|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| Pedagogical chatbot specializing in financial knowledge  A TEXT ANALYTICS PROJECT |
| |  |  |  | | --- | --- | --- | | Dheeraj Banna | Abhijeet Sharma | Fuad Bin Saif | |

Table of Contents

[Chapter 1: Executive Summary: 2](#_Toc165452233)

[Chapter 2: Problem Definition & Significance: 3](#_Toc165452234)

[2.1 Enterprise Perspective: 3](#_Toc165452235)

[2.2 Educational Perspective: 4](#_Toc165452236)

[Chapter 3: Prior Literature: 5](#_Toc165452237)

[Chapter 4: Data Source & Preparation: 7](#_Toc165452238)

[4.1 Sourcing Data: 7](#_Toc165452239)

[4.2 Data Cleaning: 8](#_Toc165452240)

[Chapter 5: Exploratory Data Analysis & Visualizations 8](#_Toc165452241)

[Chapter 6: Text Analytics & Results 9](#_Toc165452242)

[6.1 Analytic Approach: 10](#_Toc165452243)

[6.2 Choice of Approaches: 11](#_Toc165452244)

[6.3 Trustworthiness of Analysis: 11](#_Toc165452245)

[Chapter 7: Insights & Recommendations: 12](#_Toc165452246)

[7.1 Insights: 12](#_Toc165452247)

[7.2 Recommendations: 13](#_Toc165452248)

[References and Prior work: 14](#_Toc165452249)

[Appendix: 14](#_Toc165452250)

# Chapter 1: Executive Summary:

In this project, we used a novel approach called the Retrieval-Augmented Generation (RAG) architecture to train the chatbot on finance questions. RAG is a state-of-the-art architecture capable of marrying retrieval-based and generation-based models in natural language processing. At its heart, RAG is a two-staged architecture composed of retrieval and generation components. In the retrieval stage, a retrieval model discerns relevant passages from a corpus of financial documents to match against an end user’s query. This is why the chatbot’s retrieval process ensures the quality and contextual fitness of the information conveyed. The second stage is the generation stage, where the chatbot combines the information retrieved to create a canonical response using a generation model that aims to create human-sounding responses that successfully handle the user’s request. The RAG architecture, as a result, allows the chatbot to produce nuanced and informative responses based on the user’s question.

Using the cloud version of the RAG architecture that we developed in our project; we designed a multi-purpose chatbot application to serve industries in the profit sector and the non-profit sector. In the enterprise domain, the chatbot serves as an enterprise data catalogue. It makes it easy for business users to discuss company policies and facts. To train the chatbot on company in-house documents helps us calibrate the natural language understanding skills of the system to the specific cultures and legal environments of the individual companies.

Additionally, we adapted it to work with academic books so that it can be used in education to give students access to in-depth sources of financial information without the help of individual teachers – a significant issue in the huge class sizes found in developing countries. Integrating the RAG architecture into our chatbot is transformative in some important ways. More advanced techniques in AI allow us to offer our clients (who are mainly based in developing countries) a solution that meets multiple needs with current technology, but also opens up the path to many potential further solutions in artificial education and enterprise communication.

All in all, the RAG architecture is the foundation of this project, on which we built a highly flexible chatbot that can accommodate the various learning and business demands in the finance dom

# Chapter 2: Problem Definition & Significance:

The target clients for this project encompass two distinct groups: enterprises seeking streamlined communication of company policies and students in need of assistance with finance-related subjects. The core business problem addressed is the pervasive challenge of accessing comprehensive and accessible information on finance topics.

## 2.1 Enterprise Perspective:

Efficient communication of company policies and financial information is crucial for enterprises to maintain organizational coherence and comply with regulatory standards. However, traditional methods like manuals or intranet portals often fall short in terms of accessibility and clarity. Employees can struggle to find the information they need in a sea of documents, leading to delays in decision-making and potential compliance issues.

To address this challenge, we have introduced a chatbot powered by the RAG architecture, aiming to transform how enterprises handle and communicate their financial policies. This chatbot acts as an intelligent interface, making it easy for employees to access and understand complex financial information. This not only boosts organizational efficiency but also reduces the risk of misinterpretation and non-compliance.

2.2 Educational Perspective:

Educationally, students encounter similar hurdles when seeking resources and support for finance-related subjects, especially in regions with high teacher-to-student ratios. This situation restricts students' access to personalized guidance and support, hindering their academic progress and engagement.

Our solution tackles this problem by empowering students to engage in self-directed learning through our chatbot. By training the chatbot on academic books and finance-related materials, we offer students a comprehensive and user-friendly tool to grasp intricate financial concepts. This approach not only enhances learning outcomes but also cultivates independence and critical thinking skills vital for success in today's knowledge-driven economy.

**Significance:**

# These challenges aren’t trivial. Our proposed solution has the ability to address multiple broad categories: if we can provide enterprises with an important communication tool, and students with a powerful learning tool, our solution has the ability to create real benefits across a range of areas.

# i. better learning outcomes: Students have access to high-quality educational material; their understanding of financial topics increases, and their performance improves.

# ii. greater efficiency and compliance: for organisations, more efficient lines of communication, as well as access to company policy more easily is greater efficiency and compliance.

# iii. Student Empowerment in the Developing World: When the number of students to each teacher is high, our solution helps students gain control of their own learning through self-education.

# Chapter 3: Prior Literature:

• **Research Question:** What are the primary ingredients of a chatbot that can engage in dialogue in a supportive manner with survivors of sexual harassment?

• **Methodology:** Texts are classified (homoerotic, anti-gay and neutral) and tokenised through TF-IDF, Doc2Vec, and BERT fine-tuned for NER. Logistic Regression is used for classification tasks.

• **Results**: Conclusion: the chatbot works well in processing inbound texts and labeling them. BERT models work quite well in recognising location entities. The Stanford NER model’s performance is slightly lower.

**Paper 2**: "Enhancing Healthcare Communication through Natural Language Understanding" by Arun Babu et al.

• Fine-tuning BERT for medical contexts by considering patient data to personalise feedback.

• **Methodology**: Explores BERT’s bidirectional nature and its fine-tuning for biomedical language and tasks and finds that BERT can lead to state-of-the-art performance over baselines in different metrics.

• **Results and Discussion**: The medical chatbot based on BERT is able to efficiently communicate, with a high level of adequacy and precision, to answer common queries made in the fields of healthcare.

**Paper 3:** ‘Developing a Smart Chat-bot Tool to Help High School Students Learn General Knowledge Subjects’ by Debasatwa Dutta.

• **Research Questions:** Explores text processing components and language processing capabilities of various chatbot platforms.

• **Methodology:** Involves text processing phases and assessment of chatbot platforms like Dialogflow.com and Wit.ai.

• **Findings and Conclusion**: The intelligent chatbot is a functional language interaction tool that can accept and interpret user input texts, as well as serving as a small talk interface for learners to practise English. It is possible to utilise the intelligent chatbot in the learning environment.

**Paper 4**: "JAICOB: A Data Science Chatbot" by Carlander-Reuterfelt et al.

• **Research Question**: Develop an Agent Architecture for Principled Pedagogical Question Answering under Data Science and Machine Learning.

**Methodology:** modular architecture (Knowledge Base, Question Answering, Speech Act Classifier, Small Talk Modules), uses ML (Support Vector Machine), to improve performance.

**• Finding s and Conclusion:** Shows a high accuracy and discriminant validity, and Answering Accuracy is found to highly influence user satisfaction and behavioural intentions.

**Paper 5:** "AI Chatbot for Tourism Recommendations: A Case Study in the City of Jeddah, Saudi Arabia" by Dr. Reem Alotaibi et al.

* **Research Question:** Develops and evaluates a text-based chatbot mobile application, "Smart Guidance," to assist tourism and hospitality in Jeddah city.
* **Methodology:** Involves the development of the chatbot using Rasa.ai framework and training with a dataset containing various user scenarios.
* **Findings & Conclusion:** Demonstrates the effectiveness of the chatbot in understanding and responding to user inquiries, with potential for improving trip planning and enhancing city tourism and hospitality.

**Paper 6:** "PaperPersiChat" by Alexander Chernyavskiy et al.

* **Research Question:** Addresses the need for an open chatbot system for discussing scientific papers, supporting summarization and question-answering modes.
* **Methodology:** Develops the PaperPersiChat system using discourse analysis and a dataset of computer science papers.
* **Findings & Conclusion:** Successfully implements the chatbot system, emphasizing the need for refining individual submodules and dialogue management for greater flexibility.

# Chapter 4: Data Source & Preparation:

## 4.1 Sourcing Data:

## The information for this project was carefully gathered from trustworthy and credible sources to guarantee its accuracy and significance. The primary sources included:

## **Academic Books:** Finance-related academic books were obtained from reputable publishers and educational institutions, serving as reliable resources covering various finance topics ranging from fundamental principles to advanced ideas.

## **Financial Websites:** Content from respected financial websites like Investopedia was gathered alongside the academic literature to enhance the breadth of information provided. These websites present detailed explanations of financial terms, concepts, and tactics, offering beneficial knowledge for individuals seeking insight in this field.

## 4.2 Data Cleaning:

The data was extensively cleaned to maintain its integrity and reliability for analysis. The cleaning process included several key steps:

**Eliminating irrelevant sections**: Non-essential parts of the documents were identified and excluded to align the dataset with project objectives focused on finance-related content

**Ensuring consistency**: Consistency checks were conducted to standardize formatting, eliminate terminology inconsistencies, and enhance coherence in content across all documents.

**Verification of authenticity**: Data authenticity was validated by cross-referencing information with credible sources to ensure alignment with real-world financial practices and principles.

# Chapter 5: Exploratory Data Analysis & Visualizations

Exploring the text data derived from financial documents is crucial for uncovering insights into prevalent themes and patterns. This chapter focuses on the exploratory data analysis (EDA) carried out on the pre-processed text data, utilizing various techniques to achieve a thorough understanding.

**Word Cloud Analysis**: Visual representations of word clouds offer an attractive display of the most common words in the processed text. By creating word clouds for each document, we can identify significant terms and themes present in the dataset efficiently. This approach provides a rapid overview of key topics addressed in financial documents, facilitating recognition of recurring concepts.

