Rag

This notebook guides you through the process of initializing and running a Retrieval-Augmented Generation (RAG) model using the Hugging Face Transformers library. The notebook is structured to help explain each step for generating responses using the RAG model. Due to memory constraints, we have uploaded it as a PDF for illustration.

RAG Model Overview

RAG, or Retrieval-Augmented Generation, is a technique that combines the strengths of retrieval-based and generation-based models to improve the quality and relevance of generated text. In a RAG system, a retrieval model is used to fetch relevant documents or passages from a large corpus, and a generation model is then used to produce a coherent and contextually appropriate response based on the retrieved information.

Here's a step-by-step explanation of how RAG works, along with a simple example using Python and the Hugging Face Transformers library:

Retrieval & Generate phases

Retrieval-Augmented Generation (RAG) is a powerful technique that combines the strengths of retrieval-based and generation-based models to produce more accurate and contextually relevant responses. The process can be broken down into two main phases: the Retrieval Phase and the Generation Phase. Let's delve into each phase in detail.

Retrieval Phase

The Retrieval Phase involves fetching relevant documents or passages from a large corpus based on the input query. This phase is crucial because the quality of the generated response heavily depends on the relevance and accuracy of the retrieved information. Here are the key steps in the Retrieval Phase:

1. **Query Processing**: The input query is processed to extract relevant keywords or embeddings.
2. **Document Retrieval**: The processed query is used to search a pre-built index of documents. The index can be based on various retrieval models, such as BM25, Dense

Passage Retrieval (DPR), or others. For faster retrieval, you can use a "condense" index.
3. **Ranking**: The retrieved documents are ranked based on their relevance to the query. The top-k most relevant documents are selected for the next phase.

Generation Phase

The Generation Phase involves using the retrieved documents as additional context to generate a coherent and contextually appropriate response. This phase leverages a generation model to produce the final output. Here are the key steps in the Generation Phase:

1. **Context Integration**: The retrieved documents are integrated with the input query to provide additional context for the generation model.
2. **Response Generation**: The generation model uses the combined context to produce a response. The model can be fine-tuned on specific tasks to improve the quality of the generated text.
3. **Post-Processing**: The generated response may undergo post-processing steps, such as decoding and cleaning, to ensure it is in a readable format.

