```
!pip install --upgrade --force-reinstall numpy
```

```
Collecting numpy
  Downloading numpy-2.3.4-cp312-cp312-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl.metadata (62 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.1/62.1 kB 2.4 MB/s eta 0:00:00
Downloading numpy-2.3.4-cp312-cp312-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl (16.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.6/16.6 MB 26.7 MB/s eta 0:00:00
Installing collected packages: numpy
  Attempting uninstall: numpy
    Found existing installation: numpy 2.0.2
    Uninstalling numpy-2.0.2:
      Successfully uninstalled numpy-2.0.2
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This b
tensorflow 2.19.0 requires numpy<2.2.0,>=1.26.0, but you have numpy 2.3.4 which is incompatible.
opencv-python 4.12.0.88 requires numpy<2.3.0,>=2; python_version >= "3.9", but you have numpy 2.3.4 which is inco
cupy-cuda12x 13.3.0 requires numpy<2.3,>=1.22, but you have numpy 2.3.4 which is incompatible.
opencv-contrib-python 4.12.0.88 requires numpy<2.3.0,>=2; python_version >= "3.9", but you have numpy 2.3.4 which
opencv-python-headless 4.12.0.88 requires numpy<2.3.0,>=2; python_version >= "3.9", but you have numpy 2.3.4 whic
numba 0.60.0 requires numpy<2.1,>=1.22, but you have numpy 2.3.4 which is incompatible.
Successfully installed numpy-2.3.4
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force
```

```
!CMAKE_ARGS="-DLLAMA_CUBLAS=on" FORCE_CMAKE=1 pip install llama-cpp-python==0.2.28  --force-reinstall --upgrade -
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.4/9.4 MB 2.7 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.1/62.1 kB 250.0 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 45.5/45.5 kB 218.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.6/16.6 MB 200.3 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.6/44.6 kB 210.5 MB/s eta 0:00:00
  Building wheel for llama-cpp-python (pyproject.toml) ... done
```

```
!pip install tiktoken pypdf langchain langchain-community chromadb sentence-transformers huggingface_hub
```

```
Requirement already satisfied: tiktoken in /usr/local/lib/python3.12/dist-packages (0.12.0)
Requirement already satisfied: pypdf in /usr/local/lib/python3.12/dist-packages (6.1.3)
Requirement already satisfied: langchain in /usr/local/lib/python3.12/dist-packages (0.3.27)
Collecting langchain-community
  Using cached langchain_community-0.4.1-py3-none-any.whl.metadata (3.0 kB)
Collecting chromadb
  Using cached chromadb-1.3.4-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (7.2 kB)
Requirement already satisfied: sentence-transformers in /usr/local/lib/python3.12/dist-packages (5.1.2)
Requirement already satisfied: huggingface_hub in /usr/local/lib/python3.12/dist-packages (0.36.0)
Requirement already satisfied: regex>=2022.1.18 in /usr/local/lib/python3.12/dist-packages (from tiktoken) (2024
Requirement already satisfied: requests>=2.26.0 in /usr/local/lib/python3.12/dist-packages (from tiktoken) (2.32
Requirement already satisfied: langchain-core<1.0.0,>=0.3.72 in /usr/local/lib/python3.12/dist-packages (from la
Requirement already satisfied: langchain-text-splitters<1.0.0,>=0.3.9 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: langsmith>=0.1.17 in /usr/local/lib/python3.12/dist-packages (from langchain) (0
Requirement already satisfied: pydantic<3.0.0,>=2.7.4 in /usr/local/lib/python3.12/dist-packages (from langchain
Requirement already satisfied: SQLAlchemy<3,>=1.4 in /usr/local/lib/python3.12/dist-packages (from langchain) (2
