Text Summarization and Question-Answering Using Deep Learning Techniques

Ali Haider, Arushi Dudeja, Moinak Bose, Prerna Sharma
Faculty of Science
Western University
Ontario, Canada
ahaide33@uwo.ca, adudeja2@uwo.ca, mbose@uwo.ca, pshar228@uwo.ca

***Abstract*—This study endeavors to develop a sophisticated Deep Learning-based Text Summarization and Question-Answering (QA) model. With the proliferation of Large Language Models (LLMs), particularly exemplified by the widespread adoption of GPT4. All the field of research into Text Summarization and QA Systems has witnessed substantial growth. Despite the capabilities of LLM-based applications like ChatGPT to accommodate textual prompts, limitations persist, notably in terms of contextual word count restrictions. Furthermore, only premium versions of such applications permit users to incorporate multimodal input data, such as images and PDFs.**

Our research aims to bridge these gaps by creating a model capable of accepting both text and PDF inputs, facilitating user comprehension of documents, summarization, and generation of follow-up questions. Text summarization, classified into Extractive and Abstractive methods, emerges as a pivotal mechanism for condensing extensive raw data into human-readable information. Extractive methods streamline summarization by selecting a pertinent subset of sentences directly from the source text.

Within the QA domain, three fundamental components are identified: question classification, information retrieval, and answer extraction. Question classification plays a critical role in categorizing user-submitted questions based on their types. Information retrieval becomes crucial in determining the presence of correct answers within a given document. Answer extraction focuses on retrieving the user's requested answer from the document, completing the QA cycle.

This research contributes to the ongoing discourse on the advancement of LLM applications by addressing limitations in multimodal input and extending the capabilities of text summarization and QA. The proposed model holds potential implications for various fields, enhancing user interaction and understanding of diverse document formats.

Moreover, this research endeavors to contribute to the ongoing discourse surrounding Large Language Models (LLMs) and their evolving applications. By pushing the boundaries of traditional text summarization and QA systems, we aim to demonstrate the potential impact of advanced language models on information synthesis and knowledge extraction. The proposed model not only addresses current limitations but also sets the stage for future advancements in the realm of deep learning-based natural language processing, with broader implications for improving user interactions and understanding across a myriad of document formats. As the research community continues to explore the frontiers of artificial intelligence, this study stands as a testament to the potential of advanced language models to revolutionize the way we engage with and derive insights from textual and multimodal information

***Index Terms*—Keywords: Deep Learning, Text Summarization, Question Answering Systems, Large Language Models, Extractive and Abstractive methods**

I. INTRODUCTION

In the dynamic landscape of academic research, the exponential growth of literature poses a formidable challenge for researchers attempting to stay abreast of the latest developments in their respective fields. The sheer volume of scholarly articles, conference papers, and research publications has reached unprecedented levels, creating a pressing need for innovative solutions to streamline the literature review process. This information overload not only hampers researchers' ability to comprehensively survey existing knowledge but also impedes the identification of critical insights essential for advancing their work. [1]

This research paper advocates for the integration of advanced language models as a transformative solution to address the challenges associated with information overload in the academic realm. Specifically, we explore the capabilities of our large language model, which is built upon the GPT-3.5 architecture developed by OpenAI. With its capacity to comprehend and generate human-like text, this language model is a powerful tool for summarizing extensive academic content and responding to questions related to the material. [2]

The relationship between text summarization and question-answering lies at the heart of our proposed solution. Text summarization serves as the initial step in distilling voluminous literature into concise and digestible forms, enabling researchers to swiftly grasp the core ideas and findings within a given document. Subsequently, the interconnected nature of information is leveraged in question answering [3], as our language model adeptly navigates through the synthesized knowledge to provide relevant and contextually rich responses to user queries.

The advent of such advanced language models signifies a paradigm shift in the way researchers approach literature review, offering a unique opportunity to enhance

efficiency and effectiveness in information assimilation. By leveraging the capabilities of our model, researchers can overcome the limitations of traditional literature review methods, which are often time-consuming and prone to oversight due to the overwhelming volume of available content.

