compliance with all terms and conditions of the software used. In conclusion, the project did not breach any legal regulations, confirming proper adherence to intellectual property laws and copyright norms.

A.3.2 Social issues

The project addressed the potential for AI to transform educational accessibility. By enhancing digital learning platforms with AI, the technology promises to democratise education, making high-quality resources accessible to a broader audience. However, it can also raise concerns about AI "replacing humans" in educational institutions. This concern is mitigated by the fact that human educators created all the training materials, ensuring that the AI cannot function without human input.

A.3.3 Ethical issues

The primary ethical issues associated with AI typically involve bias and misinformation, which may be introduced during training or stem from the datasets used in retrieval-augmented generation. This is normally a result of human error, leading to misinformation originating from the main information source, the course data itself, rather than the data processing methodology. The project ensured it did not introduce any new information, bias, or prejudice in the data processing by solely using pre-existing lecture data as its only information source.

This project also involved a user evaluation and adhered strictly to the ethical guidelines throughout. All user participation was optional and no personal information was stored or used. Since the project did not utilise any personal information in the training of the models or implementation of RAG, it did not encounter any ethical issues.

A.3.4 Professional issues

This project was carried out adhering closely to the British Computing Society Code of Conduct (BCS, 2022). Additionally, any external work that influenced this project was properly attributed by referencing and providing links to any external materials in the appendices. Finally, a GitHub repository was used and maintained for version control to adhere to best practices and reduce the significant risk of losing the project’s codebase.

Appendix B
External Materials

All lecture material can be found on the COMP2121 Data Mining Minerva page with permission from the project supervisor (authorised access required) –

https://minerva.leeds.ac.uk/ultra/courses/_541874_1/outline

ChatGPT-3.5 can be accessed through the API or the OpenAI website –

https://chat.openai.com

OpenAI text embedding models can be accessed through the API –

https://platform.openai.com/docs/api-reference/embeddings

Hugging face was used for evaluation metrics –

https://huggingface.co/docs/evaluate/en/index

Appendix C

Consent Form (User Testing)

Title of Project: *Enhancing University Learning with Retrieval-Augmented Generation and GPT-3.5 Fine-Tuning*

Name of Project Student: Abdul Karim Abbas

Initial the box if you agree with the statement to the left

1	I confirm that I have read and understand the information sheet dated ____________ explaining the above project and I have had the opportunity to ask questions about the project.	☐
2	I understand that my participation is voluntary and that I am free to withdraw at any time without giving any reason and without there being any negative consequences. In addition, should I not wish to answer any particular question or questions, I am free to decline. Contact project student: sc21aka@leeds.ac.uk	☐
3	I understand that my responses will be kept strictly confidential. I understand that my name will not be linked with the project materials, and I will not be identified or identifiable in the report or reports that result from the project.	☐
4	I agree for the data collected from me to be used in future research.	☐
5	I agree to take part in the above project and will inform the project student should my contact details change.	☐

________________________	________________	____________________
Name of participant (*or legal representative*)	Date	Signature
________________________	________________	____________________
Name of person taking consent (*if different from project student*) *To be signed and dated in presence of the participant*	Date	Signature
________________________	________________	____________________
Project student *To be signed and dated in presence of the participant*	Date	Signature

Appendix D

Project Information Sheet

Project Title: *Enhancing University Learning with Retrieval-Augmented Generation and GPT-3.5 Fine-Tuning*

You are being invited to take part in a student project. Before you decide, it is important for you to understand the aim of the project and what participation will involve. Please take time to read the following information carefully and discuss it with others if you wish. Ask if there is anything that is not clear or if you would like more information. Take time to decide whether or not you wish to take part. Thank you for reading this.

Project Aim: *To develop and evaluate a chatbot aimed at enhancing the teaching of university modules.*

Why have I been chosen?
Participants are chosen at random with the sole requirement that they have taken the module COMP2121 Data Mining.

Do I have to take part?
It is up to you to decide whether or not to take part. If you do decide to take part, you will be given this information sheet to keep (and be asked to sign a consent form) and you can still withdraw at any time. You do not have to give a reason.

What will happen to me if I take part?
You will be required to take a short assessment in which you are shown 10 questions. For each question, there will be 3 example answers. You will be asked to pick which answer you think is the best in your opinion.

Will my taking part in this project be kept confidential?
Your replies will be kept completely anonymous and no personal information will be stored.

What type of information will be sought from me and why is the collection of this information relevant for achieving the project's objectives?
Your choices will be stored anonymously, and they will be used to evaluate which model has achieved the best feedback from users who have taken the module.

What will happen to the results of the project?
The results of this project will be published in a report to be submitted for assessment at the end of the undergraduate module COMP3931 Individual Project in the School of Computing at the University of Leeds.