Requirement already satisfied: PyYAML>=5.3 in /usr/local/lib/python3.12/dist-packages (from langchain) (6.0.3)
INFO: pip is looking at multiple versions of langchain-community to determine which version is compatible with o
Collecting langchain-community
  Using cached langchain_community-0.4-py3-none-any.whl.metadata (3.0 kB)
  Using cached langchain_community-0.3.31-py3-none-any.whl.metadata (3.0 kB)
Collecting requests>=2.26.0 (from tiktoken)
  Using cached requests-2.32.5-py3-none-any.whl.metadata (4.9 kB)
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in /usr/local/lib/python3.12/dist-packages (from langchain-
Requirement already satisfied: tenacity!=8.4.0,<10.0.0,>=8.1.0 in /usr/local/lib/python3.12/dist-packages (from
Collecting dataclasses-json<0.7.0,>=0.6.7 (from langchain-community)
  Using cached dataclasses_json-0.6.7-py3-none-any.whl.metadata (25 kB)
Requirement already satisfied: pydantic-settings<3.0.0,>=2.10.1 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: httpx-sse<1.0.0,>=0.4.0 in /usr/local/lib/python3.12/dist-packages (from langchai
Requirement already satisfied: numpy>=1.26.2 in /usr/local/lib/python3.12/dist-packages (from langchain-communit
Requirement already satisfied: build>=1.0.3 in /usr/local/lib/python3.12/dist-packages (from chromadb) (1.3.0)
Requirement already satisfied: pybase64>=1.4.1 in /usr/local/lib/python3.12/dist-packages (from chromadb) (1.4.2
Requirement already satisfied: uvicorn>=0.18.3 in /usr/local/lib/python3.12/dist-packages (from uvicorn[standard
Collecting posthog<6.0.0,>=2.4.0 (from chromadb)
  Using cached posthog-5.4.0-py3-none-any.whl.metadata (5.7 kB)
Requirement already satisfied: typing-extensions>=4.5.0 in /usr/local/lib/python3.12/dist-packages (from chromad
Collecting onnxruntime>=1.14.1 (from chromadb)
  Using cached onnxruntime-1.23.2-cp312-cp312-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl.metadata (5.1 kB)
Requirement already satisfied: opentelemetry-api>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from chromad
Collecting opentelemetry-exporter-otlp-proto-grpc>=1.2.0 (from chromadb)
  Using cached opentelemetry_exporter_otlp_proto_grpc-1.38.0-py3-none-any.whl.metadata (2.4 kB)
```

```
Requirement already satisfied: opentelemetry-sdk>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from chroma
Requirement already satisfied: tokenizers>=0.13.2 in /usr/local/lib/python3.12/dist-packages (from chromadb) (0
Requirement already satisfied: pypika>=0.48.9 in /usr/local/lib/python3.12/dist-packages (from chromadb) (0.48.
Requirement already satisfied: tqdm>=4.65.0 in /usr/local/lib/python3.12/dist-packages (from chromadb) (4.67.1)
Requirement already satisfied: overrides>=7.3.1 in /usr/local/lib/python3.12/dist-packages (from chromadb) (7.7
Requirement already satisfied: importlib-resources in /usr/local/lib/python3.12/dist-packages (from chromadb) (
Requirement already satisfied: grpcio>=1.58.0 in /usr/local/lib/python3.12/dist-packages (from chromadb) (1.76.
Requirement already satisfied: bcrypt>=4.0.1 in /usr/local/lib/python3.12/dist-packages (from chromadb) (5.0.0)
Requirement already satisfied: typer>=0.9.0 in /usr/local/lib/python3.12/dist-packages (from chromadb) (0.20.0)
Collecting kubernetes>=28.1.0 (from chromadb)
  Using cached kubernetes-34.1.0-py2.py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: mmh3>=4.0.1 in /usr/local/lib/python3.12/dist-packages (from chromadb) (5.2.0)
Requirement already satisfied: orjson>=3.9.12 in /usr/local/lib/python3.12/dist-packages (from chromadb) (3.11.