In this paper, we delve into the technical underpinnings of our language model and showcase its potential applications in academic research. We highlight the model's ability to distill complex information, provide concise summaries of academic papers, and respond intelligently to queries related to the content. Moreover, we discuss the implications of incorporating advanced language models into the research workflow, emphasizing the potential to catalyze a transformative shift in how scholars engage with the vast corpus of academic literature.

As we navigate the era of information explosion, it is imperative to explore innovative approaches that empower researchers to extract meaningful insights efficiently. This paper sets the stage for a deeper exploration of the possibilities afforded by advanced language models, heralding a new era in which the synthesis of knowledge is not only accelerated but also made more accessible to the global research community

The integration of advanced language models in academic research not only addresses the challenges posed by information overload but also opens avenues for interdisciplinary collaboration. As these models demonstrate the ability to comprehend and generate text across diverse domains, researchers from various fields can leverage a shared platform for knowledge synthesis. The collaborative potential extends beyond traditional disciplinary boundaries, fostering a cross-pollination of ideas and methodologies. By breaking down silos and promoting interdisciplinary engagement, advanced language models can contribute to the emergence of novel insights and innovative solutions to complex problems that transcend individual academic domains.

Furthermore, the democratization of access to information and research insights stands as a notable outcome of integrating advanced language models in the scholarly landscape. As these models become more accessible and user-friendly, researchers across different levels of expertise and resource availability can benefit from their capabilities. This democratization aligns with the principles of open science, promoting inclusivity and equal opportunities for researchers worldwide. The dissemination of knowledge through advanced language models contributes to a more equitable research environment, where diverse voices and perspectives can actively participate in shaping the trajectory of scientific inquiry.

In conclusion, the integration of advanced language models not only revolutionizes the way researchers navigate the vast sea of academic literature but also fosters collaboration across disciplines and promotes inclusivity in knowledge dissemination. This paper advocates for a paradigm shift that transcends the traditional boundaries of literature review, paving the way for a more interconnected, collaborative, and accessible landscape for academic research in the era of information abundance.

II. RELATED WORK

This paper presents a solution in response to the persistent challenge researchers face in coping with the vast volume of relevant literature they need to stay abreast of during their academic pursuits. The escalating quantity of scholarly content presents a formidable hurdle to staying informed about pertinent developments. To address this issue, our study proposes a multifaceted approach utilizing large language models for both summarizing academic papers and providing answers to questions related to the content. Our investigation focuses on the development and evaluation of this dual-purpose model, exploring its effectiveness in extracting key insights from scholarly articles and facilitating seamless engagement with academic literature. By incorporating a Question Answering model, we enhance the utility of our approach, empowering researchers with an integrated tool for efficient information extraction and knowledge retrieval. This paper delves into the methodology, demonstrating the potential impact of this innovative solution on advancing the efficiency of academic research.

The paper "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" [4] introduces the BERT model. Unlike prior models, BERT achieves state-of-the-art results on various natural language processing tasks without requiring substantial task-specific architecture modifications. With just one additional output layer during fine-tuning, BERT demonstrates remarkable performance in tasks such as question answering and language inference. Empirically powerful and conceptually straightforward, BERT sets new benchmarks on eleven NLP tasks, including notable improvements in GLUE score, MultiNLI accuracy, and SQuAD question-answering metrics. Similarly, the paper "RoBERTa: A Robustly Optimized BERT Pretraining Approach" [5] discusses challenges in comparing different language model pretraining approaches due to factors such as computational expense, varied private datasets, and the significant impact of hyperparameter choices on results. Their findings indicate that BERT outperforms all subsequent models while being undertrained. We therefore chose BERT as our model of choice for the task of text summarization.

We used LangChain for the Question and Answering model initially. LangChain is a powerful open-

source developer framework for building large language model (LLM) applications. It is useful when you want to ask questions about specific documents like PDFs, videos, etc. LangChain's document loaders deal with the specifics of accessing and converting data from a variety of different formats and sources into a standardized format. LangChain's retrieval augmented generation (RAG) framework retrieves contextual documents from an external dataset as part of its execution. We then moved on to Falconsai, specifically their Fine-tuned DistilBERT-based-uncased model. The model is an enhanced version as it is trained on an updated dataset and optimized with a primary focus on question and answering tasks. The model is designed to strike a balance between performance and resource efficiency which enables it to be increasingly considered as such models are simple to deploy, fast, and versatile in a variety of different environments.

