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

Collecting numpy
  Using cached numpy-2.3.4-cp312-cp312-
manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl.metadata (62 kB)
Using cached numpy-2.3.4-cp312-cp312-
manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl (16.6 MB)
Installing collected packages: numpy
  Attempting uninstall: numpy
    Found existing installation: numpy 2.3.4
    Uninstalling numpy-2.3.4:
      Successfully uninstalled numpy-2.3.4
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
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 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 incompatible.
tensorflow 2.19.0 requires numpy<2.2.0,>=1.26.0, but you have numpy
2.3.4 which is incompatible.
opencv-python-headless 4.12.0.88 requires numpy<2.3.0,>=2;
python_version >= "3.9", but you have numpy 2.3.4 which is
incompatible.
numba 0.60.0 requires numpy<2.1,>=1.22, but you have numpy 2.3.4 which
is incompatible.
cupy-cuda12x 13.3.0 requires numpy<2.3,>=1.22, but you have numpy
2.3.4 which is incompatible.
Successfully installed numpy-2.3.4

{"id":"3c6472f561a2447297929d3d48033842","pip_warning":{"packages":
["numpy"]}}

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_remount=True).

!CMAKE_ARGS="-DLLAMA_CUBLAS=on" FORCE_CMAKE=1 pip install llama-cpp-
python==0.2.28  --force-reinstall --upgrade --no-cache-dir -q
2>/dev/null

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/9.4 MB ? eta -:--:--
━━━━╸━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/9.4 MB 35.3 MB/s eta
0:00:01 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╸━━━━━━━━ 7.1/9.4 MB 103.1 MB/s
eta 0:00:01 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.4/9.4 MB 118.5
MB/s eta 0:00:00
ents to build wheel ... etadata (pyproject.toml) ...
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.1/62.1 kB 207.7 MB/s eta
```

```
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 45.5/45.5 kB 233.3 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.6/16.6 MB 256.4 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.6/44.6 kB 243.6 MB/s eta
0:00:00
a-cpp-python (pyproject.toml) ...

!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)
Requirement already satisfied: langchain-community in
/usr/local/lib/python3.12/dist-packages (0.3.31)
Requirement already satisfied: chromadb in
/usr/local/lib/python3.12/dist-packages (1.3.4)
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) (2024.11.6)
Requirement already satisfied: requests>=2.26.0 in
/usr/local/lib/python3.12/dist-packages (from tiktoken) (2.32.5)
Requirement already satisfied: langchain-core<1.0.0,>=0.3.72 in
/usr/local/lib/python3.12/dist-packages (from langchain) (0.3.79)
Requirement already satisfied: langchain-text-splitters<1.0.0,>=0.3.9
in /usr/local/lib/python3.12/dist-packages (from langchain) (0.3.11)
Requirement already satisfied: langsmith>=0.1.17 in
/usr/local/lib/python3.12/dist-packages (from langchain) (0.4.38)
Requirement already satisfied: pydantic<3.0.0,>=2.7.4 in
/usr/local/lib/python3.12/dist-packages (from langchain) (2.11.10)
Requirement already satisfied: SQLAlchemy<3,>=1.4 in
/usr/local/lib/python3.12/dist-packages (from langchain) (2.0.44)
Requirement already satisfied: PyYAML>=5.3 in
/usr/local/lib/python3.12/dist-packages (from langchain) (6.0.3)
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in
/usr/local/lib/python3.12/dist-packages (from langchain-community)
(3.13.1)
Requirement already satisfied: tenacity!=8.4.0,<10.0.0,>=8.1.0 in
/usr/local/lib/python3.12/dist-packages (from langchain-community)
(8.5.0)
Requirement already satisfied: dataclasses-json<0.7.0,>=0.6.7 in
/usr/local/lib/python3.12/dist-packages (from langchain-community)
```