Requirement already satisfied: httpx>=0.27.0 in /usr/local/lib/python3.12/dist-packages (from chromadb) (0.28.1
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.12/dist-packages (from chromadb) (13.9.4
Requirement already satisfied: jsonschema>=4.19.0 in /usr/local/lib/python3.12/dist-packages (from chromadb) (4
```

```
import json
import tiktoken
import pandas as pd
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain_community.document_loaders import PyPDFDirectoryLoader, PyPDFLoader
from langchain_community.embeddings.sentence_transformer import SentenceTransformerEmbeddings
from langchain_community.vectorstores import Chroma
from google.colab import userdata, drive
```

```
pdf_path = "/content/drive/MyDrive/Meta-2024-Sustainability-Report.pdf"
```

```
pdf_loader = PyPDFLoader(pdf_path)
```

```
pdf = pdf_loader.load()
```

```
for i in range(3):
    print(f"Page Number : {i+1}",end="\n")
    print(pdf[i].page_content,end="\n")
```

```
Page Number : 1
FOR A BETTER REALITY
For a better reality
2024 SUSTAINABILITY REPORT
Page Number : 2
Table of contents
3  Leadership messages

 3 Rachel Peterson
 5 Blair Swedeen
  and Leslie Collins

7  Executive summary
 8 About Meta
 9 About this report

 11 Sustainability vision

 17 Goals and commitments
19  How we operate
 20 Path to net zero
 26 Energy
 39 Data centers

 45 Climate risks and resilience
 47 Water
 52 Offices
 54 Responsible supply chain
2024 Sustainability Report
66  What we create
 67 Responsible AI
 68 AI for climate
 73 Climate insights
A Data index
 B Forward-looking
  statements
 C Environmental metrics
 O Environmental
  methodology
Page Number : 3
In early 2024, we celebrated the 20th anniversary of the launch of Facebook, coming off a year
focused on efficiency and artificial intelligence (AI) in 2023. We challenged ourselves to uncover new
opportunities to streamline our processes and deliver the additional data center capacity to support
the AI demand, while keeping our eye on our sustainability progress.
Since 2020, we have maintained net zero emissions in our global operations. To get there we
reduced our emissions by 94% from a 2017 baseline, primarily by matching 100% of the electricity
use of our data centers and offices with renewable energy and addressing the remaining emissions
with projects that remove carbon from the atmosphere.
Meeting our goal to achieve net zero value chain emissions in 2030 will be significantly harder. The
```

```
challenge of reaching our sustainability goals given the increased demand for energy and resources
driven by AI is not unique to Meta. And it will require major shifts in how companies like ours operate.
Since breaking ground on our first data center back in 2010, we've built a global infrastructure that
serves as the engine for the more than three billion people who use our technologies and programs
every day. AI has been an important part of these systems for many years. (CONT.)
 A message from
VICE PRESIDENT,
INFRASTRUCTURE DATA CENTERS
Rachel
```

```
pdf[5].page_content
```

```
'Meta strives to have a positive impact on \nthe communities and environment where we \noperate and collaborate
to scale our knowledge \nacross the industry. Our size and scale enable us \nto influence the future of decarbon
ization, the \nexpansion of renewable energy and the integrity \nof climate solutions  — and to help others in
\nthe industry benefit from the resources we've \ndeveloped through decades of learning  \nand improvement.\nOur
investment as a founding member of the \nClean Energy Procurement Academy (CEPA) \nis one such example. Launched
to support \nthe decarbonization of global supply chains, \nCEPA encourages renewable energy purchases \nthrough
education. \nMeta worked with five other leading \ncorporations to create a shared training \ncurriculum to educ
ate suppliers on the benefits \nof renewable energy purchases. \nThis report details how we are working toward
```

```
len(pdf)
```

```
94
```

```
text_splitter = RecursiveCharacterTextSplitter.from_tiktoken_encoder(
    encoding_name='cl100k_base',
    chunk_size=512,
    chunk_overlap= 20
)
```

```
document_chunks = pdf_loader.load_and_split(text_splitter)
```

```
len(document_chunks)
```

```
105
```

```