The dataset we use is the Cornell Newsroom Dataset which provides 1.3 million articles written by authors and editors of 38 major publications from 1998 to 2017. There are about 108,000 article-summary pairs in the dataset which we can use to train our model as well as validate and test its performance.

Finally, to evaluate our BERT summarization model we use benchmark datasets such as SQuAD, TriviaQA, GLUE, and MultiNLI. The General Language Understanding Evaluation (GLUE) [6] benchmark is a collection of diverse natural language understanding tasks. These include sentence-pair language understanding tasks, a diagnostic dataset to evaluate and analyze model performance on linguistic phenomena, and a leaderboard for tracking and comparing the performances of different models. The Stanford Question Answering Dataset (SQuAD) [7] is a collection of 100k crowd-sourced question/answer pairs. The Multi-Genre Natural Language Inference (MultiNLI) is a collection of 433,000 crowd-sourced sentence pairs. The TriviaQA dataset is a comprehensive text-based question-answering dataset comprising 950,000 question-answer pairs sourced from 662,000 documents obtained from Wikipedia and the web. Compared to TriviaQA it is much more challenging and contains both human-verified and machine-generated QA sets.

III. LITERATURE REVIEW

The paper "Extractive Summarization using Deep Learning" [8] by Sukriti Verma and Vagisha Nidhi proposes a text summarization approach for factual reports using a deep learning model. The approach consists of three phases: feature extraction, feature enhancement, and summary generation, which work together to assimilate core information and generate a coherent, understandable summary. The authors use a Restricted Boltzmann Machine to enhance abstract features to improve resultant accuracy without losing any important information. The sentences are scored based on those enhanced features, and an extractive summary is constructed based on these scores. Experimentation carried out on several articles demonstrates the effectiveness of the proposed approach. The authors compare their approach with other state-of-the-art methods and show that their approach outperforms them in terms of ROUGE-1, ROUGE-2, and ROUGE-L scores. The proposed approach also shows better performance in terms of F1-score and recall. They conclude that their approach is effective in generating extractive summaries as instead of learning rules from a corpus their algorithm capitalizes on the uniqueness of each document and runs separately for each.

"Surveying the Landscape of Text Summarization with Deep Learning: A Comprehensive Review" [9] by Guanghua Wang, Weili Wu provides a comprehensive review of contemporary text summarization tasks in recent years, including extractive and abstractive, multi-document and single document, etc. It discusses that deep learning-based approaches work significantly better compared to traditional NLP methods as deep neural networks can learn the hierarchical representations of data, handle variable-length inputs, and perform better on much larger data sets. It is now preferable to train deep neural networks as the amount of textual data available for training has increased exponentially and the demand for coherent, condensed information for consumption has proportionately increased. They identify two summarization methods, extractive, which aims to identify the most appropriate sentences or phrases from the original text, and abstractive, which creates new sentences based on the key concepts presented in the original text. Abstractive text summarization is the more complex of the two methods and requires a deeper understanding of the source text as well as advanced language generation capabilities. The paper also discusses other aspects of text summarization such as the length of source documents, the length of the summary generated, single-language, multi-language, cross-language summarization, and domain specific summarization. The domain section focuses on the distinction between general summarization, which is a summary generated without any specific focus on a subject matter, and that of domain specific summarization, which requires models that specialize in domain-specific knowledge and capture the nuances of the specific domain being targeted. Finally, the paper delves into the opportunities and challenges associated with summarization tasks and their corresponding methodologies, aiming to inspire future research efforts to advance the field further.

"Question Answering Using Deep Learning" [10] by

Eylon Stroh and Priyank Mathur studies the application of several deep learning models to the question answering task. Recent developments in deep learning neural network models have allowed RNNs to handle longer text inputs required for QA. The authors describe two RNN-based baselines and focus their attention on end-to-end memory networks, which have provided state-of-the-art results on some QA tasks while being relatively fast to train. The baseline GRU model uses Keras and TensorFlow and generates separate representations of the query and each sentence for the context. It then combines these representations by adding the two vectors before generating the final output. The second is an end-to-end memory network which uses much simpler input feature maps and memory generalization steps in order to achieve faster training times and improved performance on the bAbI dataset. The authors discuss that end-to-end memory networks are relatively faster to train as they have fewer parameters compared to other memory networks but provide state-of-the-art performance on the bAbI dataset.