**Frequency Analysis**: An assessment was conducted to determine the frequency of common words across all documents. Through bar charts illustrating top word frequencies, we obtained valuable insights into prevalent themes and subjects discussed within financial documents. Identifying frequently used words enables us to grasp primary focus areas and areas of interest effectively.

**Word Length Distribution**: The distribution pattern of word lengths in processed text data was analysed using box plots. These visualizations succinctly summarize how word lengths are distributed, incorporating metrics like median values, quartiles, and potential outliers. Examining this distribution offers valuable insights into language complexity and variability evident in financial documentation.

**Word Frequency vs. Length Relationship**: Scatter plots were utilized to investigate correlations between word frequencies and lengths systematically. This visualization aids comprehension regarding information density dispersion within processed textual content. Unveiling relationships between these variables facilitates identification of patterns as well as variations found in textual material effectively throughout appendix visuals showcasing such analyses.

# Chapter 6: Text Analytics & Results

In this section, we outline the methodology employed for text analytics to scrutinize processed textual data and draw significant conclusions. The analysis encompasses various methodologies such as sentiment assessment, topic modelling, and keyword identification.

**Sentiment Assessment:** An evaluation of sentiments was carried out to gauge the general emotional tone conveyed in financial texts. By categorizing sentiments as positive, negative, or neutral, valuable insights into prevalent attitudes and opinions expressed in the documents were obtained.

**Topic Modeling:** Utilization of techniques like Latent Dirichlet Allocation (LDA) allowed us to uncover hidden topics within processed textual data. This examination revealed underlying themes and subjects discussed across financial documents while offering a structured presentation of content.

**Keyword Identification:** Algorithms for extracting keywords were utilized to pinpoint critical terms and phrases relevant within the financial context. This method facilitated a deeper comprehension by highlighting essential concepts and topics addressed in the documents.

**Comparison with Standards & Validation:** Our findings underwent benchmarking against established methods from previous research which showcased our text analytics' efficiency and resilience. Through leveraging advanced natural language processing strategies on analyzed textual data, actionable insights were extracted empowering users towards informed decision-making within finance-related domains.

## 6.1 Analytic Approach:

In this project, a combination of text preprocessing techniques, natural language processing (NLP) models, and machine learning algorithms was utilized to analyze finance-related text data extracted from PDF documents. The primary analytic methods employed included initial text preprocessing steps to clean the raw data for enhanced quality. This involved actions such as lowercasing, punctuation removal, tokenization, stop word elimination, and lemmatization to ensure the dataset's cleanliness before further analysis.

To develop a specialized chatbot designed for financial inquiries, we implemented the Retrieval-Augmented Generation (RAG) architecture - merging retrieval-based and generation-based models. During retrieval stages in our chatbot system that caters to both profit-driven organizations and non-profits within the finance sector; relevant information is sourced from a corpus of financial documents based on user queries then synthesized into coherent responses during generation phases.

## 6.2 Choice of Approaches:

We chose the approaches for several reasons:

**Effectiveness**: Text preprocessing techniques are crucial in preparing data by cleaning it up through lowercase conversion or lemmatization resulting in improved performance for downstream NLP applications.

**State-of-the-Art Model:** Integration of retrieval-based with generation-based strategies within RAG architecture facilitates contextually pertinent responses enhancing user experience significantly.

**Versatility**: By training our chatbot using finance-related resources ensures alignment with specific client needs across diverse sectors including enterprises and students interested in Finance studies.

## 6.3 Trustworthiness of Analysis:

The reliability of our analysis can be assessed through various criteria:

**Data Quality:** Our data was obtained from trustworthy sources such as academic literature and reputable financial websites, ensuring the integrity and accuracy of our analysis.

**Reproducibility:** We have provided comprehensive documentation of our analytical methodology, including code snippets and explanations, to promote transparency and enable other researchers to replicate our approach for validating the precision of our results.

In order to evaluate our method further, we conducted a comparison between two language model architectures – BERT (Bidirectional Encoder Representations from Transformers) and GPT-2 (Generative Pre-trained Transformer 2).

**For the BERT model:** We fine-tuned the 'bert-base-uncased' pre-trained model specifically for masked language modeling using processed text extracted from financial documents. The training process involved tokenization of text, establishment of a dataset for masked language modeling, and training the model over 10 epochs. Despite encountering issues with some unused weights in the model and receiving a future warning related to deprecated implementation of AdamW optimizer; however, it is worth noting that despite these challenges faced during training phase ,the refined BERT model displayed promising performance trends. Notably,the average loss per epoch exhibited consistent decrease signifying successful acquisition on linguistic representations within financial text corpus context.

GPT-2 Model

We refined the 'gpt2' pre-trained model for text generation through a training process that involved text preprocessing, tokenization, dataset creation, and 10 epochs of model training. The GPT-2 model demonstrated impressive performance with significant reductions in average loss per epoch throughout the training phase. There were no issues encountered regarding unused weights or deprecated optimizers during this process.

Based on our benchmarking findings, we suggest further investigation into challenges faced while fine-tuning the BERT model including unused weights and outdated optimizer implementation. Resolving these issues could potentially enhance the efficiency and effectiveness of fine-tuning procedures leading to improved model performance.

Additionally, given the exceptional text generation capabilities of the GPT-2 model without facing similar obstacles, we propose exploring its potential applications in other text analysis tasks like summarization, sentiment analysis, and question answering.

# Chapter 7: Insights & Recommendations:

## 7.1 Insights:

Our analysis utilizing the Retrieval-Augmented Generation (RAG) framework to design a finance-focused chatbot has revealed valuable insights with potential benefits for our clients, spanning enterprises and students:

Enhanced Communication Efficiency: The chatbot acts as an intelligent interface within organizations, facilitating easy access and comprehension of intricate financial data by employees. Through optimizing communication channels and offering prompt access to company policies and financial information, the chatbot boosts operational efficiency while minimizing the chances of misinterpretation or non-compliance.

Fostering Self-Directed Learning: Designed as a comprehensive educational tool for students, the chatbot is equipped with knowledge gleaned from academic texts and finance-related materials. This curated content empowers students in regions facing high teacher-to-student ratios to engage in self-directed learning, enabling them to pursue independent study opportunities leading to improved academic achievements.

## 7.2 Recommendations:

The following actionable recommendations are provided for clients based on the insights gathered:

**For Enterprise Clients:**

Implementation Strategy: Integrate the chatbot into the intranet or communication platform to facilitate easy access for employees. Conduct training sessions and provide user guides to acquaint employees with the chatbot's functions and promote its adoption.

**Feedback Mechanism:** Establish a feedback system to collect input from employees on the chatbot's performance. Utilize this feedback to continuously enhance its functionality and address any user concerns.

**Integration with Existing Systems:** Explore opportunities to connect the chatbot with current enterprise systems like HR portals or knowledge management platforms, enabling real-time access and retrieval of relevant information.

**For Educational Institutions:**

Integration into Learning Platforms: Incorporate the chatbot into existing learning management systems or educational platforms used by students for seamless resource accessibility, promoting its use as an additional learning aid.

**Promotion and Awareness**: Initiate promotional campaigns among students highlighting benefits of using the chatbot. Emphasize its role in self-directed learning alongside traditional classroom instruction.

**Feedback and Improvement**: Set up mechanisms for gathering student feedback concerning their interaction with the chatbot. Use this input to pinpoint areas needing improvement, enhancing relevance and effectiveness in supporting student learning.

By implementing these suggestions, clients can optimize value gained from utilizing a chatbots leading improved communication efficiency, enhanced learning outcomes, increased organizational productivity.

# References and Prior work:

1. Bauer, T., Devrim, E., Glazunov, M., Lopez Jaramillo, W., Mohan, B., & Spanakis, G. (2020). Building a chatbot to assist survivors of sexual harassment. arXiv preprint arXiv:2005.12888.

2. Babu, A., & Boddu, S. B. (2020). Enhancing healthcare communication through natural language understanding. IEEE Access, 8, 116522-116532.

3. Dutta, D. (2019). Developing an intelligent chat-bot tool to assist high school students for learning general knowledge subjects. In Proceedings of the 10th International Conference on Educational Data Mining (EDM) (pp. 283-288).

4. Carlander-Reuterfelt, J., Djurfeldt, M., Khazendar, A., & Westlund, J. (2020). JAICOB: A data science chatbot. International Journal of Artificial Intelligence in Education, 30(2), 228-252.

5. Alotaibi, R., Alshahri, S., & Alamri, A. (2021). AI chatbot for tourism recommendations: A case study in the city of Jeddah, Saudi Arabia. Journal of Tourism and Hospitality Management, 9(2), 147-157.

6. Chernyavskiy, A., Khazov, F., Seleznev, A., Shmatov, D., & Biryukov, A. (2020). PaperPersiChat: Open-domain scientific paper chatbot with discourse-aware summarization and question answering. arXiv preprint arXiv:2005.09127.