document_chunks[-2].page_content
```

```
'additional 1,780,000 cubic meters of water withdrawn for the construction of Meta data centers.\nWater consumpt
ion\nFor our data centers, we determine our water consumption via two methods:\n1. Calculating the difference be
tween water withdrawal and wastewater discharge\n2. Calculating consumption based on cycles of concentration fro
m our cooling systems\nFor our offices, we estimate our water consumption based on industry averages. All of our
wastewater is discharged to local wastewater \nfacilities.\nWater risk\nWe use water stress metrics in the World
Resources Institute's Aqueduct tool ↗ to conduct initial assessments of our water risks. When \nappropriate, we
increase the level of water risk based on additional local knowledge.\nLeadership messages Executive summary How
```

```
document_chunks[0].page_content
```

```
'FOR A BETTER REALITY\nFor a better reality\n2024 SUSTAINABILITY REPORT'
```

```
embedding_model = SentenceTransformerEmbeddings(model_name='thenlper/gte-large')
```

```
/tmp/ipython-input-4198310515.py:1: LangChainDeprecationWarning: The class `HuggingFaceEmbeddings` was deprecated
  embedding_model = SentenceTransformerEmbeddings(model_name='thenlper/gte-large')
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
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 (https://huggingface.co/settings/t
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
```

```
modules.json: 100%                                  385/385 [00:00<00:00, 32.8kB/s]
README.md:      67.9k/? [00:00<00:00, 4.75MB/s]
sentence_bert_config.json: 100%                               57.0/57.0 [00:00<00:00, 2.93kB/s]
config.json: 100%                                  619/619 [00:00<00:00, 45.4kB/s]
model.safetensors: 100%                                  670M/670M [00:09<00:00, 48.2MB/s]
tokenizer_config.json: 100%                                  342/342 [00:00<00:00, 15.4kB/s]
vocab.txt:      232k/? [00:00<00:00, 8.82MB/s]
tokenizer.json:      712k/? [00:00<00:00, 27.8MB/s]
special_tokens_map.json: 100%                                  125/125 [00:00<00:00, 11.5kB/s]
config.json: 100%                                  191/191 [00:00<00:00, 13.9kB/s]
```

```
embedding_1 = embedding_model.embed_query(document_chunks[0].page_content)
embedding_2 = embedding_model.embed_query(document_chunks[1].page_content)
print("Dimension of the embedding vector ",len(embedding_1))
len(embedding_1)==len(embedding_2)
```

```
Dimension of the embedding vector  1024
True
```

```
import os
out_dir = 'apple_db'

if not os.path.exists(out_dir):
  os.makedirs(out_dir)
```

```
vectorstore = Chroma.from_documents(
    document_chunks,
    embedding_model,
    persist_directory=out_dir
)
```

```
vectorstore = Chroma(persist_directory=out_dir,embedding_function=embedding_model)
```

```
/tmp/ipython-input-2756559696.py:1: LangChainDeprecationWarning: The class `Chroma` was deprecated in LangChain 0
  vectorstore = Chroma(persist_directory=out_dir,embedding_function=embedding_model)
```

```
vectorstore.embeddings
```

```
HuggingFaceEmbeddings(client=SentenceTransformer(
  (0): Transformer({'max_seq_length': 512, 'do_lower_case': False, 'architecture': 'BertModel'})
  (1): Pooling({'word_embedding_dimension': 1024, 'pooling_mode_cls_token': False, 'pooling_mode_mean_tokens':
True, 'pooling_mode_max_tokens': False, 'pooling_mode_mean_sqrt_len_tokens': False,
'pooling_mode_weightedmean_tokens': False, 'pooling_mode_lasttoken': False, 'include_prompt': True})
  (2): Normalize()
), model_name='thenlper/gte-large', cache_folder=None, model_kwargs={}, encode_kwargs={}, multi_process=False,
show_progress=False)
```

```
vectorstore.similarity_search("Apple Steve Jobs iPhone ",k=3)
```

```
[]
```

```
retriever = vectorstore.as_retriever(
    search_type='similarity',
    search_kwargs={'k': 2}
)
```

```
rel_docs = retriever.get_relevant_documents("How does does Apple develop and ship products that requires good co
rel_docs
```

```
/tmp/ipython-input-3586710401.py:1: LangChainDeprecationWarning: The method `BaseRetriever.get_relevant_documents
  rel_docs = retriever.get_relevant_documents("How does does Apple develop and ship products that requires good c
[]
```

```
!pip uninstall -y llama-cpp-python