Example Code

Below is an example code that demonstrates both the Retrieval Phase and the Generation Phase using the Hugging Face Transformers library. This example uses the `RagTokenizer`, `RagRetriever`, and `RagSequenceForGeneration` classes. We will use the "condense" index for faster retrieval.

```
!pip install transformers
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in /usr/local/lib/
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/py
Requirement already satisfied: hf-xet<2.0.0,>=1.1.2 in /usr/local/lib/python3.
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/pyth
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11
```

```
pip install torch
```

```
Requirement already satisfied: torch in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/p
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-pack
Collecting nvidia-cuda-nvrtc-cu12==12.4.127 (from torch)
  Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-manylinux2014_x86_64.
Collecting nvidia-cuda-runtime-cu12==12.4.127 (from torch)
  Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_6
Collecting nvidia-cuda-cupti-cu12==12.4.127 (from torch)
  Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-manylinux2014_x86_64.
Collecting nvidia-cudnn-cu12==9.1.0.70 (from torch)
  Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl.m
Collecting nvidia-cublas-cu12==12.4.5.8 (from torch)
  Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-manylinux2014_x86_64.whl.
Collecting nvidia-cufft-cu12==11.2.1.3 (from torch)
  Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl.m
Collecting nvidia-curand-cu12==10.3.5.147 (from torch)
  Downloading nvidia_curand_cu12-10.3.5.147-py3-none-manylinux2014_x86_64.wh
Collecting nvidia-cusolver-cu12==11.6.1.9 (from torch)
  Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.wh
Collecting nvidia-cusparse-cu12==12.3.1.170 (from torch)
  Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-manylinux2014_x86_64.
Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in /usr/local/l
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in /usr/local/lib/py
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib/
Collecting nvidia-nvjitlink-cu12==12.4.127 (from torch)
```

```
  Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.w
Requirement already satisfied: triton==3.2.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/di
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/
Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-manylinux2014_x86_64.whl (3
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 3.8 MB/s eta 0:00
Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-manylinux2014_x86_64.wh
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 23.0 MB/s eta 0:00:
Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-manylinux2014_x86_64.wh
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 17.4 MB/s eta 0:00:
Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_64.
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 14.7 MB/s eta 0:0
Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl (66
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 2.5 MB/s eta 0:00
Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl (21
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 211.5/211.5 MB 4.2 MB/s eta 0:00
Downloading nvidia_curand_cu12-10.3.5.147-py3-none-manylinux2014_x86_64.whl
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 9.8 MB/s eta 0:00:0
Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.whl
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 127.9/127.9 MB 8.1 MB/s eta 0:00
Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-manylinux2014_x86_64.wh
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.5/207.5 MB 4.6 MB/s eta 0:00
Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 26.3 MB/s eta 0:00:
Installing collected packages: nvidia-nvjitlink-cu12, nvidia-curand-cu12, nv
  Attempting uninstall: nvidia-nvjitlink-cu12
    Found existing installation: nvidia-nvjitlink-cu12 12.5.82
    Uninstalling nvidia-nvjitlink-cu12-12.5.82:
      Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
```

```
!pip install faiss-cpu
```

```
Requirement already satisfied: faiss-cpu in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: numpy<3.0,>=1.25.0 in /usr/local/lib/python3.11
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-pac
```

```
from transformers import RagTokenizer, RagRetriever, RagSequenceForGeneration
import torch
```

```
# Initialize the tokenizer
tokenizer = RagTokenizer.from_pretrained("facebook/rag-token-nq")
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: Use
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access pu
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/transformers/models/bart/configuration
  warnings.warn(
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'DPRQuestionEncoderTokenizer'.
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'DPRQuestionEncoderTokenizerFast'.
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'BartTokenizer'.
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'BartTokenizerFast'.
```

```
# Initialize the tokenizer
tokenizer = RagTokenizer.from_pretrained("facebook/rag-token-nq")
```

```
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'DPRQuestionEncoderTokenizer'.
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'DPRQuestionEncoderTokenizerFast'.
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'BartTokenizer'.
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'BartTokenizerFast'.
```

```
# Initialize the tokenizer
tokenizer = RagTokenizer.from_pretrained("facebook/rag-token-nq")
```

```
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'DPRQuestionEncoderTokenizer'.
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'DPRQuestionEncoderTokenizerFast'.
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'BartTokenizer'.
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'BartTokenizerFast'.
```

```
# Initialize the retriever with a pre-built index from the Hugging Face model hub
retriever = RagRetriever.from_pretrained("facebook/rag-token-nq", index_name="exa
```

```
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'DPRQuestionEncoderTokenizer'.
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'DPRQuestionEncoderTokenizerFast'.
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'BartTokenizer'.
The tokenizer class you load from this checkpoint is not the same type as the
The tokenizer class you load from this checkpoint is 'RagTokenizer'.
The class this function is called from is 'BartTokenizerFast'.
```

```
Downloading data files:  82%                                    129/157 [22:15<04:11,  8.97s/it]

Downloading data: 100%                                          545M/545M [00:09<00:00, 74.8MB/s]

Downloading data: 100%                                          546M/546M [00:08<00:00, 77.9MB/s]

Downloading data: 100%                                          546M/546M [00:13<00:00, 13.3MB/s]

Downloading data: 100%                                          546M/546M [00:15<00:00, 78.3MB/s]

Downloading data: 100%                                          546M/546M [00:12<00:00, 73.8MB/s]

Downloading data: 100%                                          545M/545M [00:14<00:00, 68.2MB/s]

Downloading data: 100%                                          544M/544M [00:10<00:00, 72.3MB/s]

Downloading data: 100%                                          537M/537M [00:11<00:00, 54.2MB/s]

Downloading data: 100%                                          530M/530M [00:11<00:00, 26.8MB/s]

Downloading data: 100%                                          538M/538M [00:14<00:00, 58.6MB/s]
```

Downloading data: 100% 546M/546M [00:12<00:00, 41.0MB/s]

Downloading data: 100% 545M/545M [00:09<00:00, 74.8MB/s]

Downloading data: 100% 545M/545M [00:08<00:00, 73.5MB/s]

Downloading data: 100% 545M/545M [00:10<00:00, 79.7MB/s]

Downloading data: 100% 544M/544M [00:08<00:00, 46.9MB/s]

Downloading data: 100% 545M/545M [00:08<00:00, 74.4MB/s]