Contact for further information:
sc21aka@leeds.ac.uk

If you decide to participate in this project, you will be given a copy of this information sheet and a signed consent form to keep. Thank you very much for taking the time to read this information sheet.

Appendix E
User Evaluation Q&A

Model	What is the similarity between named entity recognition and information retrieval?
gpt-3.5-turbo	Both named entity recognition and information retrieval involve extracting relevant information from a given text or dataset. Named entity recognition focuses on identifying and classifying named entities such as people, organizations, locations, and dates, while information retrieval involves retrieving relevant documents or data based on a user's query. Both tasks require natural language processing techniques and algorithms to accurately identify and extract relevant information from unstructured text.
gpt-3.5-turbo + RAG	The similarity between named entity recognition and information retrieval is that both involve the identification and classification of entities in text. Named entity recognition specifically focuses on identifying proper names and other references to entities in text and categorizing them into predefined categories such as persons, organizations, locations, dates, and times. On the other hand, information retrieval involves searching for documents or information based on keywords or queries provided by the user. Both processes are essential in extracting meaningful information from text data. This information was extracted from the lecture titled "5.2 Information Extraction.pdf".
gpt-3.5-turbo FT + RAG	Both named entity recognition and information retrieval involve identifying proper names and other references to entities in text.

Model	Who founded Sketch Engine?
gpt-3.5-turbo	Sketch Engine was founded by Adam Kilgarriff and Milos Jakubicek in 2003.
gpt-3.5-turbo + RAG	The founder of Sketch Engine is Adam Kilgarriff, as mentioned in the lecture content ""1-3 Introduction to SketchEngine.pdf"".
gpt-3.5-turbo FT + RAG	Adam Kilgarriff

Model	What is named entity recognition?
gpt-3.5-turbo	Named entity recognition (NER) is a natural language processing (NLP) technique that is used to identify and classify named entities mentioned in a text into predefined categories such as names of persons, organizations, locations, dates, and more. The goal of NER is to extract and label specific entities within a text to better understand the context and extract useful information from the text. NER is commonly used in various NLP applications such as information retrieval, question answering, and sentiment analysis."
gpt-3.5-turbo + RAG	Named entity recognition is the identification of proper names and other references to entities in text and their classification into a set of categories of interest. It involves finding spans of text that constitute proper names and tagging the type of the entity, such as person, organization, location, date, time, or other types depending on the task at hand. This information extraction technique serves as a foundation for building more complex information extraction systems. (Source: lecture 3.2.pptx)
gpt-3.5-turbo FT + RAG	Named entity recognition is the identification of proper names and other references to entities in text and their classification into a set of categories of interest, such as persons, organizations, locations, dates, times, and other specific entity classes.

Model	**What is the necessary step to ensure a successful connection according to the lecture?**
gpt-3.5-turbo	The necessary step to ensure a successful connection according to the lecture is to actively listen and engage with the other person, showing genuine interest in what they are saying and responding appropriately. This involves being present in the moment, maintaining eye contact, and using verbal and nonverbal cues to demonstrate understanding and empathy. By actively listening and engaging with the other person, a successful connection can be established and maintained.
gpt-3.5-turbo + RAG	The necessary step to ensure a successful connection according to the lecture ""1-3 Introduction to SketchEngine.pdf"" is to connect via the VPN. The lecturer mentioned that one must make sure to connect via the VPN for YouTube videos and Sketch Engine to work properly.
gpt-3.5-turbo FT + RAG	You have to make sure you connect via the VPN.

Model	**What is data wrangling in the context of data mining?**
gpt-3.5-turbo	Data wrangling in the context of data mining refers to the process of cleaning, transforming, and organizing raw data into a format that is suitable for analysis. This involves tasks such as removing duplicates, handling missing values, standardizing data formats, and merging data from multiple sources. Data wrangling is a crucial step in the data mining process as it ensures that the data used for analysis is accurate, complete, and consistent."
gpt-3.5-turbo + RAG	In the context of data mining, data wrangling refers to the process of taking data from various different sources and transforming it into the right format so that machine learning algorithms can effectively work on it. This process typically involves converting the data into a structured format, such as a spreadsheet with rows and columns, where each row represents an instance and each column represents a feature. Data analysts often spend a significant amount of time on data wrangling, which involves tasks like data collection, data understanding, data annotation, and preparing the data for analysis. This information is from the lecture titled ""1-1 Background practical applications.pdf"".

gpt-3.5-turbo FT + RAG	Data wrangling in the context of data mining involves taking data from various different sources and getting it into the right format so that machine learning can work on it.