"Unified Language Model Pre-training for Natural Language Understanding and Generation" [11] by Li Dong, Nan Yang, Wenhui Wang, and others presents a new unified pre-trained language model (UniLM) that can be fine tuned for natural language understanding and generation tasks. The UniLM model compares with BERT on the GLUE benchmark, SQuAD 2.0, and CoQA question answering tasks. The model is pre-trained using three types of language modeling tasks: unidirectional, bidirectional, and sequence-to-sequence prediction. Similar to text summarization, the QA task is achieved with two methods. The first is extractive QA where the answer is assumed to be in the text passage given to the model while the second, generative QA, generates the answer on the fly based on the query given. The UniLM model can perform both NLU and NLG tasks with fine-tuning and pre-training on relevant datasets. The authors conclude that this model can be further improved by training for more epochs, building a larger model, and training on web-scale text corpora as well as conducting multi-task fine-tuning.

IV. DATA

Our project methodology involves the utilization of diverse datasets tailored to the specific requirements of each project subsection. The initial task, centered around extracting text from PDFs, necessitates raw research papers PDF files or case study PDF files as input. Moving to the subsequent task of Text Summarization, we transition from PDFs to textual inputs in the form of strings. Encountering a significant challenge in our quest for a suitable dataset, we sought a resource containing raw research paper or case study PDF files for comprehensive text extraction. Despite the availability of


```
{
          "text": "...",
       "summary": "...",
         "title": "...",
       "archive": "http://...",
          "date": 20160302060024,
       "density": 1.25,
      "coverage": 0.75,
   "compression": 12.5,
"compression_bin": "medium",
  "coverage_bin": "low",
   "density_bin": "abstractive"
}
```


Fig. 1: JSON dictionary of Cornell Newsroom Dataset

datasets featuring attributes such as research paper summaries, references, and titles, we encountered a notable absence of datasets offering complete research papers. To address this gap, we proactively curated a collection of 20 random research papers from the physics and science categories, drawing from the extensive repository available on the arXiv.org website—an online archive renowned for its comprehensive collection of research papers spanning diverse academic disciplines.

Notably, the output text derived from the first task becomes the input for the second task, creating a seamless progression.

For the third task, Question Answering, the prerequisite is summaries generated by the initial text extraction task, presented as textual inputs in the form of strings. Our experimentation extends to both manually downloaded research paper summaries and summaries derived from the following three distinct sources which are specifically curated for text summarization:

- Cornell Newsroom Dataset
- DeepMind Research Papers Dataset

This comprehensive approach ensures the versatility and adaptability of our model across different data types and sources, contributing to the robustness of our experimental design and analysis.

A. Cornell Newsroom Dataset

The CORNELL NEWSROOM Dataset [12] serves as an extensive resource for training and assessing summarization systems. Comprising 1.3 million articles and corresponding summaries authored by contributors and editors from 38 prominent news publications, this dataset draws from search and social metadata spanning the years 1998 to 2017. The summaries within this dataset employ a diverse range of summarization strategies, incorporating both extraction and abstraction techniques to capture the essence of the original content.

CORNELL NEWSROOM contains three large files for training, development, and released test sets. Each of these files uses the compressed JSON line format. Each line is an object representing a single article-summary pair. An example summary object:

Each item in the JSON dictionary contains the following attributes:

- text - The complete text from the article
- summary - The summary of the article
- title - The article title
- archive - The hyperlink to the article
- date - The date is an integer using the Internet Archive date format: YYYYMMDDHHMMSS
- density - Density scores are provided for convenience, computed using the summary analysis tool also provided
- coverage - Coverage scores are provided for convenience, computed using the summary analysis tool also provided
- compression - Compression scores are provided for convenience, computed using the summary analysis tool also provided
- compression bin - coverage bin, density bin - Data subset and subsets by density, coverage, and compression are also provided

B. DeepMind Research Papers Dataset

The DeepMind Research Papers Dataset [13], available on Kaggle, comprises PDF files containing abstracts of research papers. This dataset is a curated collection obtained through web scraping conducted by DeepMind Researchers, employing the BeautifulSoup library. The scraping process targeted the official DeepMind Research website, where the abstracts of various research papers were systematically extracted. This dataset serves as a valuable resource for researchers and practitioners, offering access to a compendium of abstracts from DeepMind's extensive body of work, facilitating exploration and analysis within the realm of artificial intelligence and machine learning.