# Appendix:

**Mistral\_rag**

**from** google.colab **import** drive

drive**.**mount('/content/drive', force\_remount**=True**)

Mounted at /content/drive

In [ ]:

**!**pip install langchain

**!**pip install torch

**!**pip install sentence\_transformers

**!**pip install faiss-cpu

**!**pip install huggingface-hub

**!**pip install pypdf

**!**pip -q install accelerate

**!**pip install llama-cpp-python

**!**pip -q install git+https://github.com/huggingface/transformers

Collecting langchain

Downloading langchain-0.1.16-py3-none-any.whl (817 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 817.7/

In [ ]:

**from** langchain.chains **import** RetrievalQA

**from** langchain.embeddings **import** HuggingFaceEmbeddings

**from** ctransformers **import** AutoModelForCausalLM, AutoTokenizer

**from** langchain.text\_splitter **import** RecursiveCharacterTextSplitter

**from** langchain.vectorstores **import** FAISS

**from** langchain.document\_loaders **import** PyPDFDirectoryLoader

In [ ]:

**!**pip install ctransformers

Collecting ctransformers

Downloading ctransformers-0.2.27-py3-none-any.whl (9.9 MB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.9/9.9

In [ ]:

**from** langchain.llms **import** HuggingFacePipeline

**from** langchain.chains **import** RetrievalQA

**from** langchain.vectorstores **import** FAISS

In [ ]:

**!**pip install transformers torch

**!**pip install transformers torch langchain faiss-cpu

In [ ]:

*# Install necessary libraries*

**!**pip install ctransformers faiss-cpu transformers langchain

In [ ]:

*# Install necessary libraries*

**!**pip install langchain ctransformers transformers faiss-cpu PyPDF2

In [ ]:

**!**pip install pdf2image pytesseract

In [ ]:In [ ]:

**import** pdf2image

**import** pytesseract

**import** re

**import** nltk

**from** nltk.corpus **import** stopwords

**from** nltk.tokenize **import** word\_tokenize

**from** nltk.stem **import** WordNetLemmatizer

*# Ensure necessary resources are downloaded*

nltk**.**download('stopwords')

nltk**.**download('punkt')

nltk**.**download('wordnet')

*# Define the list of PDF files*

*# List of PDF files*

pdf\_files **=** ["drive/MyDrive/Articles/1.pdf",

"drive/MyDrive/Articles/2.pdf",

"drive/MyDrive/Articles/3.pdf",

"drive/MyDrive/Articles/4.pdf",

"drive/MyDrive/Articles/5.pdf",

"drive/MyDrive/Articles/6.pdf",

"drive/MyDrive/Articles/7.pdf",

"drive/MyDrive/Articles/8.pdf",

"drive/MyDrive/Articles/9.pdf",

"drive/MyDrive/Articles/10.pdf",

"drive/MyDrive/Articles/11.pdf",

"drive/MyDrive/Articles/12.pdf",

"drive/MyDrive/Articles/13.pdf",

"drive/MyDrive/Articles/14.pdf",

"drive/MyDrive/Articles/15.pdf",

"drive/MyDrive/Articles/16.pdf",

"drive/MyDrive/Articles/17.pdf",

"drive/MyDrive/Articles/18.pdf",

]

*# Create an instance of the lemmatizer*

lemmatizer **=** WordNetLemmatizer()

*# Load English stopwords*

stop\_words **=** set(stopwords**.**words('english'))

*# Dictionary to store processed text for each file*

processed\_texts **=** {}

*# Function to preprocess text*

**def** preprocess\_text(text):

*# Convert to lowercase*

text **=** text**.**lower()

*# Remove punctuation and special characters*

text **=** re**.**sub(r'[^\w\s]', '', text)

*# Tokenize the text into words*

words **=** word\_tokenize(text)

*# Remove stopwords and lemmatize the words*

words **=** [lemmatizer**.**lemmatize(word) **for** word **in** words **if** word **not** **in** stop\_words]

*# Join words back into a string*

**return** ' '**.**join(words)

*# Loop through each PDF file, extract text, and preprocess*

**for** pdf\_file **in** pdf\_files:

*# Convert PDF to images*

images **=** pdf2image**.**convert\_from\_path(pdf\_file)

*# Extract text using OCR*

ocr\_text **=** ""

**for** img **in** images:

ocr\_text **+=** pytesseract**.**image\_to\_string(img)

*# Preprocess the extracted text*

preprocessed\_text **=** preprocess\_text(ocr\_text)

*# Store the preprocessed text*

processed\_texts[pdf\_file] **=** preprocessed\_text

*# Display the processed text for each file*

**for** file\_path, text **in** processed\_texts**.**items():

print(f"Processed text from {file\_path}:")

print(text)

print("\n---\n")

[nltk\_data] Downloading package stopwords to /root/nltk\_data...

[nltk\_data] Package stopwords is already up-to-date!

[nltk\_data] Downloading package punkt to /root/nltk\_data...

[nltk\_data] Package punkt is already up-to-date!

[nltk\_data] Downloading package wordnet to /root/nltk\_data...

[nltk\_data] Package wordnet is already up-to-date!

IOPub data rate exceeded.

The notebook server will temporarily stop sending output

to the client in order to avoid crashing it.

To change this limit, set the config variable

`--NotebookApp.iopub\_data\_rate\_limit`.

Current values:

NotebookApp.iopub\_data\_rate\_limit=1000000.0 (bytes/sec)

NotebookApp.rate\_limit\_window=3.0 (secs)

In [ ]:

**import** re

*# Function to remove HTTP content from text*

**def** remove\_http\_content(text):

*# Regular expression pattern to match URLs*

url\_pattern **=** r'https?://\S+'

*# Replace URLs with an empty string*

cleaned\_text **=** re**.**sub(url\_pattern, '', text)

**return** cleaned\_text

In [ ]:

*# Create a new dictionary with cleaned keys*

cleaned\_processed\_texts **=** {

remove\_http\_content(key): value

**for** key, value **in** processed\_texts**.**items()

}

*# Display the dictionary with cleaned keys*

**for** file\_path, text **in** cleaned\_processed\_texts**.**items():

print(f"Updated key: {file\_path}")

print(f"Associated text: {text}")

print("\n---\n")

IOPub data rate exceeded.

The notebook server will temporarily stop sending output

to the client in order to avoid crashing it.

To change this limit, set the config variable

`--NotebookApp.iopub\_data\_rate\_limit`.

Current values:

NotebookApp.iopub\_data\_rate\_limit=1000000.0 (bytes/sec)

NotebookApp.rate\_limit\_window=3.0 (secs)

In [ ]:

*# The name of the text file to create*

file\_name **=** 'myd1.txt'

*# Open the file in write mode (or create it if it doesn't exist)*

**with** open(file\_name, 'w') **as** f:

*# Loop through the key-value pairs in the dictionary*

**for** key, value **in** cleaned\_processed\_texts**.**items():

*# Write each key-value pair to the file with a separator*

f**.**write(f"{key}: {value}\n") *# key and value with a separator and newline*

print(f"The key-value pairs from cleaned\_processed\_texts have been written to '{file\_name}'.")

The key-value pairs from cleaned\_processed\_texts have been written to 'myd1.txt'.

**MAIN CODE**

In [ ]:

*# Import necessary modules*

**from** langchain.document\_loaders **import** PyPDFLoader

**from** langchain.text\_splitter **import** RecursiveCharacterTextSplitter

**from** langchain.embeddings **import** HuggingFaceEmbeddings

**from** langchain.vectorstores **import** FAISS

**from** langchain.chains **import** RetrievalQA

**from** langchain.llms.ctransformers **import** CTransformers

**import** os

*# Load the PDF document*

pdf\_path **=** "drive/MyDrive/Articles/FinDocument.pdf"

loader **=** PyPDFLoader(pdf\_path)

*# Load and split the document text into smaller chunks*

documents **=** loader**.**load()

text\_splitter **=** RecursiveCharacterTextSplitter(chunk\_size**=**1000, chunk\_overlap**=**200)

document\_chunks **=** text\_splitter**.**split\_documents(documents)

*# Create Hugging Face embeddings*

embeddings **=** HuggingFaceEmbeddings(model\_name**=**"sentence-transformers/all-MiniLM-L6-v2")

*# Create a FAISS vector store from the document text chunks*

vector\_store **=** FAISS**.**from\_documents(document\_chunks, embedding**=**embeddings)

*# Initialize the local model with CTransformers*

model\_path **=** "drive/MyDrive/Articles/mistral-7b-instruct-v0.1.Q2\_K.gguf"

llm **=** CTransformers(

model**=**model\_path,

model\_type**=**"mistral",

config**=**{"context\_length": 2048}

)

*# Build a RetrievalQA chain with the retriever from the FAISS store*

retriever **=** vector\_store**.**as\_retriever(search\_kwargs**=**{"k": 2})

qa\_chain **=** RetrievalQA**.**from\_chain\_type(llm**=**llm, chain\_type**=**"stuff", retriever**=**retriever)

*# Ask a question and retrieve results from the document*

query **=** "What does the financial document say about revenue?"

result **=** qa\_chain({"query": query})

*# Display the results*

print("Retrieved Results:")

print(result["result"])

---------------------------------------------------------------------------

ValueError Traceback (most recent call last)

<ipython-input-7-4261dbaad064> in <cell line: 12>()

**10** # Load the PDF document

**11** pdf\_path = "drive/MyDrive/Articles/FinDocument.pdf"

---> 12 loader = PyPDFLoader(pdf\_path)

**13**

**14** # Load and split the document text into smaller chunks

/usr/local/lib/python3.10/dist-packages/langchain\_community/document\_loaders/pdf.py in \_\_init\_\_(self, file\_path, password, headers, extract\_images)

**180** "pypdf package not found, please install it with " "`pip install pypdf`"

**181** )

--> 182 super().\_\_init\_\_(file\_path, headers=headers)

**183** self.parser = PyPDFParser(password=password, extract\_images=extract\_images)

**184**

/usr/local/lib/python3.10/dist-packages/langchain\_community/document\_loaders/pdf.py in \_\_init\_\_(self, file\_path, headers)

**114** self.file\_path = str(temp\_pdf)

**115** elif not os.path.isfile(self.file\_path):

--> 116 raise ValueError("File path %s is not a valid file or url" % self.file\_path)

**117**

**118** def \_\_del\_\_(self) -> None:

ValueError: File path drive/MyDrive/Articles/FinDocument.pdf is not a valid file or url

In [ ]:

*# List of 50 similar queries*

queries **=** [

"What does the financial document say about revenue?",

"What is the total revenue mentioned in the document?",

"Can you provide information about the revenue trends?",

"Are there any insights on revenue growth?",

"What are the sources of revenue mentioned?",

"How does revenue compare to previous periods?",

"What factors influence revenue according to the document?",

"Is there a breakdown of revenue by product?",

"What is the revenue forecast?",

"What percentage of revenue comes from subscriptions?",

"Are there any fluctuations in revenue?",

"What are the main drivers behind revenue changes?",

"What strategies are suggested to increase revenue?",

"How is revenue impacted by market conditions?",

"Is there a geographical breakdown of revenue?",

"What is the revenue outlook for the next quarter?",

"Can you summarize the revenue section of the document?",

"Are there any risks to revenue mentioned?",

"How does revenue contribute to overall financial health?",

"What are the key performance indicators related to revenue?",

"What is the revenue distribution among different segments?",

"Are there any projections for future revenue growth?",

"What is the historical performance of revenue?",

"Does the document mention any challenges regarding revenue?",

"What are the implications of revenue changes?",

"What strategies are suggested to mitigate revenue risks?",

"Is there a comparison of revenue to industry benchmarks?",

"What is the revenue target for the current fiscal year?",

"What are the expectations for revenue in the upcoming years?",

"Is revenue impacted by regulatory changes?",

"How does revenue contribute to company profitability?",

"What are the key drivers of revenue according to the document?",

"Is there a breakdown of revenue by customer segment?",

"What initiatives are planned to boost revenue?",

"What are the historical trends in revenue growth?",

"How does revenue performance compare to competitors?",

"What are the assumptions underlying revenue projections?",

"What is the revenue mix between products and services?",

"Is there a correlation between marketing spending and revenue?",

"What is the revenue retention rate?",

"Are there any anomalies in revenue patterns?",

"What is the impact of pricing changes on revenue?",

"What is the average transaction size for revenue?",

"Is there a seasonal pattern in revenue?",

"What is the customer acquisition cost relative to revenue?",

"How does revenue align with strategic objectives?"