```
(0.6.7)
Requirement already satisfied: pydantic-settings<3.0.0,>=2.10.1 in
/usr/local/lib/python3.12/dist-packages (from langchain-community)
(2.11.0)
Requirement already satisfied: httpx-sse<1.0.0,>=0.4.0 in
/usr/local/lib/python3.12/dist-packages (from langchain-community)
(0.4.3)
Requirement already satisfied: numpy>=1.26.2 in
/usr/local/lib/python3.12/dist-packages (from langchain-community)
(2.3.4)
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]>=0.18.3->chromadb) (0.38.0)
Requirement already satisfied: posthog<6.0.0,>=2.4.0 in
/usr/local/lib/python3.12/dist-packages (from chromadb) (5.4.0)
Requirement already satisfied: typing-extensions>=4.5.0 in
/usr/local/lib/python3.12/dist-packages (from chromadb) (4.15.0)
Requirement already satisfied: onnxruntime>=1.14.1 in
/usr/local/lib/python3.12/dist-packages (from chromadb) (1.23.2)
Requirement already satisfied: opentelemetry-api>=1.2.0 in
/usr/local/lib/python3.12/dist-packages (from chromadb) (1.38.0)
Requirement already satisfied: opentelemetry-exporter-otlp-proto-
grpc>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from chromadb)
(1.38.0)
Requirement already satisfied: opentelemetry-sdk>=1.2.0 in
/usr/local/lib/python3.12/dist-packages (from chromadb) (1.38.0)
Requirement already satisfied: tokenizers>=0.13.2 in
/usr/local/lib/python3.12/dist-packages (from chromadb) (0.22.1)
Requirement already satisfied: pypika>=0.48.9 in
/usr/local/lib/python3.12/dist-packages (from chromadb) (0.48.9)
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) (7.7.0)
Requirement already satisfied: importlib-resources in
/usr/local/lib/python3.12/dist-packages (from chromadb) (6.5.2)
Requirement already satisfied: grpcio>=1.58.0 in
/usr/local/lib/python3.12/dist-packages (from chromadb) (1.76.0)
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)
Requirement already satisfied: kubernetes>=28.1.0 in
/usr/local/lib/python3.12/dist-packages (from chromadb) (34.1.0)
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) (3.11.4)
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) (4.25.1)
Requirement already satisfied: transformers<5.0.0,>=4.41.0 in
/usr/local/lib/python3.12/dist-packages (from sentence-transformers)
(4.57.1)
Requirement already satisfied: torch>=1.11.0 in
/usr/local/lib/python3.12/dist-packages (from sentence-transformers)
(2.8.0+cu126)
Requirement already satisfied: scikit-learn in
/usr/local/lib/python3.12/dist-packages (from sentence-transformers)
(1.6.1)
Requirement already satisfied: scipy in
/usr/local/lib/python3.12/dist-packages (from sentence-transformers)
(1.16.3)
Requirement already satisfied: Pillow in
/usr/local/lib/python3.12/dist-packages (from sentence-transformers)
(11.3.0)
Requirement already satisfied: filelock in
/usr/local/lib/python3.12/dist-packages (from huggingface_hub)
(3.20.0)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.12/dist-packages (from huggingface_hub)
(2025.3.0)
Requirement already satisfied: packaging>=20.9 in
/usr/local/lib/python3.12/dist-packages (from huggingface_hub) (25.0)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in
/usr/local/lib/python3.12/dist-packages (from huggingface_hub) (1.2.0)
Requirement already satisfied: aiohappyeyeballs>=2.5.0 in
/usr/local/lib/python3.12/dist-packages (from aiohttp<4.0.0,>=3.8.3-
>langchain-community) (2.6.1)
Requirement already satisfied: aiosignal>=1.4.0 in
/usr/local/lib/python3.12/dist-packages (from aiohttp<4.0.0,>=3.8.3-
>langchain-community) (1.4.0)
Requirement already satisfied: attrs>=17.3.0 in
/usr/local/lib/python3.12/dist-packages (from aiohttp<4.0.0,>=3.8.3-
>langchain-community) (25.4.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.12/dist-packages (from aiohttp<4.0.0,>=3.8.3-
>langchain-community) (1.8.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.12/dist-packages (from aiohttp<4.0.0,>=3.8.3-
>langchain-community) (6.7.0)
```