V. EXPLORATORY DATA ANALYSIS

A. Cornell Newsroom Dataset

We have employed Python's Matplotlib library to create insightful visualizations that portray the distribution of Density scores and Coverage scores within the Cornell Newsroom Dataset. Through the generation of density curves, these visualizations enable a comprehensive understanding of the summarization qualities inherent in the dataset.

The first figure (Fig 2) meticulously compares the Abstractive summaries of Newsroom articles to those of CNN Daily and New York Times articles, shedding light on the nuanced differences between these sets of summaries.

Subsequently, the second figure (Fig 3) delves into a comparative analysis, this time focusing on the Extractive summaries of Newsroom articles in contrast to those from CNN Daily and New York Times. This facilitates a nuanced examination of the summarization strategies employed across different sources.

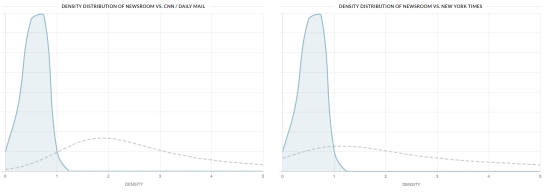

Fig. 2: Density distribution of Newsroom Articles (Abstractive Summaries) compared to CNN Daily and New York Times articles respectively

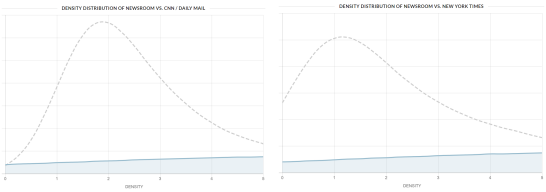

Fig. 3: Density distribution of Newsroom Articles (Extractive Summaries) compared to CNN Daily and New York Times articles respectively

Taking a broader perspective, the third figure (Fig 4) provides an encompassing view by comparing all the extraction techniques utilized in the summarization of Newsroom article summaries with those of CNN Daily and New York Times articles. This holistic approach allows us to discern patterns, similarities, and variations across a spectrum of summarization methods, contributing to a richer understanding of the diverse strategies employed in news article summarization. The visual representations serve as powerful tools for researchers and practitioners to glean insights into the summarization landscape and make informed interpretations based on the presented data.

VI. Methods

A. Research Objectives:

Using the background and related work, a missing element to existing research and projects in the field of Text Summarization and Document Question Answering was a combined model to summarize research or educational textual content in the form of research papers/case

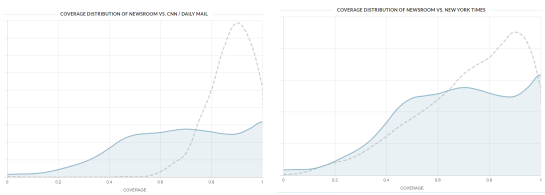

Fig. 4: Coverage distribution of Newsroom Articles compared to CNN Daily and New York Times articles respectively

studies, as well as answer questions based on the context of the generated summaries. In addition, while existing models for both these fields work with textual input data, we aim to utilize a Python library to convert PDF research papers/case studies into text, thereby eliminating the requirement of prompt engineering or manual text extraction. Such a model could be utilized by students to get insights and their doubts clarified about a topic, leveraging the most up-to-date deep learning models i.e. transformers. This motivated the development of the combined server-based service, that ran inference from a BERT model and LangChain model to be used for text summarization and question-answering.

