OPEN DOMAIN QUESTION ANSWERING MODEL

**Introduction:**

**Summary: Open-Domain Question Answering Platform**

An **Open-Domain Question Answering (ODQA) platform** leverages advanced AI to provide accurate and concise answers to user queries across a wide range of topics, without being limited to a specific domain. By combining natural language processing and information retrieval, it extracts answers directly from vast datasets like Wikipedia or CommonCrawl, making it more effective than traditional search engines.

We chose an ODQA platform because it revolutionizes how information is accessed, offering users quick, precise responses without sifting through extensive search results. This capability makes it invaluable for applications in education, research, and customer support, where efficiency and accuracy are critical. It provides a seamless, conversational way for users to interact with knowledge bases, enhancing productivity and user experience.  
  
Initially we worked on creating an open-domain question-answering platform based on NCERT textbooks, and came up with a code(can be found in GitHub under initialODQAmodel) which served as a starting point for preprocessing data. We have used PyMuPDF to extract text from NCERT textbooks in PDF format and have written a function to clean and normalize that text. The preprocessing steps include removing non-ASCII characters, cleaning up symbols, tokenizing, lemmatizing, and filtering out stopwords.

We’ve automated the process to handle multiple PDFs, saving the cleaned output as text files with a preprocessed\_ prefix. This setup allowed us to turn raw textbook data into a clean, structured text corpus, ready to be transformed into a dataset for training or generating questions. This was a crucial step in our research to build the platform.

Due to lack of availability of datasets regarding NCERT we had to switch over to the Natural questions dataset to proceed further.

**Link to the research paper:-** <https://arxiv.org/abs/2101.00027>  
  
**Summary of the research paper:-**

The research paper "Datasheet for the Pile" provides a detailed overview of "The Pile," a large-scale text dataset compiled by EleutherAI. The dataset, which totals 825 GiB, is created specifically for training and testing large-scale language models. It consists of 22 diverse text sources, ranging from original scrapes conducted by the EleutherAI team, to data shared by the owners, and third-party text from online scrapes.

The paper outlines the motivation behind creating The Pile, which was to provide a massive, high-quality dataset to address the growing needs of natural language processing (NLP). The authors describe the various sources of data, such as PubMed Central, arXiv, Wikipedia, GitHub, and others, each contributing a unique aspect of human-authored text. The dataset has been preprocessed to ensure the removal of non-textual content, errors, and noise, though the authors acknowledge the inherent challenges of processing such vast amounts of data.

In addition to explaining the components of the dataset, the paper also discusses ethical considerations, the collection process, and the dataset's potential use in various NLP tasks like language modeling, text generation, and research in specific fields like medicine and law. Furthermore, the paper addresses the legal and ethical implications, including concerns about potential biases and the handling of sensitive data.

**Datasets and their origin:**

1. Wikipedia Dataset

- Source: http://eaidata.bmk.sh/data/wikipedia-en.tar.gz

- Size: 6.85 GB

- Number of Documents: 6,033,151

- Description: A comprehensive English-language dataset of Wikipedia articles, providing a broad range of encyclopedic knowledge across various topics.

2. OpenSubtitles Dataset

- Source: http://eaidata.bmk.sh/data/opensubtitles\_out.tar

- Size: 13.94 GB

- Number of Documents: 446,612

- Description: A collection of subtitles from movies and TV shows, which can be useful for natural language processing tasks involving dialogue and conversational text.

3. BookCorpus Dataset

- Sources:

- https://the-eye.eu/public/AI/pile\_preliminary\_components/books1.tar.gz

- http://battle.shawwn.com/sdb/books1/books1.tar.gz

- Size: 6.77 GB

- Number of Documents: 17,868

- Description: A collection of books in EPUB format, providing a diverse range of literary texts for training language models.