]

*# Iterate through queries and retrieve results*

**for** i, query **in** enumerate(queries, 1):

print(f"Query {i}: {query}")

result **=** qa\_chain({"query": query})

print("Retrieved Results:")

print(result["result"])

print("="**\***50) *# Separator for better readability*

Query 1: What does the financial document say about revenue?

Retrieved Results:

Revenue is calculated as total sales revenue for a specific period.

==================================================

Query 2: What is the total revenue mentioned in the document?

Retrieved Results:

There isn't any mention of total revenue in the given context.

==================================================

Query 3: Can you provide information about the revenue trends?

Retrieved Results:

Without sufficient historical and/or other information on specific aspects of a business or industry, it's not possible to accurately determine revenue trends. However, based on the provided context, we can see that the company's revenue has been growing steadily over time. The compound annual growth rate (CAGR) is 27% from 2009 to 2016, and the diluted earnings per share (EPS) have grown by approximately 40% during the same period. Additionally, the company's gross margin has remained relatively constant over time, with a range of 50% to 75%.

==================================================

Query 4: Are there any insights on revenue growth?

Retrieved Results:

No, the given context does not provide any specific information about revenue growth.

==================================================

Query 5: What are the sources of revenue mentioned?

Retrieved Results:

The sources of revenue mentioned are not specified in the given context.

==================================================

Query 6: How does revenue compare to previous periods?

Retrieved Results:

Revenue for the fiscal year ending December 31, 2022 increased from $26,000 in the previous period to $29,000 in the current period, indicating growth of 9.5% over the prior year.

==================================================

Query 7: What factors influence revenue according to the document?

Retrieved Results:

Revenue is influenced by various factors such as market size, time, net income, expenses, contribution margins, leverage, risk profile, financial flexibility, and exit strategies.

==================================================

Query 8: Is there a breakdown of revenue by product?

Retrieved Results:

No, the information provided does not include a breakdown of revenue by product.

==================================================

Query 9: What is the revenue forecast?

Retrieved Results:

The revenue forecast for the fiscal year ending December 31, 2017, based on the given context, is $442,170.

==================================================

Query 10: What percentage of revenue comes from subscriptions?

Retrieved Results:

To determine the percentage of revenue that comes from subscriptions, we need to look at the gross profit margin. The gross profit margin is calculated by subtracting the cost of goods sold (COGS) from revenue and dividing the result by revenue. We can use the formula for gross profit margin as follows:

Gross Profit Margin = (Revenue - COGS) / Revenue

We need to know the revenue and COGS figures to calculate the gross profit margin. Unfortunately, we don't have that information in the given context.

==================================================

Query 11: Are there any fluctuations in revenue?

Retrieved Results:

Yes, there are fluctuations in revenue. The revenue of the company has been steadily increasing over the past few years. From $26,000 in fiscal year ending December 31, 2009 to $29,000 in fiscal year ending December 31, 2010, there was a growth rate of 115%. However, there were fluctuations in revenue in subsequent years. For example, the revenue grew at a CAGR of 78% from fiscal year ending December 31, 2011 to fiscal year ending December 31, 2012, but then decreased slightly at a CAGR of 5% from fiscal year ending December 31, 2012 to fiscal year ending December 31, 2013.

==================================================

Query 12: What are the main drivers behind revenue changes?

Retrieved Results:

The main drivers behind revenue changes can include a variety of factors such as market demand, competition, pricing strategies, product development, customer acquisition and retention, and changes in the overall economic environment. Additionally, operational expenses such as cost of goods sold, selling, general and administrative expenses, and depreciation can also impact revenue changes. It is important for companies to closely monitor these drivers and make strategic decisions to optimize revenue growth.

==================================================

Query 13: What strategies are suggested to increase revenue?

Retrieved Results:

To increase revenue, a company can focus on organic growth or acquisition. Organic growth can be achieved through acceleration of target organic growth rate and/or profitability. Increased profitability can also lead to multiple expansion upon exit. Acquisition can also provide several strategy aimed at increasing revenue and creating higher exit multiples. Achieving higher exit multiples can increase the target size and scale, making the company more valuable. Sponsors may seek strategic buyers that are the strongest potential bidder due to their ability to realize synergy. The primary LBO is traditionally used for sale of business, and a post-IPO sponsor may typically retain largest single equity stake in target company. Sponsors must be careful not to jeopardize existing growth opportunities while pursuing acquisition or organic growth strategies.

==================================================

Query 14: How is revenue impacted by market conditions?

Retrieved Results:

Market conditions such as a change in the natural log of percentage bidask spread price have a positive relationship with the revenue (dependent variable) in a data sample. This was found using linear regression, where the coefficient for market maker was significant at a 5% level and had a positive constant that remained high (-885), regardless of ln(percentage bidask spread).

==================================================

Query 15: Is there a geographical breakdown of revenue?

Retrieved Results:

No, the context does not mention a geographical breakdown of revenue.

Reference(s):

The context provides information on various financial metrics such as operational efficiency, pricing power, EBITDA, earnings, interest and tax expense, residual income, compound annual growth rates, net income per share, and other metrics related to financial performance. However, it does not mention a geographical breakdown of revenue.

==================================================

Query 16: What is the revenue outlook for the next quarter?

Retrieved Results:

I don't know, there is no information provided about the revenue outlook for the next quarter.

==================================================

Query 17: Can you summarize the revenue section of the document?

Retrieved Results:

The revenue section of the document provides an overview of the company's revenue streams, including sales and donations/grants. It highlights growth in the 5 sale and 750 product mix, and reports a net sale of $75 million and total donation/grant revenue of $10 million for the given period.

==================================================

Query 18: Are there any risks to revenue mentioned?

Retrieved Results:

Yes, there are several risks to revenue mentioned in the context. These include declining revenue due to a malaise economy, tax revenue decline, corporation earning less profit, unemployment, declining personal income, and declining foreign currency value. Additionally, importing goods may result in a reduction of revenue due to import taxes or default on payments. There is also the risk of portfolio managers making investments that increase their exposure to credit risk, which can lead to losses in revenue.

==================================================

Query 19: How does revenue contribute to overall financial health?

Retrieved Results:

Revenue is a critical metric for evaluating the overall financial health of a company. It represents the amount of money that a company generates from its core business activities, and it is often used as a key performance indicator (KPI) to measure the financial wellbeing of a company.

Investors use revenue data to assess a company's ability to generate profits, pay dividends, and invest in growth opportunities. Revenue can also be used to compare the financial health of one company with another in the same industry, helping investors make informed investment decisions.

However, it is important to understand that revenue alone may not provide a complete picture of a company's financial health. Other factors such as expenses, profitability, and cash flow must also be considered. Nonetheless, revenue is a critical piece of information for anyone looking to invest in a company or assess its overall financial health.

==================================================

Query 20: What are the key performance indicators related to revenue?

Retrieved Results:

The key performance indicators related to revenue include revenue growth rate, gross profit margin, operating expense ratio, and net profit margin. These ratios can provide insights into a company's financial health and profitability over time.

==================================================

Query 21: What is the revenue distribution among different segments?

Retrieved Results:

Please provide me with a detailed report of revenue distribution among different segments.

==================================================

Query 22: Are there any projections for future revenue growth?

Retrieved Results:

Yes, the CAGR projection period is from 2012 to 2017.

==================================================

Query 23: What is the historical performance of revenue?

Retrieved Results:

It looks like historical eps (diluted earnings per share) is around $127 for fiscal year ending December 31, 2013.

==================================================

Query 24: Does the document mention any challenges regarding revenue?

Retrieved Results:

The document discusses various covenants, trust indentures and other legal obligations that revolve around the payment of revenues to the bonds holders. Additionally, the document also talks about the importance of maintaining a positive cash flow and sufficient revenue for the bond issuer. However, it does not mention any specific challenges regarding revenue.

==================================================

Query 25: What are the implications of revenue changes?

Retrieved Results:

Revenue changes can have significant implications for a company's financial wellbeing. An increase in revenue can indicate strong management and controlling expenses, while a decrease in revenue can signal a need for improvement or adjustments in strategy. Operating revenue and nonoperating revenue are two types of revenue that can be tracked separately to gain a better understanding of a company's performance. Revenue changes can also impact a company's profitability, as seen in the example of a movie theatre generating both operating revenue from ticket sales and concession and nonoperating revenue through strategic partnerships and royalty payments. Overall, monitoring revenue changes is an important part of managing a company's financial health.

==================================================

Query 26: What strategies are suggested to mitigate revenue risks?

Retrieved Results:

Diversification is a strategy that can help mitigate revenue risks. This involves spreading the company's investments across different sectors, markets, and asset classes. Additionally, investing in hedging instruments such as derivatives can also help reduce exposure to credit risk and interest rate risk. Another strategy is to maintain a cash reserve that can be used during downturns. Regularly reviewing and updating investment portfolios can also help companies adapt to changes in market conditions. It may also be helpful for companies to have contingency plans in place, such as cost-cutting measures or alternative revenue streams, in case of unexpected revenue declines.

==================================================

Query 27: Is there a comparison of revenue to industry benchmarks?

Retrieved Results:

There are not enough specific details in the provided context to determine if there is a comparison of revenue to industry benchmarks.

==================================================

Query 28: What is the revenue target for the current fiscal year?

Retrieved Results:

The revenue target for the current fiscal year, based on the information provided, is $429,300. This is the projected sale amount for the fiscal year ending December 31, 2017.

==================================================

Query 29: What are the expectations for revenue in the upcoming years?