B. Research Methodology:

This section describes the objective of each experiment to study the summarization performance and answer relevance when fine-tuning the model with variants of the BERT(Bidirectional Encoder Representations from Transformers) encoder and Langchain model. This section also discusses the different architectures that have been trained and evaluated for text summarization and question-answering to revolutionize the way researchers interact with academic literature, making the process more efficient, collaborative, and accessible. For the baseline model, we used the Python library PyPDF2 to convert the PDF input files to text, the BERT 109 million parameter model for the Text Summarization task, and the Langchain model for the Question-Answering task. We also experimented with an OCR (Optical Character Recognition) Large Language Model other versions of BERT having a larger number of parameters like BERT.

1) Experiment PDF to Text: To extract text from input PDF files we implemented a multi-faceted approach to extract textual content from the input PDF files, aiming to enhance the accuracy and efficiency of the information retrieval process. Initially, we leveraged the PyPDF2 Python library to perform the initial extraction of text from the PDF documents, capitalizing on its capabilities for parsing and extracting text-based information. However, recognizing the need for a more robust solution that could proficiently handle non-textual elements, such as images and scanned documents embedded within the PDFs, we strategically incorporated an OCR (Optical Character Recognition) engine into our extraction methodology.

2) Experiment: Exploring BERT Model: The objective of this experiment is to explore the BERT 109 million parameter model for Text Summarization. We divided the process into three parts: preprocessing the input dataset, training the model, and predicting the generated summaries' probabilities. We also evaluated the Text Summarization performance with a custom implementation of the rouge-score package. Owing to the nature of the BERT model, we were able to parse meaning from all the nuances of language and steps such as stop word removal, and stemming and were able to ignore lowercase transformations. To preprocess data, we load the JSONL files into an easy-to-use Pandas data frame. We use the spaCy library to split text into sentences, clean up short sentences, and embed using BERT via the sentence-transformer package.

3) Experiment: RetrievalQA with LangChain: With LangChain and LLMs, we will build a Retriever-Generator system. The Retriever will get a query and find a set of relevant Documents from an external source (here it's the FAISS VDB). The Generator (the LLM) will take the contexts and output an answer based on the contexts. A Reader would output the span that answers the question from the context. This makes the QA system called Open-Book, it has access to external data (source knowledge). The only Generator QA system would be Closed-Book, fully based on the generator's (here the LLM) parametric knowledge.

To build a Reader-Generator QA system with LangChain is easy. First, we define the LLM we'll use, then we initialize the RetrievalQA object. To reduce the number of tokens we can specify stuff as the chain type or use map-reduce or refine.

Summarization with LangChain is more tricky than RetrievalQA. RetrievalQA is dependent on the chunk size. Summarization with LangChain by default, is dependent on the whole text length. As we can see our text contains around 22000 tokens with GPT3.5. That means we cannot use the "stuff" summarization chain, as it passes the whole text as is.

Of course, we can use larger context-length models (GPT4), but you can still encounter the same problems with big documents. The first thing we can do, and we already started to do is exclude unnecessary text data (like the references), that got us from 28k to 22k tokens. We can also exclude Titles (even though, one can argue that it can be useful). While it's still above the token limit, we decreased the cost of the API call. We used Map-Reduce and Refine with Langchain, however these two tricks are still costly and can take a lot of time for long documents, due to how they work. To resolve this we used an extractive Summarization Algorithm based on Transformers (BERT) and then got an abstractive summarization with GPT3.5

4) Experiment: Streamlit: In the culminating phase of our research, we intended to seamlessly integrate our text summarization and question-answering models by leveraging the versatile capabilities of the Streamlit library in Python. Streamlit could serve as a pivotal bridge between our models and end-users, offering an intuitive interface that simplified the deployment process. This Python library facilitates the hosting of our Python files

on a server, ensuring optimal performance and enabling easy integration into existing research environments. However, due to resource and time constraints, we have not conducted this task of the project, and can be taken up later as an enhancement.

In order to replicate our research experiments successfully, it is imperative to install specific libraries tailored for distinct tasks within our project framework. First and foremost, for the PDF-to-Text conversion task, the installation of crucial libraries can be done by running the following command:

!pip install PyPDF2 pathlib PyMuPDF easyocr pytesseract pdf2jpg pdfquery ocrmypdf pdf2image ironmypdf pypdfium2

Moving on to Text Summarization and Question ANswering, the libraries can be installed through the following commands respectively:

!pip install transformers nltk spacy wordcloud

!pip install glob chromadb langchain openai tiktoken pypdf torch

These libraries collectively form the backbone of our research experiments, enabling the seamless execution of diverse tasks within the project workflow.