Downloading data: 100% 545M/545M [00:09<00:00, 79.2MB/s]

Downloading data: 100% 545M/545M [00:08<00:00, 48.0MB/s]

Downloading data: 100% 545M/545M [00:10<00:00, 52.3MB/s]

Downloading data: 100% 544M/544M [00:08<00:00, 66.8MB/s]

Downloading data: 100% 544M/544M [00:08<00:00, 71.0MB/s]

Downloading data: 100% 545M/545M [00:11<00:00, 16.7MB/s]

Downloading data: 100% 545M/545M [00:13<00:00, 62.6MB/s]

Downloading data: 100% 545M/545M [00:10<00:00, 52.0MB/s]

Downloading data: 100% 545M/545M [00:09<00:00, 74.1MB/s]

Downloading data: 100% 545M/545M [00:11<00:00, 64.4MB/s]

Downloading data: 100% 544M/544M [00:09<00:00, 77.0MB/s]

Downloading data: 100% 544M/544M [00:07<00:00, 69.9MB/s]

Downloading data: 100% 544M/544M [00:09<00:00, 75.8MB/s]

Downloading data: 100% 544M/544M [00:10<00:00, 83.1MB/s]

Downloading data: 100% 545M/545M [00:10<00:00, 76.6MB/s]

Downloading data: 100% 545M/545M [00:11<00:00, 76.0MB/s]

Downloading data: 100% 545M/545M [00:07<00:00, 74.7MB/s]

Downloading data: 100% 544M/544M [00:08<00:00, 75.1MB/s]

Downloading data: 100% 544M/544M [00:11<00:00, 68.7MB/s]

Downloading data: 100% 544M/544M [00:11<00:00, 71.9MB/s]

Downloading data: 100% 544M/544M [00:09<00:00, 51.7MB/s]

Downloading data: 100% 544M/544M [00:08<00:00, 74.7MB/s]

Downloading data: 100% 545M/545M [00:11<00:00, 47.6MB/s]

Downloading data: 100% 544M/544M [00:10<00:00, 69.3MB/s]

Downloading data: 100% 544M/544M [00:09<00:00, 65.4MB/s]

Downloading data: 100% 544M/544M [00:07<00:00, 80.0MB/s]

Downloading data: 100% 544M/544M [00:09<00:00, 57.8MB/s]

Downloading data: 100% 544M/544M [00:10<00:00, 74.6MB/s]

Downloading data: 100% 544M/544M [00:11<00:00, 48.2MB/s]

Downloading data: 100% 544M/544M [00:07<00:00, 71.0MB/s]

Downloading data: 100% 544M/544M [00:09<00:00, 71.3MB/s]

Downloading data: 100% 544M/544M [00:13<00:00, 77.6MB/s]

Downloading data: 100% 544M/544M [00:09<00:00, 49.8MB/s]

Downloading data: 100% 544M/544M [00:07<00:00, 74.0MB/s]

Downloading data: 100% 544M/544M [00:08<00:00, 72.8MB/s]

Downloading data: 100% 544M/544M [00:07<00:00, 52.2MB/s]

Downloading data: 100% 544M/544M [00:09<00:00, 51.4MB/s]

Downloading data: 100% 544M/544M [00:08<00:00, 80.2MB/s]

Downloading data: 100% 544M/544M [00:07<00:00, 74.6MB/s]

Downloading data: 100% 543M/543M [00:10<00:00, 74.3MB/s]

Downloading data: 100% 544M/544M [00:10<00:00, 78.2MB/s]

Downloading data: 100% 544M/544M [00:11<00:00, 75.2MB/s]

Downloading data: 100% 544M/544M [00:07<00:00, 48.7MB/s]

Downloading data: 100% 544M/544M [00:12<00:00, 50.6MB/s]

Downloading data: 100% 544M/544M [00:08<00:00, 75.6MB/s]

Downloading data: 100% 543M/543M [00:09<00:00, 72.9MB/s]

Downloading data: 100% 542M/542M [00:11<00:00, 26.2MB/s]

Downloading data: 100% 543M/543M [00:10<00:00, 75.5MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 80.0MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 73.8MB/s]

Downloading data: 100% 544M/544M [00:08<00:00, 68.9MB/s]

Downloading data: 100% 543M/543M [00:11<00:00, 78.8MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 68.0MB/s]