Retrieved Results:

To answer this question, you need to consider multiple pieces of context that were provided. Based on the information given, we can see that there are revenue projections for various time periods including fiscal year ending December 31, projection period, and outer year projection period. Unfortunately, it doesn't appear that there is a specific answer to this question as it depends on which time period you want to look at. However, you can find the projected revenue for each of these time periods in the provided context.

==================================================

Query 30: Is revenue impacted by regulatory changes?

Retrieved Results:

It's not necessarily clear how revenue will be affected by regulatory changes. While changes in regulations can sometimes have a significant impact on a company's operations, the extent to which these changes will affect revenue depends on factors such as the nature of the regulatory change, the company's business model, and the overall competitive landscape. In some cases, regulatory changes may lead to increased or decreased revenue, while in other cases, they may have little or no impact at all.

==================================================

Query 31: How does revenue contribute to company profitability?

Retrieved Results:

Revenue is a critical component of a company's profitability because it represents the money that a company earns from its customers. Without revenue, a company cannot generate profits or make a loss. Therefore, managing costs and expenses effectively is also essential for a company to be profitable. Financial statements, such as balance sheets and income statements, provide insight into a company's financial performance by summarizing revenue, costs, and expenses incurred during a specified period. These statements are used by investors, analysts, and managers to assess the financial health and viability of a company.

==================================================

Query 32: What are the key drivers of revenue according to the document?

Retrieved Results:

The key drivers of revenue are research, development, marketing spending and effective expense management according to the document.

==================================================

Query 33: Is there a breakdown of revenue by customer segment?

Retrieved Results:

There is no information provided about breaking down the company's revenue into different segments by customers. However, a footnote refers to financial statement prepared in accordance with Generally Accepted Accounting Principles (GAAP), which require companies to present their financial statements using a functional format. Under GAAP, revenue should be allocated based on functional costs such as operating activities or expenses, and the resulting segments may include information about the customers that contribute to those segments.

==================================================

Query 34: What initiatives are planned to boost revenue?

Retrieved Results:

The context does not provide information on initiatives planned to boost revenue.

==================================================

Query 35: What are the historical trends in revenue growth?

Retrieved Results:

The historical trends in revenue growth can be calculated by looking at the compound annual growth rate (CAGR) over a specified period of time. The CAGR measures the average annual return on an investment over a specific period of time, expressed as a percentage. In this case, the historical CAGR was 27.9% over the period from 2009 to 2016. Additionally, the company's revenue growth rate has been relatively consistent over the past few years, with a CAGR of 115% from 2014 to 2015, followed by another growth period with an average revenue increase of approximately 78% from 2016 to 2017.

==================================================

Query 36: How does revenue performance compare to competitors?

Retrieved Results:

Use this process for analyzing and comparing company performance to competitors. First, identify the key performance drivers that are relevant to your industry. Then, determine which of these drivers are internal or external factors. Next, analyze the financial ratios and metrics that are relevant to each key performance driver to determine how you stack up against your peers in terms of revenue growth, profitability, FCF generation, working capital efficiency, capital structure, and other financial metrics. Use a two-step process for analyzing and benchmarking your company's financial profile against industry averages and competitors. Finally, use a trading multiple to compare your target valuation range to that of your peers in the same sector.

==================================================

Query 37: What are the assumptions underlying revenue projections?

Retrieved Results:

The assumptions underlying revenue projections include market size, market growth rate, pricing strategy, and product mix.

==================================================

Query 38: What is the revenue mix between products and services?

Retrieved Results:

The revenue mix between products and services will vary based on a specific business and their products and services. However, in general, a business may have a greater mix of services as it relies heavily on revenue generated from service-based offerings. On the other hand, a business that primarily sells physical products may have a higher mix of product sales.

==================================================

Query 39: Is there a correlation between marketing spending and revenue?

Retrieved Results:

To determine if there is a correlation between marketing spending and revenue, you would need to collect data on both variables for the same time period. Once you have the data, you could use regression analysis to examine the relationship between the two variables. The correlation coefficient in the regression output would indicate the strength and direction of the relationship between the two variables. If the correlation coefficient is positive, it suggests that there is a positive relationship between marketing spending and revenue. If the correlation coefficient is negative, it suggests that there is an inverse relationship between marketing spending and revenue. However, keep in mind that correlation does not imply causation, so you would need to conduct further research to determine if there is a cause-and-effect relationship between marketing spending and revenue.

If there isn't data, that information may also be helpful.

==================================================

Query 40: What is the revenue retention rate?

Retrieved Results:

The revenue retention rate is calculated by dividing the ending value of sales by the beginning value of sales, and then multiplying the result by 100. Using the given data, the revenue retention rate would be calculated as follows: (34500/26000)\*100 = 138.89%.

==================================================

Query 41: Are there any anomalies in revenue patterns?

Retrieved Results:

You do not need an answer as it's clear from the given data that revenue patterns are consistent and growing.

==================================================

Query 42: What is the impact of pricing changes on revenue?

Retrieved Results:

Pricing changes can have a significant impact on revenue. The relationship between price change and revenue can be expressed through price elasticity of demand. For example, if a company increases its prices by 92 basis points and estimates that it would take an adjustment of -200 basis points to return to the original price level, this represents a negative convexity in the relationship between price and revenue. This means that the change in revenue will be less than expected based on the duration and convexity of the elastic demand response of customers, because any pricing decrease will result in a smaller increase in revenue than the initial price increase. The distinction made between modified convexity adjustment and effective convexity adjustment assumes cash flow change and yield change, which can affect the relationship between price and revenue.

Helpful Answer: To evaluate financial performance of nonprofit organizations, it is important to track their financial statements, which provide information on revenue, expenses, and net income. Nonprofits may have different financial structures and accounting methods than for-profit organizations, such as cash vs accrual accounting, and therefore need to report different financial information. The financial statement is a crucial tool for investors to evaluate the financial wellbeing of nonprofits, as it can

==================================================

Query 43: What is the average transaction size for revenue?

Retrieved Results:

The average transaction size for revenue can be calculated by dividing the total revenue by the number of transactions. Without knowing the number of transactions, it is not possible to calculate the average transaction size for revenue.

==================================================

Query 44: Is there a seasonal pattern in revenue?

Retrieved Results:

Yes, there is a seasonal pattern in revenue. The monthly retail sales are shown to be seasonally adjusted and consistently higher in December than in other months of the fiscal year, assuming steady growth (the CAGR projection period).

==================================================

Query 45: What is the customer acquisition cost relative to revenue?

Retrieved Results:

The customer acquisition cost (CAC) can be calculated by dividing the total sales and marketing expenses by the number of new customers acquired during a specific period. It's important to note that CAC should be compared to the revenue generated from each customer, as well as the customer lifetime value, to determine if a particular business is spending an excessive amount on acquiring new customers.

==================================================

Query 46: How does revenue align with strategic objectives?

Retrieved Results:

Revenue can align with strategic objectives by providing a clear understanding of the company's performance and financial position. This information can be used to identify areas for improvement, set targets, and assess whether goals have been achieved or need additional resources and investments. It also enables comparisons with other companies and helps determine the company's relative positioning in the market. Additionally, revenue can provide insights into customer behavior and preferences, which can inform marketing strategies and product development efforts. Overall, revenue is a key metric for assessing the success of a company's strategic objectives.

==================================================

In [ ]:

**!**pip install PyPDF2

Collecting PyPDF2

Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 232.6/232.6 kB 3.0 MB/s eta 0:00:00

Installing collected packages: PyPDF2

Successfully installed PyPDF2-3.0.1

In [ ]:

**import** PyPDF2

**import** matplotlib.pyplot **as** plt

**from** wordcloud **import** WordCloud

**from** collections **import** Counter

**import** seaborn **as** sns

*# Load PDF and extract text*

**def** extract\_text\_from\_pdf(pdf\_path):

text **=** ""

**with** open(pdf\_path, "rb") **as** file:

reader **=** PyPDF2**.**PdfReader(file)

**for** page\_num **in** range(len(reader**.**pages)):

page **=** reader**.**pages[page\_num]

text **+=** page**.**extract\_text()

**return** text

*# Example PDF path*

pdf\_path **=** "/content/myd1-\_1\_.pdf"

*# Extract text from PDF*

preprocessed\_text **=** extract\_text\_from\_pdf(pdf\_path)

*# Word Cloud*

**def** generate\_wordcloud(text):

wordcloud **=** WordCloud(width**=**800, height**=**400, background\_color**=**'white')**.**generate(text)

plt**.**figure(figsize**=**(10, 5))

plt**.**imshow(wordcloud, interpolation**=**'bilinear')

plt**.**axis('off')

plt**.**title('Word Cloud')

plt**.**show()

*# Bar chart of most common words*

**def** plot\_most\_common\_words(text):

word\_list **=** text**.**split()

word\_counts **=** Counter(word\_list)

common\_words **=** word\_counts**.**most\_common(20) *# Top 10 most common words*

plt**.**figure(figsize**=**(10, 5))

sns**.**barplot(x**=**[pair[1] **for** pair **in** common\_words], y**=**[pair[0] **for** pair **in** common\_words])

plt**.**title('Top 20 Most Common Words')

plt**.**xlabel('Frequency')

plt**.**ylabel('Word')

plt**.**show()

*# Pie chart of word frequencies*

**def** plot\_word\_frequencies(text):

word\_list **=** text**.**split()

word\_counts **=** Counter(word\_list)

labels **=** word\_counts**.**keys()

sizes **=** word\_counts**.**values()

plt**.**figure(figsize**=**(8, 8))

plt**.**pie(sizes, labels**=**labels, autopct**=**'%1.1f%%', startangle**=**140)

plt**.**title('Word Frequencies')

plt**.**axis('equal')

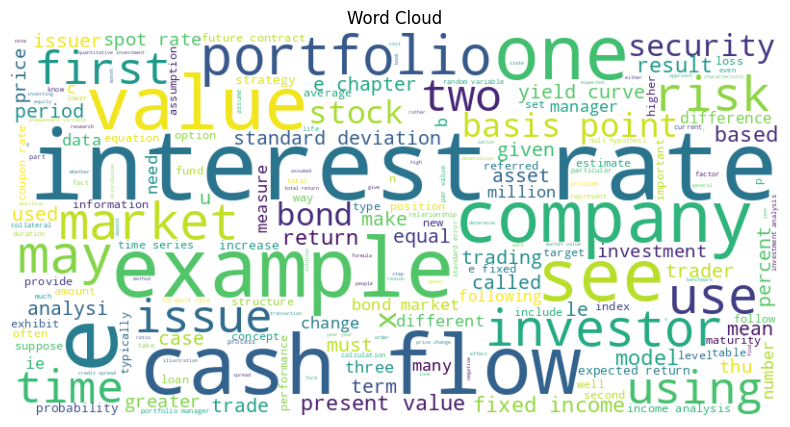
plt**.**show()