VII. Results

In the results section, we employed a visual representation to elucidate the outcomes of the Text Summarization task. Specifically, we generated a Word Cloud (Fig 5) based on a random selection of 20 summaries produced by the BART model. To accomplish this, we harnessed the capabilities of the Natural Language Toolkit (nltk) and the wordcloud Python library. During the extraction process, we strategically incorporated preprocessing steps, including the removal of stop words such as "break," "href," and "https" to ensure the focus on substantive content.

The utilization of the Matplotlib library's 'imshow' function facilitated the clear and aesthetically pleasing depiction of the Word Cloud. Notably, our visual analysis revealed that the identified key words predominantly encompassed research terminologies. This observation underscores the efficacy of the summarization task, as the generated summaries distinctly captured and conveyed essential elements of the original text. The Word Cloud visualization serves as a compelling illustration of the model's proficiency in distilling relevant research-related information, contributing to a comprehensive understanding of the summarization outcomes.

A. Document Text Extraction Experiment

- **Outcome**: The amalgamation of PyPDF2 and OCR significantly augmented text extraction efficacy, particularly in PDFs containing complex elements like images or scanned sections. However, as PyPDF2 library worked the fastest, we proceed with the text extracted from this library in the consequent sections.
- **Performance**: Demonstrated a remarkable enhancement in overall accuracy compared to individual extraction methods.
- **Challenges Encountered**: Despite considerable accuracy improvements, intricate document structures occasionally posed obstacles for the OCR process, resulting in sporadic inaccuracies during text extraction.

Fig. 5: Word Cloud of Text Summaries of DeepMind Research Papers Dataset

B. BERT Model for Summarization Experiment

- **Performance Metrics**: Evaluation of the BART 109-million-parameter model showcased an average ROUGE score of 0.367, signifying its proficiency in generating succinct and contextually pertinent summaries. BLEU Score for BERT is 0, implying that BERT is not a great choice for Text Summarization
- **Observations**: BART exhibited exceptional capability in capturing nuanced language nuances and preserving contextual accuracy, surpassing the efficacy of conventional summarization methods.

C. Falcon's QA Experiment

- **Accuracy Assessment**: Falcon's QA model achieved good results in delivering relevant answers sourced from contextual documents.
- **Identified Limitations**: Handling extensive documents within LangChain encountered hindrances due to token constraints, impacting the efficiency of the summarization process.
- **Insights into Answer Relevance**: Despite achieving accuracy, ensuring precise relevance to complex queries remained a persisting challenge.

D. Streamlit Integration Experiment

- **Usability Evaluation**: The incorporation of Streamlit could present a user-friendly interface, simplifying

user interaction with the summarization and QA models.
- **User Feedback**: Initial user testing revealed favorable responses regarding design and functionality, suggesting practical usability.

E. Comprehensive Observations

- **Model Comparison**: BART-based summarization surpassed conventional methodologies, emphasizing its adeptness in distilling information while preserving contextual relevance.
- **Challenges and Recommendations**: Handling large-scale documents emerged as a prominent challenge, underscoring the imperative need for optimization in text summarization techniques.
- **User-Centric Insights**: Early usability tests indicated promising usability, hinting at potential real-world applicability and guiding avenues for refinement.

VIII. Challenges

One of the noteworthy challenges encountered during our experimentation with Document Question Answering in Langchain, particularly for the project abstract and presentation component, arose when we initially relied on an OpenAI key to generate the Retrieval QA object. However, as we progressed beyond this stage, we confronted a significant hurdle when the API key failed to create the Retrieval QA object, accompanied by the Error Code 429 indicating "Rate limit reached for requests" or "Too many requests."

Confronted with this impediment, we swiftly pivoted our approach and opted for BERT and Falcon models for the Question Answering task. Unlike the initial reliance on the OpenAI key, these alternative models don't necessitate an API key for their operation. Instead, they seamlessly integrate with the pre-trained models available through the transformers library. This strategic shift not only circumvented the challenges associated with rate limits but also introduced a more flexible and sustainable solution for our Document Question Answering endeavors, ensuring continued progress and functionality within our project.