4. OpenWebText Dataset

- Source: Google Drive (https://drive.google.com/uc?id=1EA5V0oetDCOke7afsktL\_JDQ-ETtNOvx)

- Size: 39.76 GB

- Number of Documents: 8,013,769

- Description: A web crawled dataset of high-quality text from the internet, created as an open-source alternative to the WebText dataset used to train GPT-2.

5. ArXiv Dataset

- Source: http://eaidata.bmk.sh/data/arxiv.jsonl.zst

- Size: 60.35 GB

- Number of Documents: 1,264,405

- Description: A collection of academic research papers from the arXiv repository, covering various scientific and academic disciplines.

6. GitHub Dataset

- Sources:

- https://the-eye.eu/public/AI/pile\_preliminary\_components/github.tar

- http://eaidata.bmk.sh/data/github.tar

- Size: 677.14 GB (Full), 102.18 GB (Small)

- Number of Documents: 56,626,342 (Full), 19,021,454 (Small)

- Description: A comprehensive dataset of source code from GitHub repositories, useful for code-related machine learning tasks.

7. CommonCrawl Dataset

- Source: http://eaidata.bmk.sh/data/pile\_cc\_filtered\_deduped.jsonl.zst

- Size: 243.87 GB

- Number of Documents: 54,953,117

- Description: A massive web crawl dataset that provides a broad sample of text from across the internet, filtered and deduplicated.

Some interesting observations:-

- Most datasets are sourced from multiple locations for redundancy

- Datasets cover a wide range of domains: encyclopedic knowledge, literature, academic papers, web text, source code, and more

- Sizes range from a few gigabytes to hundreds of gigabytes

- Document counts vary widely, from tens of thousands to millions

**Model:**  
  
  
The \*EleutherAI GPT-Neo 1.3B model\* is a large-scale, open-source language model designed to generate human-like text. As part of the \*GPT (Generative Pretrained Transformer)\* family, it employs a transformer-based architecture to predict the next word in a sequence, making it well-suited for a variety of natural language processing tasks, including \*text generation, \*\*question answering, and \*\*summarization\*. With 1.3 billion parameters, the model strikes a balance between size and performance, offering powerful capabilities for applications like creative writing, conversational AI, and automated content creation. Its open-source nature allows for customization and fine-tuning, making it a popular choice in the AI community.

To make the GPT-Neo 1.3B model more accessible and user-friendly, the \*Hugging Face Transformers pipeline\* provides a streamlined approach for loading and utilizing the model. The pipeline("text-generation", model="EleutherAI/gpt-neo-1.3B") function abstracts the complexities of model setup and inference, enabling users to easily generate text by specifying parameters like \*max\_length\* (for response length) and \*temperature\* (for controlling randomness). This simplicity empowers developers to focus on crafting solutions without delving into the intricacies of the underlying model architecture.

However, beyond efficient model inference, providing an intuitive interface is key to democratizing access to such powerful tools. Here, \*Gradio integration\* bridges the gap between advanced machine learning models and end-users. Gradio enables the creation of customizable web interfaces using its \*Blocks API\*, combining components like text boxes for user prompts, sliders for adjusting parameters, and buttons for triggering responses. This results in an interactive platform where users can experiment with the model in real time, regardless of their programming expertise.

Together, the integration of \*GPT-Neo 1.3B, the Hugging Face \*\*Transformers pipeline, and the \*\*Gradio interface\* creates a seamless environment for open-domain question answering and text generation tasks. Users can input prompts, modify response behavior through settings like max\_length and temperature, and receive real-time feedback, making it suitable for both developers and non-technical users. This setup can be deployed locally or hosted online, providing a versatile platform for exploring the model’s capabilities across diverse applications.

By combining \*EleutherAI's powerful language model, \*\*Hugging Face’s model management tools, and \*\*Gradio’s user-friendly interface\*, this ecosystem ensures that cutting-edge AI research is not only accessible but also practical for real-world use cases.

**Results:**   
  
  