!CMAKE_ARGS="-DLLAMA_CUBLAS=off" pip install llama-cpp-python --no-cache-dir
```

```
Found existing installation: llama_cpp_python 0.2.28
Uninstalling llama_cpp_python-0.2.28:
  Successfully uninstalled llama_cpp_python-0.2.28
Collecting llama-cpp-python
  Downloading llama_cpp_python-0.3.16.tar.gz (50.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 50.7/50.7 MB 87.2 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: typing-extensions>=4.5.0 in /usr/local/lib/python3.12/dist-packages (from llama-cp
Requirement already satisfied: numpy>=1.20.0 in /usr/local/lib/python3.12/dist-packages (from llama-cpp-python) (
Requirement already satisfied: diskcache>=5.6.1 in /usr/local/lib/python3.12/dist-packages (from llama-cpp-python
Requirement already satisfied: jinja2>=2.11.3 in /usr/local/lib/python3.12/dist-packages (from llama-cpp-python)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2>=2.11.3->l
Building wheels for collected packages: llama-cpp-python
  Building wheel for llama-cpp-python (pyproject.toml) ... done
  Created wheel for llama-cpp-python: filename=llama_cpp_python-0.3.16-cp312-cp312-linux_x86_64.whl size=4422316
  Stored in directory: /tmp/pip-ephem-wheel-cache-h7rgvlqc/wheels/90/82/ab/8784ee3fb99ddb07fd36a679ddbe63122cc077
Successfully built llama-cpp-python
Installing collected packages: llama-cpp-python
Successfully installed llama-cpp-python-0.3.16
```

```
from llama_cpp import Llama
```

```
model_name_or_path = "TheBloke/Mistral-7B-Instruct-v0.2-GGUF"
model_basename = "mistral-7b-instruct-v0.2.Q6_K.gguf"
model_path = hf_hub_download(
    repo_id=model_name_or_path,
    filename=model_basename
)
```

```
mistral-7b-instruct-v0.2.Q6_K.gguf: 100%                                  5.94G/5.94G [01:32<00:00, 105MB/s]
```

```
llm = Llama(
    model_path=model_path,
    n_ctx=2300,
    n_gpu_layers=38,
    n_batch=512
)
```

```
llama_model_loader: loaded meta data with 24 key-value pairs and 291 tensors from /root/.cache/huggingface/hub/
llama_model_loader: Dumping metadata keys/values. Note: KV overrides do not apply in this output.
llama_model_loader: - kv   0:                       general.architecture str              = llama
llama_model_loader: - kv   1:                               general.name str              = mistralai_mistral-7
llama_model_loader: - kv   2:                       llama.context_length u32              = 32768
llama_model_loader: - kv   3:                     llama.embedding_length u32              = 4096
llama_model_loader: - kv   4:                          llama.block_count u32              = 32
llama_model_loader: - kv   5:                  llama.feed_forward_length u32              = 14336
llama_model_loader: - kv   6:                 llama.rope.dimension_count u32              = 128
llama_model_loader: - kv   7:                 llama.attention.head_count u32              = 32
llama_model_loader: - kv   8:              llama.attention.head_count_kv u32              = 8
llama_model_loader: - kv   9:     llama.attention.layer_norm_rms_epsilon f32              = 0.000010
llama_model_loader: - kv  10:                       llama.rope.freq_base f32              = 1000000.000000
llama_model_loader: - kv  11:                          general.file_type u32              = 18
llama_model_loader: - kv  12:                       tokenizer.ggml.model str              = llama
llama_model_loader: - kv  13:                      tokenizer.ggml.tokens arr[str,32000]   = ["<unk>", "<s>", "<
llama_model_loader: - kv  14:                      tokenizer.ggml.scores arr[f32,32000]   = [0.000000, 0.000000
llama_model_loader: - kv  15:                  tokenizer.ggml.token_type arr[i32,32000]   = [2, 3, 3, 6, 6, 6, 
llama_model_loader: - kv  16:                tokenizer.ggml.bos_token_id u32              = 1
llama_model_loader: - kv  17:                tokenizer.ggml.eos_token_id u32              = 2
llama_model_loader: - kv  18:            tokenizer.ggml.unknown_token_id u32              = 0
llama_model_loader: - kv  19:            tokenizer.ggml.padding_token_id u32              = 0
llama_model_loader: - kv  20:               tokenizer.ggml.add_bos_token bool             = true
llama_model_loader: - kv  21:               tokenizer.ggml.add_eos_token bool             = false