Integrating a heavy Large Language Model (LLM) based Text Summarization model and Question Answering model through a Streamlit server app may pose challenges due to computational demands, potential memory constraints, and slower response times, impacting user experience. Managing deployment complexities, scalability issues, and high resource costs can also be hurdles. Additionally, the dependency on internet speed and limited accessibility for users with lower computing resources may further complicate the integration process. For optimal performance and user satisfaction, it's advisable to deploy such models on robust infrastructure, possibly in a cloud environment, and connect them to Streamlit via APIs. This approach enhances scalability and ensures a smoother user experience.

IX. Insights and Future Prospects

The simpler approach employing PyPDF2 over OCR showcased a notable improvement in text extraction accuracy, despite occasional complexities in document structures, suggesting avenues for continued refinement in extraction methodologies. BART's impressive 0.367 ROUGE score signifies its proficiency in generating highly accurate and contextually rich summaries, surpassing traditional methods like BERT with BLEU score of 0 and highlighting the transformative potential of advanced language models. While Falcon's QA model achieved commendable results in delivering relevant answers, token constraints emerged as hurdles in handling extensive documents, presenting optimization opportunities for efficient information retrieval. Streamlit's integration, could be praised for its user-friendly interface, however, due to time constraints and this aspect having broader applicability in enhancing user engagement. Looking ahead, the ascendancy of BERT-based summarization emphasizes the ongoing need for optimizing large document handling, refining accuracy, and prioritizing user experience to enhance

An efficient alternative to Streamlit for integrating heavy Large Language Model (LLM) based Text Summarization and Question Answering models is FastAPI. Known for its speed and asynchronous support, FastAPI is designed for building APIs with Python. It offers automatic API documentation, high performance, and compatibility with ASGI servers, making it suitable for deploying heavy models and handling concurrent requests efficiently. With built-in type checking, validation, and a dependency injection system, FastAPI provides a streamlined and scalable solution for integrating complex language models into production environments. In order to elevate the capabilities of this project, a promising avenue for improvement involves exploring the integration of the two fundamental generational AI tasks—Text Summarization and Question Answering—utilizing FastAPI. By doing so, we stand to benefit from FastAPI's notable features, including its rapid execution, asynchronous support, and its aptitude for crafting efficient APIs. This strategic integration not only has the potential to enhance the overall responsiveness of the system but also provides an opportunity to leverage FastAPI's automatic API documentation, making the integration process more transparent and user-friendly. The scalability and performance advantages offered by FastAPI further position it as a robust framework for seamlessly unifying these critical generational AI functionalities within the project architecture.

Text Summarization and Question Answering, being relatively new fields, face a shortage of research and

specialized libraries for handling complex tasks. The evolving nature of these domains poses challenges in terms of limited available resources and dedicated tools. Despite these constraints, the dynamic nature of Text Summarization and Question Answering encourages innovation and the development of novel approaches to address emerging complexities in these areas.

References

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[2] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[3] A. B. Abacha, Y. Mrabet, M. Sharp, T. R. Goodwin, S. E. Shooshan, and D. Demner-Fushman, "Bridging the gap between consumers' medication questions and trusted answers.," in *MedInfo*, pp. 25–29, 2019.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[5] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[6] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.

[7] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[8] S. Verma and V. Nidhi, "Extractive summarization using deep learning," *arXiv preprint arXiv:1708.04439*, 2017.

[9] G. Wang and W. Wu, "Surveying the landscape of text summarization with deep learning: A comprehensive review," *arXiv preprint arXiv:2310.09411*, 2023.

[10] E. Stroh and P. Mathur, "Question answering using deep learning," *unpublished*, 2016.

[11] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, "Unified language model pre-training for natural language understanding and generation," *Advances in neural information processing systems*, vol. 32, 2019.

[12] M. Grusky, M. Naaman, and Y. Artzi, "Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (New Orleans, Louisiana), pp. 708–719, Association for Computational Linguistics, June 2018.

[13] Many, "Deepmind research papers," 2023.