```
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.12/dist-packages (from aiohttp<4.0.0,>=3.8.3-
>langchain-community) (0.4.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in
/usr/local/lib/python3.12/dist-packages (from aiohttp<4.0.0,>=3.8.3-
>langchain-community) (1.22.0)
Requirement already satisfied: pyproject_hooks in
/usr/local/lib/python3.12/dist-packages (from build>=1.0.3->chromadb)
(1.2.0)
Requirement already satisfied: marshmallow<4.0.0,>=3.18.0 in
/usr/local/lib/python3.12/dist-packages (from dataclasses-
json<0.7.0,>=0.6.7->langchain-community) (3.26.1)
Requirement already satisfied: typing-inspect<1,>=0.4.0 in
/usr/local/lib/python3.12/dist-packages (from dataclasses-
json<0.7.0,>=0.6.7->langchain-community) (0.9.0)
Requirement already satisfied: anyio in
/usr/local/lib/python3.12/dist-packages (from httpx>=0.27.0->chromadb)
(4.11.0)
Requirement already satisfied: certifi in
/usr/local/lib/python3.12/dist-packages (from httpx>=0.27.0->chromadb)
(2025.10.5)
Requirement already satisfied: httpcore==1.* in
/usr/local/lib/python3.12/dist-packages (from httpx>=0.27.0->chromadb)
(1.0.9)
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-
packages (from httpx>=0.27.0->chromadb) (3.11)
Requirement already satisfied: h11>=0.16 in
/usr/local/lib/python3.12/dist-packages (from httpcore==1.*-
>httpx>=0.27.0->chromadb) (0.16.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/usr/local/lib/python3.12/dist-packages (from jsonschema>=4.19.0-
>chromadb) (2025.9.1)
Requirement already satisfied: referencing>=0.28.4 in
/usr/local/lib/python3.12/dist-packages (from jsonschema>=4.19.0-
>chromadb) (0.37.0)
Requirement already satisfied: rpds-py>=0.7.1 in
/usr/local/lib/python3.12/dist-packages (from jsonschema>=4.19.0-
>chromadb) (0.28.0)
Requirement already satisfied: six>=1.9.0 in
/usr/local/lib/python3.12/dist-packages (from kubernetes>=28.1.0-
>chromadb) (1.17.0)
Requirement already satisfied: python-dateutil>=2.5.3 in
/usr/local/lib/python3.12/dist-packages (from kubernetes>=28.1.0-
>chromadb) (2.9.0.post0)
Requirement already satisfied: google-auth>=1.0.1 in
/usr/local/lib/python3.12/dist-packages (from kubernetes>=28.1.0-
>chromadb) (2.38.0)
Requirement already satisfied: websocket-client!=0.40.0,!=0.41.*,!
=0.42.*,>=0.32.0 in /usr/local/lib/python3.12/dist-packages (from
```

```
kubernetes>=28.1.0->chromadb) (1.9.0)
Requirement already satisfied: requests-oauthlib in
/usr/local/lib/python3.12/dist-packages (from kubernetes>=28.1.0-
>chromadb) (2.0.0)
Requirement already satisfied: urllib3<2.4.0,>=1.24.2 in
/usr/local/lib/python3.12/dist-packages (from kubernetes>=28.1.0-
>chromadb) (2.3.0)
Requirement already satisfied: durationpy>=0.7 in
/usr/local/lib/python3.12/dist-packages (from kubernetes>=28.1.0-
>chromadb) (0.10)
Requirement already satisfied: jsonpatch<2.0.0,>=1.33.0 in
/usr/local/lib/python3.12/dist-packages (from langchain-
core<1.0.0,>=0.3.72->langchain) (1.33)
Requirement already satisfied: requests-toolbelt>=1.0.0 in
/usr/local/lib/python3.12/dist-packages (from langsmith>=0.1.17-
>langchain) (1.0.0)
Requirement already satisfied: zstandard>=0.23.0 in
/usr/local/lib/python3.12/dist-packages (from langsmith>=0.1.17-
>langchain) (0.25.0)
Requirement already satisfied: coloredlogs in
/usr/local/lib/python3.12/dist-packages (from onnxruntime>=1.14.1-
>chromadb) (15.0.1)
Requirement already satisfied: flatbuffers in
/usr/local/lib/python3.12/dist-packages (from onnxruntime>=1.14.1-
>chromadb) (25.9.23)
Requirement already satisfied: protobuf in
/usr/local/lib/python3.12/dist-packages (from onnxruntime>=1.14.1-
>chromadb) (5.29.5)
Requirement already satisfied: sympy in
/usr/local/lib/python3.12/dist-packages (from onnxruntime>=1.14.1-
>chromadb) (1.13.3)
Requirement already satisfied: importlib-metadata<8.8.0,>=6.0 in
/usr/local/lib/python3.12/dist-packages (from opentelemetry-
api>=1.2.0->chromadb) (8.7.0)
Requirement already satisfied: googleapis-common-protos~=1.57 in
/usr/local/lib/python3.12/dist-packages (from opentelemetry-exporter-
otlp-proto-grpc>=1.2.0->chromadb) (1.71.0)
Requirement already satisfied: opentelemetry-exporter-otlp-proto-
common==1.38.0 in /usr/local/lib/python3.12/dist-packages (from
opentelemetry-exporter-otlp-proto-grpc>=1.2.0->chromadb) (1.38.0)
Requirement already satisfied: opentelemetry-proto==1.38.0 in
/usr/local/lib/python3.12/dist-packages (from opentelemetry-exporter-
otlp-proto-grpc>=1.2.0->chromadb) (1.38.0)
Requirement already satisfied: opentelemetry-semantic-
conventions==0.59b0 in /usr/local/lib/python3.12/dist-packages (from
opentelemetry-sdk>=1.2.0->chromadb) (0.59b0)
Requirement already satisfied: backoff>=1.10.0 in
/usr/local/lib/python3.12/dist-packages (from posthog<6.0.0,>=2.4.0-
>chromadb) (2.2.1)
```