Model	Who is the Professor of Artificial Intelligence for Language in the School of Computing at Leeds University?
gpt-3.5-turbo	The Professor of Artificial Intelligence for Language in the School of Computing at Leeds University is Prof. He Yulan.
gpt-3.5-turbo + RAG	The Professor of Artificial Intelligence for Language in the School of Computing at Leeds University is Eric Atwell. This information is from the lecture titled ""1-1 Background practical applications.pdf"".
gpt-3.5-turbo FT + RAG	Eric Atwell

Model	How is the module structured?
gpt-3.5-turbo	The module is typically structured in a hierarchical format, with different sections or units organized in a logical sequence. Each section may contain sub-sections or modules that further break down the content into smaller, more manageable chunks. The module may also include an introduction, learning objectives, instructional materials, activities, assessments, and a conclusion or summary. Additionally, the module may include links to additional resources, references, and further reading materials for students to explore. The structure of the module is designed to guide students through the learning process and help them achieve the stated learning objectives.
gpt-3.5-turbo + RAG	The module is structured into six units. The first unit is an introduction to Corpus linguistics, text analytics, and sketch engine. The second unit delves into more data about text data mining and a tool called Weka (Waikato Environment for Knowledge Analysis). The third unit focuses on the meanings of words and text to compare two texts based on their meanings. Additionally, there are tests conducted throughout the course, with an online test covering units one and two, and a final test covering all units from one to six. The assessment includes two tests, with the first test accounting for

	20% of the marks and the second test accounting for 30% of the marks.
gpt-3.5-turbo FT + RAG	The module is structured into six units. Unit one serves as an introduction to Corpus linguistics, text analytics, and sketch engine. Unit two delves deeper into text data mining and the Weka tool. Unit three focuses on comparing the meanings of words and text, while unit four looks at information retrieval from the web.

Model	**What are the two types of chatbot architectures mentioned in the lectures?**
gpt-3.5-turbo	The two types of chatbot architectures mentioned in the lectures are rule-based chatbots and machine learning-based chatbots.
gpt-3.5-turbo + RAG	The two types of chatbot architectures mentioned in the lectures are rule-based systems and corpus-based systems. Rule-based systems involve computational linguists writing rules, trying them out, and extending them until the system works well. On the other hand, corpus-based systems use a training corpus that is appropriately annotated, and a machine learning algorithm extracts rules from it.
gpt-3.5-turbo FT + RAG	The two types of chatbot architectures mentioned are rule-based systems and corpus-based machine learning systems.

Model	**Tell me about the final group project.**
gpt-3.5-turbo	The final group project is a collaborative effort that typically takes place towards the end of a course or program. It often involves a team of students working together to complete a specific task or solve a problem within a set timeframe. The project can vary in scope and complexity depending on the requirements of the course or program. It may involve conducting research, analyzing data, developing a presentation, creating a product or service, or any other type of project that demonstrates the students' knowledge and skills in the subject area. Group members are usually assigned specific roles and responsibilities to ensure that

	the project is completed successfully. They must communicate effectively, collaborate, and coordinate their efforts to achieve the desired outcome. The final group project is typically presented to the instructor or a panel of judges for evaluation and feedback. It provides students with an opportunity to showcase their skills, creativity, and teamwork, and to demonstrate what they have learned throughout the course or program.
gpt-3.5-turbo + RAG	The final group project for COMP2121 Data Mining involves developing a research project proposal that utilizes data mining and text analytics theory, methods, and technologies for a practical application of the students' choice. The project report must be a maximum of seven pages long and should include a research project proposal outlining how data mining and text analytics will be applied in the chosen practical application. Students are encouraged to refer to the Engineering and Physical Sciences Research Council guidance on how to write research project proposals for assistance in preparing their project reports.
gpt-3.5-turbo FT + RAG	For the final group project, students have to write and submit a report that is a maximum of seven pages long. The report should be a research project proposal using data mining and text analytics for a practical application of their choice.

Model	**In what lecture does Eric Atwell talk about multi-word expressions?**
gpt-3.5-turbo	Eric Atwell talks about multi-word expressions in his lecture on Natural Language Processing (NLP).
gpt-3.5-turbo + RAG	Eric Atwell talks about multi-word expressions in the lecture titled "6.2 Multi-word expressions.pdf."
gpt-3.5-turbo FT + RAG	Lecture title: 6.2 Multi-word expressions

Appendix F
Code Repository

All code can be accessed through the GitHub Repository:

https://github.com/icekreem27/final-year-project-icekreem27.git

For code reproducibility:

```
git clone https://github.com/icekreem27/final-year-project-icekreem27.git

cd final-year-project-icekreem27

pip install -r requirements.txt
```