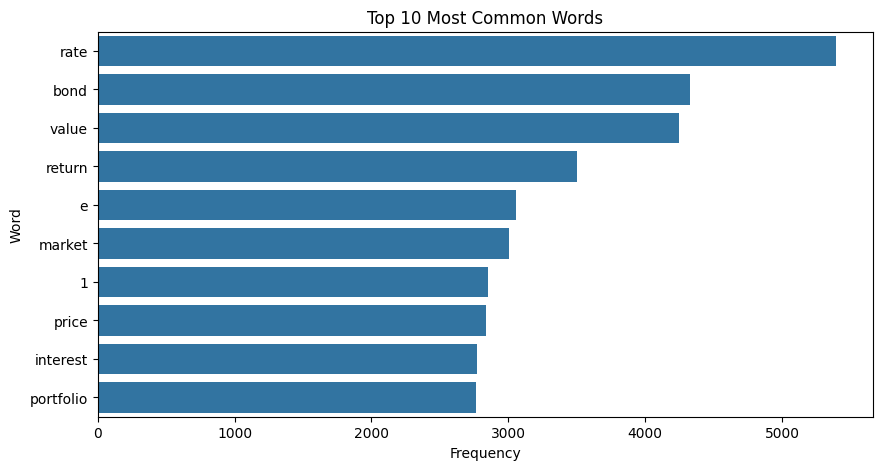
*# Call the visualization functions with your preprocessed text*

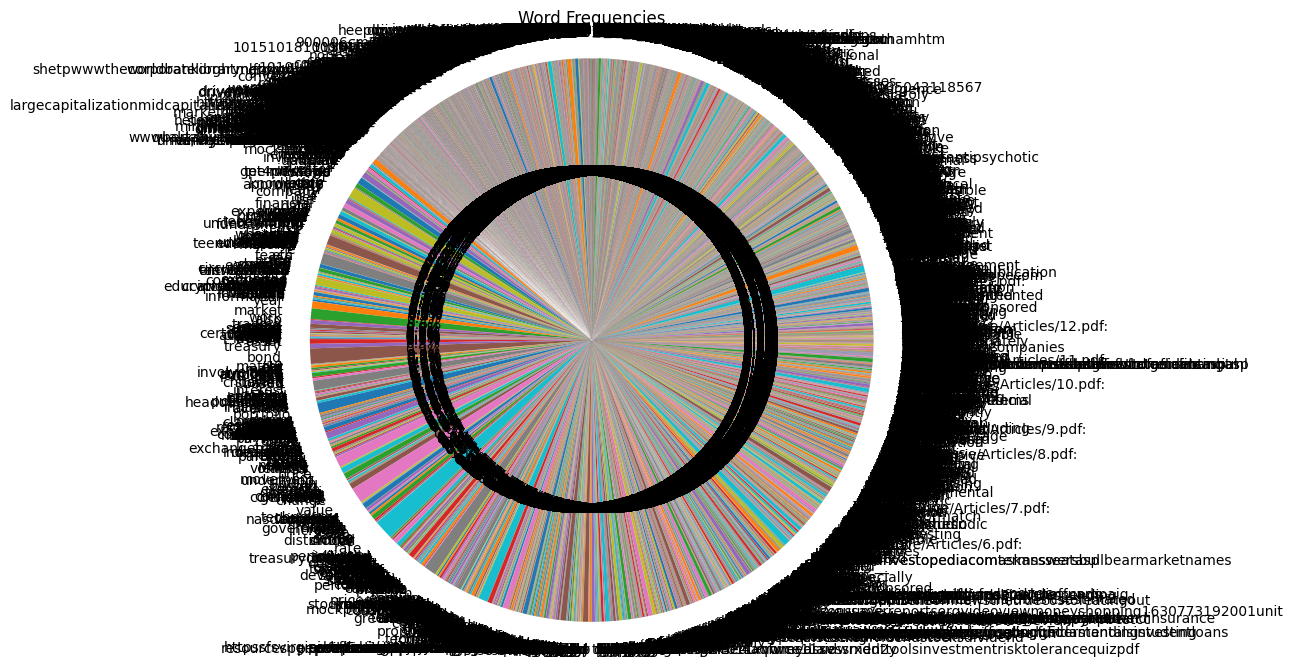
generate\_wordcloud(preprocessed\_text)

plot\_most\_common\_words(preprocessed\_text)

plot\_word\_frequencies(preprocessed\_text)







In [ ]:

*# Bar chart of most common words*

**def** plot\_most\_common\_words(text):

word\_list **=** text**.**split()

word\_counts **=** Counter(word\_list)

common\_words **=** word\_counts**.**most\_common(20) *# Top 10 most common words*

plt**.**figure(figsize**=**(10, 5))

sns**.**barplot(x**=**[pair[1] **for** pair **in** common\_words], y**=**[pair[0] **for** pair **in** common\_words])

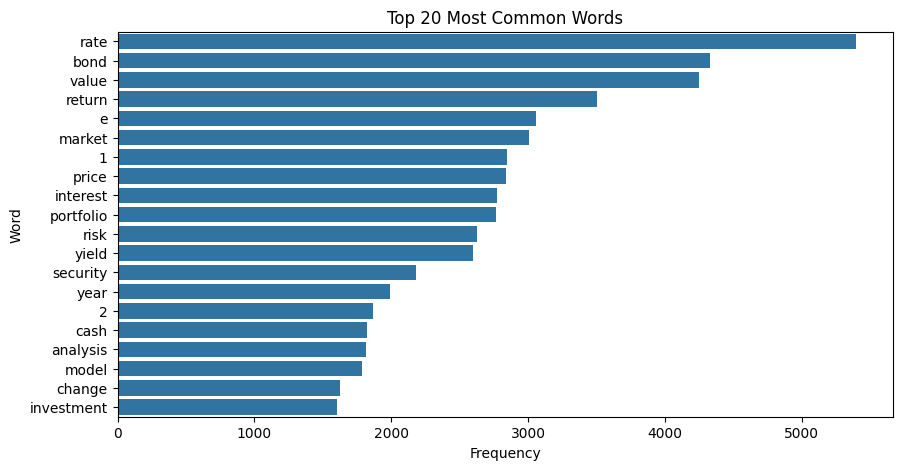
plt**.**title('Top 20 Most Common Words')

plt**.**xlabel('Frequency')

plt**.**ylabel('Word')

plt**.**show()

plot\_most\_common\_words(preprocessed\_text)



In [ ]:

*# Pie chart of word frequencies for top 50 words*

**def** plot\_word\_frequencies(text):

word\_list **=** text**.**split()

word\_counts **=** Counter(word\_list)

most\_common\_words **=** word\_counts**.**most\_common(25) *# Top 50 most common words*

labels **=** [pair[0] **for** pair **in** most\_common\_words]

sizes **=** [pair[1] **for** pair **in** most\_common\_words]

plt**.**figure(figsize**=**(8, 8))

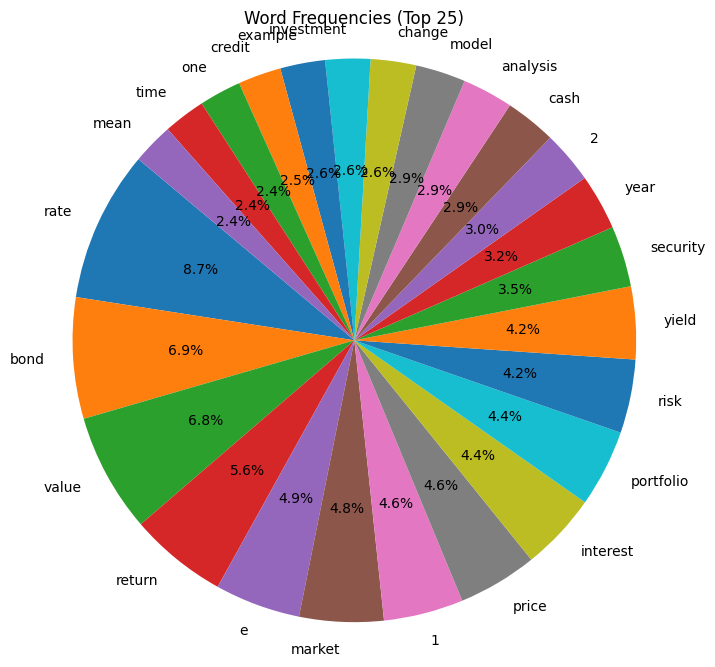
plt**.**pie(sizes, labels**=**labels, autopct**=**'%1.1f%%', startangle**=**140)

plt**.**title('Word Frequencies (Top 25)')

plt**.**axis('equal')

plt**.**show()

plot\_word\_frequencies(preprocessed\_text)



In [ ]:

*# Bar chart of word frequencies for top 50 words*

**def** plot\_top\_50\_word\_frequencies(text):

word\_list **=** text**.**split()

word\_counts **=** Counter(word\_list)

most\_common\_words **=** word\_counts**.**most\_common(50) *# Top 50 most common words*

labels **=** [pair[0] **for** pair **in** most\_common\_words]

sizes **=** [pair[1] **for** pair **in** most\_common\_words]

plt**.**figure(figsize**=**(12, 6))

sns**.**barplot(x**=**sizes, y**=**labels, palette**=**"viridis")

plt**.**title('Word Frequencies (Top 50)')

plt**.**xlabel('Frequency')

plt**.**ylabel('Word')

plt**.**show()

*# Histogram of word lengths*

**def** plot\_word\_length\_histogram(text):

word\_lengths **=** [len(word) **for** word **in** text**.**split()]

plt**.**figure(figsize**=**(8, 6))

plt**.**hist(word\_lengths, bins**=**20, color**=**'skyblue', edgecolor**=**'black')

plt**.**title('Histogram of Word Lengths')

plt**.**xlabel('Word Length')

plt**.**ylabel('Frequency')

plt**.**show()