Downloading data: 100% 543M/543M [00:09<00:00, 50.0MB/s]

Downloading data: 100% 543M/543M [00:11<00:00, 46.9MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 71.5MB/s]

Downloading data: 100% 543M/543M [00:11<00:00, 72.6MB/s]

Downloading data: 100% 544M/544M [00:13<00:00, 17.0MB/s]

Downloading data: 100% 543M/543M [00:10<00:00, 52.8MB/s]

Downloading data: 100% 543M/543M [00:11<00:00, 40.9MB/s]

Downloading data: 100% 543M/543M [00:10<00:00, 61.2MB/s]

Downloading data: 100% 542M/542M [00:10<00:00, 64.7MB/s]

Downloading data: 100% 543M/543M [00:11<00:00, 72.7MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 43.4MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 45.1MB/s]

Downloading data: 100% 543M/543M [00:11<00:00, 70.7MB/s]

Downloading data: 100% 543M/543M [00:13<00:00, 48.6MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 74.8MB/s]

Downloading data: 100% 543M/543M [00:11<00:00, 72.4MB/s]

Downloading data: 100% 543M/543M [00:11<00:00, 45.5MB/s]

Downloading data: 100% 543M/543M [00:09<00:00, 74.6MB/s]

Downloading data: 100% 543M/543M [00:11<00:00, 75.2MB/s]

Downloading data: 100% 543M/543M [00:11<00:00, 48.2MB/s]

Downloading data: 100% 542M/542M [00:15<00:00, 76.5MB/s]

Downloading data: 100% 543M/543M [00:09<00:00, 42.7MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 53.6MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 76.1MB/s]

Downloading data: 100% 544M/544M [00:11<00:00, 79.3MB/s]

Downloading data: 100% 543M/543M [00:15<00:00, 40.0MB/s]

Downloading data: 100% 543M/543M [00:10<00:00, 75.2MB/s]

Downloading data: 100% 543M/543M [00:14<00:00, 20.0MB/s]

Downloading data: 100% 542M/542M [00:09<00:00, 73.2MB/s]

Downloading data: 100% 543M/543M [00:09<00:00, 48.8MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 57.6MB/s]

Downloading data: 100% 543M/543M [00:10<00:00, 79.7MB/s]

Downloading data: 100% 543M/543M [00:11<00:00, 68.7MB/s]

Downloading data: 100% 543M/543M [00:10<00:00, 45.2MB/s]

Downloading data: 100% 543M/543M [00:07<00:00, 78.7MB/s]

Downloading data: 100% 542M/542M [00:08<00:00, 71.7MB/s]

Downloading data: 100% 543M/543M [00:09<00:00, 46.8MB/s]

Downloading data: 100% 543M/543M [00:07<00:00, 78.7MB/s]

Downloading data: 100% 543M/543M [00:10<00:00, 76.0MB/s]

Downloading data: 100% 543M/543M [00:12<00:00, 72.1MB/s]

Downloading data: 100% 543M/543M [00:11<00:00, 44.9MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 78.5MB/s]

Downloading data: 100% 542M/542M [00:10<00:00, 75.6MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 74.5MB/s]

Downloading data: 100% 543M/543M [00:12<00:00, 47.4MB/s]

Downloading data: 100% 544M/544M [00:13<00:00, 40.3MB/s]

Downloading data: 100% 543M/543M [00:07<00:00, 72.2MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 73.0MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 45.1MB/s]

Downloading data: 100% 542M/542M [00:08<00:00, 70.0MB/s]

Downloading data: 100% 543M/543M [00:09<00:00, 68.9MB/s]

Downloading data: 100% 543M/543M [00:09<00:00, 52.6MB/s]

Downloading data: 100% 543M/543M [00:07<00:00, 77.1MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 84.6MB/s]

Downloading data: 100% 543M/543M [00:08<00:00, 44.4MB/s]

```
Downloading data: 100%                                  542M/542M [00:10<00:00, 74.9MB/s]

Downloading data: 100%                                  542M/542M [00:09<00:00, 75.7MB/s]

Downloading data: 100%                                  543M/543M [00:09<00:00, 69.8MB/s]

Downloading data: 100%                                  543M/543M [00:07<00:00, 78.2MB/s]

Downloading data: 100%                                  543M/543M [00:08<00:00, 75.2MB/s]

Downloading data:  85%                                  460M/543M [00:09<00:02, 40.3MB/s]
```

```
---------------------------------------------------------------------------
OSError                                   Traceback (most recent call last)
/usr/local/lib/python3.11/dist-packages/datasets/utils/file_utils.py in
temp_file_manager(mode)
    624             with open(incomplete_path, mode) as f:
--> 625                 yield f
    626
```