llama_model_loader: - kv  22:                    tokenizer.chat_template str              = {{ bos_token }}{% f
llama_model_loader: - kv  23:               general.quantization_version u32              = 2
llama_model_loader: - type  f32:   65 tensors
llama_model_loader: - type q6_K:  226 tensors
print_info: file format = GGUF V3 (latest)
print_info: file type   = Q6_K
print_info: file size   = 5.53 GiB (6.56 BPW)
init_tokenizer: initializing tokenizer for type 1
load: control token:      2 '</s>' is not marked as EOG
load: control token:      1 '<s>' is not marked as EOG
load: special_eos_id is not in special_eog_ids - the tokenizer config may be incorrect
load: printing all EOG tokens:
load:   - 2 ('</s>')
load: special tokens cache size = 3
load: token to piece cache size = 0.1637 MB
print_info: arch             = llama
print_info: vocab_only       = 0
print_info: n_ctx_train      = 32768
print_info: n_embd           = 4096
print_info: n_layer          = 32
print_info: n_head           = 32
print_info: n_head_kv        = 8
print_info: n_rot            = 128
print_info: n_swa            = 0
print_info: is_swa_any       = 0
print_info: n_embd_head_k    = 128
print_info: n_embd_head_v    = 128
print_info: n_gqa            = 4
print_info: n_embd_k_gqa     = 1024
print_info: n_embd_v_gqa     = 1024
print_info: f_norm_eps       = 0.0e+00
print_info: f_norm_rms_eps   = 1.0e-05
print_info: f_clamp_kqv      = 0.0e+00
print_info: f_max_alibi_bias = 0.0e+00
```

```
def generate_rag_response(user_input,k=3,max_tokens=128,temperature=0,top_p=0.95,top_k=50):
    global qna_system_message,qna_user_message_template
    relevant_document_chunks = retriever.get_relevant_documents(query=user_input,k=k)
    context_list = [d.page_content for d in relevant_document_chunks]
    context_for_query = ". ".join(context_list)

    user_message = qna_user_message_template.replace('{context}', context_for_query)
    user_message = user_message.replace('{question}', user_input)

    prompt = qna_system_message + '\n' + user_message
    try:
        response = llm(
                  prompt=prompt,
                  max_tokens=max_tokens,
                  temperature=temperature,
                  top_p=top_p,
                  top_k=top_k
                  )

        response = response['choices'][0]['text'].strip()
    except Exception as e:
        response = f'Sorry, I encountered the following error: \n {e}'

    return response
```

```
def generate_ground_relevance_response(user_input,k=3,max_tokens=128,temperature=0,top_p=0.95,top_k=50):
    global qna_system_message,qna_user_message_template
    relevant_document_chunks = retriever.get_relevant_documents(query=user_input,k=3)
    context_list = [d.page_content for d in relevant_document_chunks]
    context_for_query = ". ".join(context_list)
    prompt = f"""[INST]{qna_system_message}\n
                {'user'}: {qna_user_message_template.format(context=context_for_query, question=user_input)}
                [/INST]"""

    response = llm(
            prompt=prompt,
            max_tokens=max_tokens,
            temperature=temperature,
            top_p=top_p,
            top_k=top_k,
            stop=['INST'],
            )

    answer =  response["choices"][0]["text"]
    groundedness_prompt = f"""[INST]{groundedness_rater_system_message}\n
                {'user'}: {user_message_template.format(context=context_for_query, question=user_input, answer=a
                [/INST]"""


    relevance_prompt = f"""[INST]{relevance_rater_system_message}\n
                {'user'}: {user_message_template.format(context=context_for_query, question=user_input, answer=a
                [/INST]"""

    response_1 = llm(
            prompt=groundedness_prompt,
            max_tokens=max_tokens,
            temperature=temperature,
            top_p=top_p,
            top_k=top_k,
            stop=['INST'],
            )

    response_2 = llm(
            prompt=relevance_prompt,
            max_tokens=max_tokens,
            temperature=temperature,
            top_p=top_p,
            top_k=top_k,
            stop=['INST'],
            )

    return response_1['choices'][0]['text'],response_2['choices'][0]['text']
```

```
Start coding or generate with AI.
```