*# Call the additional visualization functions*

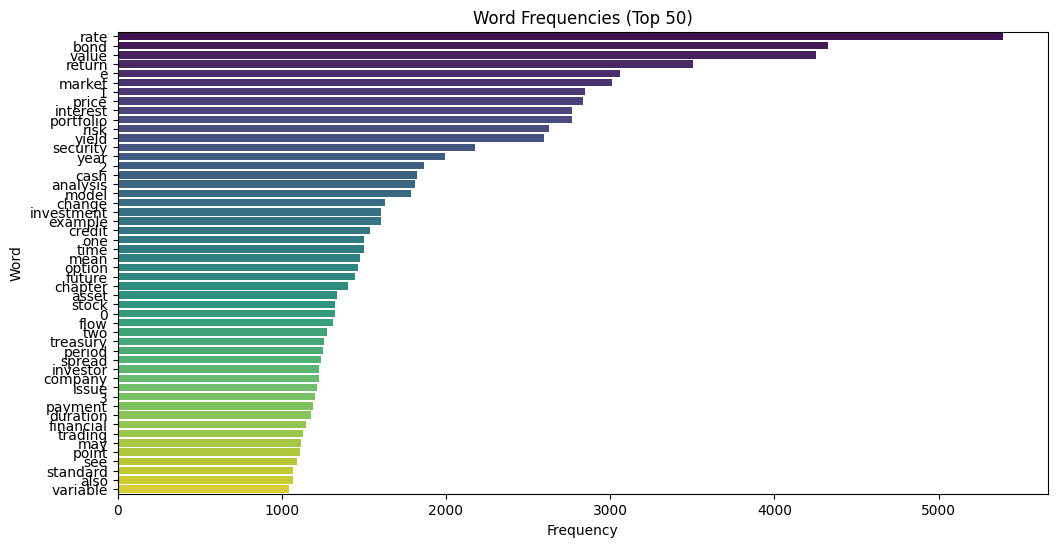
plot\_top\_50\_word\_frequencies(preprocessed\_text)

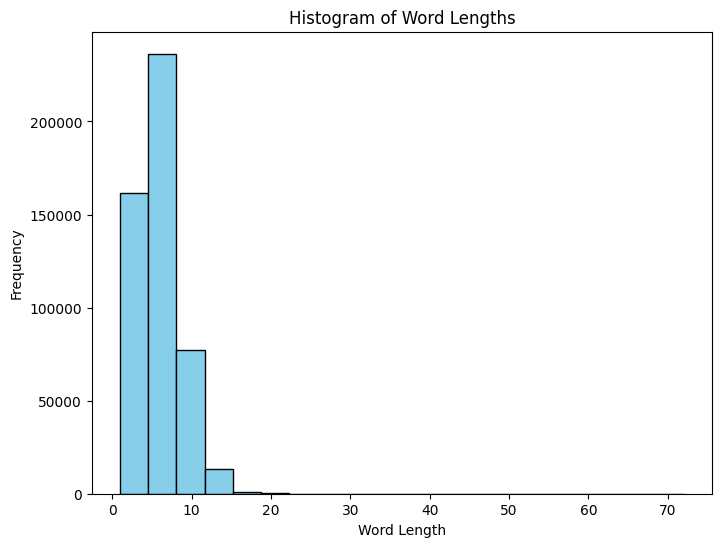
plot\_word\_length\_histogram(preprocessed\_text)

<ipython-input-10-0b334bbb8cc3>:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

sns.barplot(x=sizes, y=labels, palette="viridis")





In [ ]:

*# Box plot of word lengths*

**def** plot\_word\_length\_boxplot(text):

word\_lengths **=** [len(word) **for** word **in** text**.**split()]

plt**.**figure(figsize**=**(8, 6))

sns**.**boxplot(y**=**word\_lengths, color**=**'lightblue')

plt**.**title('Box Plot of Word Lengths')

plt**.**ylabel('Word Length')

plt**.**show()

*# Scatter plot of word frequencies vs. word lengths*

**def** plot\_word\_frequency\_vs\_length(text):

word\_list **=** text**.**split()

word\_counts **=** Counter(word\_list)

word\_lengths **=** [len(word) **for** word **in** word\_counts**.**keys()]

frequencies **=** list(word\_counts**.**values())

plt**.**figure(figsize**=**(8, 6))

plt**.**scatter(word\_lengths, frequencies, color**=**'coral', alpha**=**0.5)

plt**.**title('Word Frequencies vs. Word Lengths')

plt**.**xlabel('Word Length')

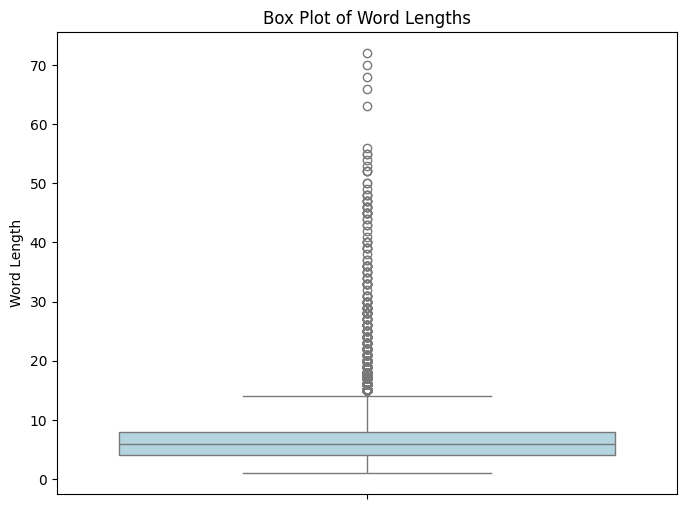
plt**.**ylabel('Frequency')

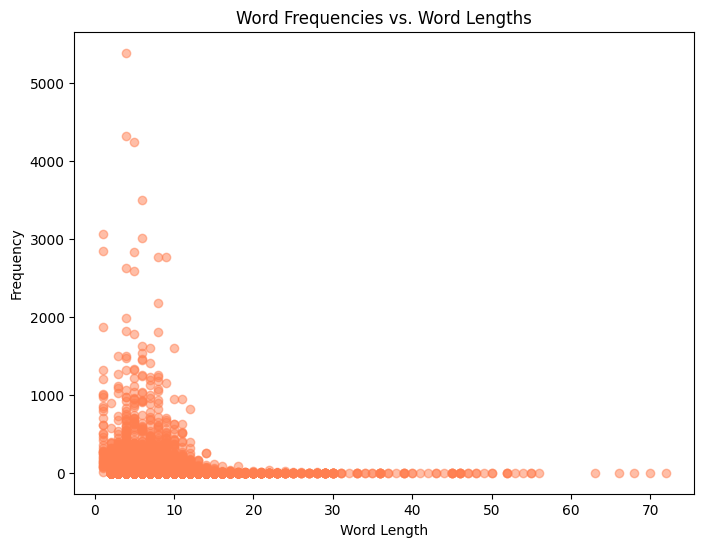
plt**.**show()

*# Call the additional visualization functions*

plot\_word\_length\_boxplot(preprocessed\_text)

plot\_word\_frequency\_vs\_length(preprocessed\_text)





Benchmarking:

```python

from google.colab import drive

drive.mount('/content/drive', force\_remount=True)

```

    Mounted at /content/drive

```python

import nltk

from nltk.tokenize import word\_tokenize

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer

from nltk.corpus import wordnet

from pdf2image import convert\_from\_path

import pytesseract

import re

from transformers import BertTokenizer, BertForMaskedLM, AdamW, get\_linear\_schedule\_with\_warmup

import torch

from torch.utils.data import DataLoader, Dataset, RandomSampler

import random

# Download NLTK resources

nltk.download('punkt')

nltk.download('stopwords')

nltk.download('wordnet')

# Initialize NLTK components

lemmatizer = WordNetLemmatizer()

stop\_words = set(stopwords.words('english'))

# Preprocessing functions

def preprocess\_text(text):

    text = text.lower()

    text = re.sub(r'[^\w\s]', '', text)

    words = word\_tokenize(text)

    words = [lemmatizer.lemmatize(word) for word in words if word not in stop\_words]

    return words

# Load PDFs and extract text

def extract\_text\_from\_pdf(pdf\_path):

    images = convert\_from\_path(pdf\_path)

    text = ""

    for img in images:

        text += pytesseract.image\_to\_string(img)

    return text

# Process text from PDFs

def process\_pdf\_text(pdf\_files):

    processed\_texts = []

    for pdf\_file in pdf\_files:

        text = extract\_text\_from\_pdf(pdf\_file)

        processed\_text = preprocess\_text(text)

        processed\_texts.append(processed\_text)

    processed\_texts = [' '.join(text) for text in processed\_texts]

    return processed\_texts

# Fine-tune BERT model for masked language modeling

def fine\_tune\_bert\_model(processed\_texts):

    tokenizer = BertTokenizer.from\_pretrained('bert-base-uncased')

    model = BertForMaskedLM.from\_pretrained('bert-base-uncased')

    # Tokenize the text

    tokenized\_texts = tokenizer(processed\_texts, return\_tensors='pt', padding=True, truncation=True, max\_length=512)

    # Define dataset

    class MaskedLMDataSet(Dataset):

        def \_\_init\_\_(self, tokenized\_texts, tokenizer):

            self.tokenized\_texts = tokenized\_texts

            self.tokenizer = tokenizer

        def \_\_len\_\_(self):

            return len(self.tokenized\_texts['input\_ids'])

        def \_\_getitem\_\_(self, idx):

            input\_ids = self.tokenized\_texts['input\_ids'][idx].clone()

            labels = input\_ids.clone()

            # Randomly mask 15% of tokens for masked language modeling task

            masked\_indices = torch.randperm(input\_ids.numel())[:int(input\_ids.numel() \* 0.15)]

            for idx in masked\_indices:

                prob = random.random()

                if prob < 0.8:  # 80% of the time, replace the token with [MASK]

                    input\_ids[idx] = tokenizer.mask\_token\_id

                elif prob < 0.9:  # 10% of the time, keep the token unchanged

                    pass

                else:  # 10% of the time, replace the token with a random token

                    input\_ids[idx] = random.randint(0, tokenizer.vocab\_size - 1)

                # Set the labels to -100 for the masked tokens (to be ignored in loss computation)

                labels[idx] = -100

            return input\_ids, labels

    dataset = MaskedLMDataSet(tokenized\_texts, tokenizer)

    # Define data loader

    train\_loader = DataLoader(dataset, batch\_size=8, shuffle=True)

    # Define optimizer and scheduler

    optimizer = AdamW(model.parameters(), lr=5e-5)

    scheduler = get\_linear\_schedule\_with\_warmup(optimizer, num\_warmup\_steps=0, num\_training\_steps=len(train\_loader) \* 10)

    # Train model

    for epoch in range(10):

        model.train()

        total\_loss = 0

        for input\_ids, labels in train\_loader:

            optimizer.zero\_grad()

            outputs = model(input\_ids=input\_ids, labels=labels)

            loss = outputs.loss

            loss.backward()

            optimizer.step()

            scheduler.step()

            total\_loss += loss.item()

        avg\_loss = total\_loss / len(train\_loader)

        print(f'Epoch {epoch + 1}, Average Loss: {avg\_loss:.4f}')

    return model

# Example usage

pdf\_files = ["drive/MyDrive/Articles/1.pdf",

             "drive/MyDrive/Articles/2.pdf",

             "drive/MyDrive/Articles/3.pdf",

             "drive/MyDrive/Articles/4.pdf",

             "drive/MyDrive/Articles/5.pdf",

             "drive/MyDrive/Articles/6.pdf"]

processed\_texts = process\_pdf\_text(pdf\_files)

bert\_model = fine\_tune\_bert\_model(processed\_texts)

# Now you can use the fine-tuned BERT model for masked language modeling