19 frames

```
OSError: [Errno 28] No space left on device

During handling of the above exception, another exception occurred:

OSError                                   Traceback (most recent call last)
/usr/local/lib/python3.11/dist-packages/datasets/utils/file_utils.py in
temp_file_manager(mode)
    622         @contextmanager
    623         def temp_file_manager(mode="w+b"):
--> 624             with open(incomplete_path, mode) as f:
    625                 yield f
    626

OSError: [Errno 28] No space left on device
```

Next steps: Explain error

Next steps:

Given the memory constraints on Colab, the code

```
retriever = RagRetriever.from_pretrained("facebook/rag-token-nq", index_name="ex
```

won't complete running.

Alternative Solution: Upgrade to Colab Pro

Upgrade to Colab Pro for more RAM and GPU resources by clicking on the `Upgrade to Pro` button in the top-right corner of your Colab notebook.

```
# Initialize the RAG model
model = RagSequenceForGeneration.from_pretrained("facebook/rag-token-nq", retrie

# Example query
query = "What is the capital of France?"

# Tokenize the query
input_ids = tokenizer(query, return_tensors="pt").input_ids

# Retrieval Phase
# The retriever fetches relevant documents based on the query
retrieved_docs = retriever(input_ids)

# Generation Phase
# The model generates a response using the retrieved documents and the query
output = model.generate(input_ids, num_return_sequences=1)

# Decode the generated response
generated_text = tokenizer.batch_decode(output, skip_special_tokens=True)

print("Generated Text:", generated_text[0])
```

Explanation of the Code

1. **Tokenizer**: The `RagTokenizer` is used to tokenize the input query.
2. **Retriever**: The `RagRetriever` is used to fetch relevant documents from a pre-built index. In this example, we use the "condense" index, which is optimized for faster retrieval.
3. **Model**: The `RagSequenceForGeneration` model combines the retrieval and generation steps. It takes the tokenized query and the retrieved documents to generate a response.
4. **Query**: The input query is tokenized and passed to the retriever.
5. **Retrieval**: The retriever fetches relevant documents based on the query.
6. **Generation**: The model generates a response based on the query and the retrieved documents.
7. **Decoding**: The generated response is decoded back into human-readable text.

Notes

- This example uses a pre-trained RAG model and retriever from the Hugging Face model hub. You can also fine-tune these models on your own dataset for better performance.
- The retriever in this example uses a "condense" index for faster retrieval. In practice, you might want to experiment with different retrieval models like Dense Passage Retrieval (DPR) for better performance.
- The `num_return_sequences` parameter

Conclusion

In this notebook, we have walked through the process of initializing and running a Retrieval-Augmented Generation (RAG) model using the Hugging Face Transformers library. We addressed memory constraints on Google Colab by providing memory management techniques and suggesting alternative solutions, such as upgrading to Colab Pro. This approach ensures smooth execution and helps in generating high-quality, contextually relevant responses.

However, we encountered an `OSError: [Errno 28] No space left on device` error, indicating that the available disk space was insufficient to complete the operation. This issue highlights the limitations of the free tier of Google Colab, especially for memory-intensive tasks.

Next Steps

To mitigate this issue, consider the following steps:

1. **Upgrade to Colab Pro**:
    - Upgrading to Colab Pro or Colab Pro+ provides more RAM and disk space, which can help in handling larger models and datasets.
2. **Use a Smaller Dataset**:
    - Reduce the size of the dataset to fit within the available memory and disk space. This can help in completing the operation without running into space constraints.
3. **Clear Unused Variables**:
    - Regularly clear unused variables and free up memory using `gc.collect()` and `torch.cuda.empty_cache()` to manage memory more effectively.
4. **Batch Processing**:
    - Process the data in smaller batches rather than all at once. This can help in managing memory and disk space more effectively.

Next, I will upload the RAG model on a smaller dataset to further illustrate its capabilities and efficiency.