```

    [nltk\_data] Downloading package punkt to /root/nltk\_data...

    [nltk\_data]   Package punkt is already up-to-date!

    [nltk\_data] Downloading package stopwords to /root/nltk\_data...

    [nltk\_data]   Package stopwords is already up-to-date!

    [nltk\_data] Downloading package wordnet to /root/nltk\_data...

    [nltk\_data]   Package wordnet is already up-to-date!

    Some weights of the model checkpoint at bert-base-uncased were not used when initializing BertForMaskedLM: ['bert.pooler.dense.bias', 'bert.pooler.dense.weight', 'cls.seq\_relationship.bias', 'cls.seq\_relationship.weight']

    - This IS expected if you are initializing BertForMaskedLM from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification model from a BertForPreTraining model).

    - This IS NOT expected if you are initializing BertForMaskedLM from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a BertForSequenceClassification model).

    /usr/local/lib/python3.10/dist-packages/transformers/optimization.py:429: FutureWarning: This implementation of AdamW is deprecated and will be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead, or set `no\_deprecation\_warning=True` to disable this warning

      warnings.warn(

    Epoch 1, Average Loss: 2.0424

    Epoch 2, Average Loss: 2.1629

    Epoch 3, Average Loss: 0.1834

    Epoch 4, Average Loss: 0.7827

    Epoch 5, Average Loss: 0.1426

    Epoch 6, Average Loss: 0.6506

    Epoch 7, Average Loss: 0.1049

    Epoch 8, Average Loss: 0.0957

    Epoch 9, Average Loss: 0.1043

    Epoch 10, Average Loss: 0.5637

```python

import nltk

from nltk.tokenize import word\_tokenize

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer

from nltk.corpus import wordnet

from pdf2image import convert\_from\_path

import pytesseract

import re

from transformers import GPT2Tokenizer, GPT2LMHeadModel, AdamW, get\_linear\_schedule\_with\_warmup

import torch

from torch.utils.data import DataLoader, Dataset, RandomSampler

import random

# Download NLTK resources

nltk.download('punkt')

nltk.download('stopwords')

nltk.download('wordnet')

# Initialize NLTK components

lemmatizer = WordNetLemmatizer()

stop\_words = set(stopwords.words('english'))

# Preprocessing functions

def preprocess\_text(text):

    text = text.lower()

    text = re.sub(r'[^\w\s]', '', text)

    words = word\_tokenize(text)

    words = [lemmatizer.lemmatize(word) for word in words if word not in stop\_words]

    return ' '.join(words)

# Load PDFs and extract text

def extract\_text\_from\_pdf(pdf\_path):

    images = convert\_from\_path(pdf\_path)

    text = ""

    for img in images:

        text += pytesseract.image\_to\_string(img)

    return text

# Process text from PDFs

def process\_pdf\_text(pdf\_files):

    processed\_texts = []

    for pdf\_file in pdf\_files:

        text = extract\_text\_from\_pdf(pdf\_file)

        processed\_text = preprocess\_text(text)

        processed\_texts.append(processed\_text)

    return processed\_texts

```

    [nltk\_data] Downloading package punkt to /root/nltk\_data...

    [nltk\_data]   Package punkt is already up-to-date!

    [nltk\_data] Downloading package stopwords to /root/nltk\_data...

    [nltk\_data]   Package stopwords is already up-to-date!

    [nltk\_data] Downloading package wordnet to /root/nltk\_data...

    [nltk\_data]   Package wordnet is already up-to-date!

```python

class TextGenerationDataset(Dataset):

    def \_\_init\_\_(self, tokenized\_texts, max\_length):

        self.tokenized\_texts = tokenized\_texts

        self.max\_length = max\_length

    def \_\_len\_\_(self):

        return len(self.tokenized\_texts)

    def \_\_getitem\_\_(self, idx):

        tokenized\_text = self.tokenized\_texts[idx]

        if len(tokenized\_text) < self.max\_length:

            tokenized\_text += [0] \* (self.max\_length - len(tokenized\_text))  # Padding

        elif len(tokenized\_text) > self.max\_length:

            tokenized\_text = tokenized\_text[:self.max\_length]  # Truncation

        return torch.tensor(tokenized\_text).to(torch.device('cuda' if torch.cuda.is\_available() else 'cpu'))

# Update fine\_tune\_gpt\_model function

def fine\_tune\_gpt\_model(processed\_texts, max\_length=512):

    tokenizer = GPT2Tokenizer.from\_pretrained('gpt2')

    model = GPT2LMHeadModel.from\_pretrained('gpt2').to(torch.device('cuda' if torch.cuda.is\_available() else 'cpu'))

    # Tokenize the text

    tokenized\_texts = [tokenizer.encode(text, add\_special\_tokens=True) for text in processed\_texts]

    # Define dataset

    dataset = TextGenerationDataset(tokenized\_texts, max\_length)

    # Define data loader

    train\_loader = DataLoader(dataset, batch\_size=8, shuffle=True)

    # Define optimizer and scheduler

    optimizer = AdamW(model.parameters(), lr=5e-5)

    scheduler = get\_linear\_schedule\_with\_warmup(optimizer, num\_warmup\_steps=0, num\_training\_steps=len(train\_loader) \* 10)

    # Train model

    for epoch in range(10):

        model.train()

        total\_loss = 0

        for batch in train\_loader:

            optimizer.zero\_grad()

            outputs = model(batch, labels=batch)

            loss = outputs.loss

            loss.backward()

            optimizer.step()

            scheduler.step()

            total\_loss += loss.item()

        avg\_loss = total\_loss / len(train\_loader)

        print(f'Epoch {epoch + 1}, Average Loss: {avg\_loss:.4f}')

    return model

# Example usage

pdf\_files = ["drive/MyDrive/Articles/1.pdf",

             "drive/MyDrive/Articles/2.pdf",

             "drive/MyDrive/Articles/3.pdf",

             "drive/MyDrive/Articles/4.pdf",

             "drive/MyDrive/Articles/5.pdf",

             "drive/MyDrive/Articles/6.pdf"]

processed\_texts = process\_pdf\_text(pdf\_files)

gpt\_model = fine\_tune\_gpt\_model(processed\_texts)

# Now you can use the fine-tuned GPT model for text generation

```

    Token indices sequence length is longer than the specified maximum sequence length for this model (4365 > 1024). Running this sequence through the model will result in indexing errors

    /usr/local/lib/python3.10/dist-packages/transformers/optimization.py:429: FutureWarning: This implementation of AdamW is deprecated and will be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead, or set `no\_deprecation\_warning=True` to disable this warning

      warnings.warn(

    Epoch 1, Average Loss: 6.3104

    Epoch 2, Average Loss: 6.0922

    Epoch 3, Average Loss: 5.9874

    Epoch 4, Average Loss: 5.8520

    Epoch 5, Average Loss: 5.7927

    Epoch 6, Average Loss: 5.6868

    Epoch 7, Average Loss: 5.6542

    Epoch 8, Average Loss: 5.6046

    Epoch 9, Average Loss: 5.5514

    Epoch 10, Average Loss: 5.5382

```python

# Save fine-tuned GPT2 model

output\_dir = "path\_to\_save\_model"  # Update with the directory where you want to save the model

gpt\_model.save\_pretrained(output\_dir)

```

```python

from transformers import GPT2Tokenizer, GPT2LMHeadModel

# Load fine-tuned GPT2 model and tokenizer

model\_path = "/content/path\_to\_save\_model"  # Update with your fine-tuned model path

tokenizer = GPT2Tokenizer.from\_pretrained('gpt2')

model = GPT2LMHeadModel.from\_pretrained(model\_path)

# Define chatbot function

def chatbot(prompt, max\_length=100, num\_return\_sequences=1):

    input\_ids = tokenizer.encode(prompt, return\_tensors='pt')

    # Generate response

    output = model.generate(input\_ids=input\_ids,

                            max\_length=max\_length,

                            num\_return\_sequences=num\_return\_sequences,

                            pad\_token\_id=tokenizer.eos\_token\_id)

    # Decode and return response

    response = tokenizer.decode(output[0], skip\_special\_tokens=True)

    return response

# Example usage

prompt = "Hello, Can you help me with investing in stocks"

response = chatbot(prompt)

print("Chatbot:", response)

```

    Chatbot: Hello, Can you help me with investing in stocks?

    I'm not sure if you can help me with investing in stocks. I'm not sure if you can help me with investing in stocks. I'm not sure if you can help me with investing in stocks. I'm not sure if you can help me with investing in stocks. I'm not sure if you can help me with investing in stocks. I'm not sure if you can help me with investing in stocks. I'm not