```
Requirement already satisfied: distro>=1.5.0 in
/usr/local/lib/python3.12/dist-packages (from posthog<6.0.0,>=2.4.0-
>chromadb) (1.9.0)
Requirement already satisfied: annotated-types>=0.6.0 in
/usr/local/lib/python3.12/dist-packages (from pydantic<3.0.0,>=2.7.4-
>langchain) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in
/usr/local/lib/python3.12/dist-packages (from pydantic<3.0.0,>=2.7.4-
>langchain) (2.33.2)
Requirement already satisfied: typing-inspection>=0.4.0 in
/usr/local/lib/python3.12/dist-packages (from pydantic<3.0.0,>=2.7.4-
>langchain) (0.4.2)
Requirement already satisfied: python-dotenv>=0.21.0 in
/usr/local/lib/python3.12/dist-packages (from pydantic-
settings<3.0.0,>=2.10.1->langchain-community) (1.2.1)
Requirement already satisfied: charset_normalizer<4,>=2 in
/usr/local/lib/python3.12/dist-packages (from requests>=2.26.0-
>tiktoken) (3.4.4)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.12/dist-packages (from rich>=10.11.0->chromadb)
(4.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.12/dist-packages (from rich>=10.11.0->chromadb)
(2.19.2)
Requirement already satisfied: greenlet>=1 in
/usr/local/lib/python3.12/dist-packages (from SQLAlchemy<3,>=1.4-
>langchain) (3.2.4)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (75.2.0)
Requirement already satisfied: networkx in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (3.5)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (3.1.6)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (12.6.77)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (12.6.80)
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (9.10.2.21)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in
```

```
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (12.6.4.1)
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (11.3.0.4)
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (10.3.7.77)
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (11.7.1.2)
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (12.5.4.2)
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (0.7.1)
Requirement already satisfied: nvidia-nccl-cu12==2.27.3 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (2.27.3)
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (12.6.77)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (12.6.85)
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (1.11.1.6)
Requirement already satisfied: triton==3.4.0 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-
transformers) (3.4.0)
Requirement already satisfied: safetensors>=0.4.3 in
/usr/local/lib/python3.12/dist-packages (from
transformers<5.0.0,>=4.41.0->sentence-transformers) (0.6.2)
Requirement already satisfied: click>=8.0.0 in
/usr/local/lib/python3.12/dist-packages (from typer>=0.9.0->chromadb)
(8.3.0)
Requirement already satisfied: shellingham>=1.3.0 in
/usr/local/lib/python3.12/dist-packages (from typer>=0.9.0->chromadb)
(1.5.4)
Requirement already satisfied: httptools>=0.6.3 in
/usr/local/lib/python3.12/dist-packages (from
uvicorn[standard]>=0.18.3->chromadb) (0.7.1)
Requirement already satisfied: uvloop>=0.15.1 in
/usr/local/lib/python3.12/dist-packages (from
uvicorn[standard]>=0.18.3->chromadb) (0.22.1)
Requirement already satisfied: watchfiles>=0.13 in
/usr/local/lib/python3.12/dist-packages (from
```

```
uvicorn[standard]>=0.18.3->chromadb) (1.1.1)
Requirement already satisfied: websockets>=10.4 in
/usr/local/lib/python3.12/dist-packages (from
uvicorn[standard]>=0.18.3->chromadb) (15.0.1)
Requirement already satisfied: joblib>=1.2.0 in
/usr/local/lib/python3.12/dist-packages (from scikit-learn->sentence-
transformers) (1.5.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.12/dist-packages (from scikit-learn->sentence-
transformers) (3.6.0)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/usr/local/lib/python3.12/dist-packages (from google-auth>=1.0.1-
>kubernetes>=28.1.0->chromadb) (5.5.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.12/dist-packages (from google-auth>=1.0.1-
>kubernetes>=28.1.0->chromadb) (0.4.2)
Requirement already satisfied: rsa<5,>=3.1.4 in
/usr/local/lib/python3.12/dist-packages (from google-auth>=1.0.1-
>kubernetes>=28.1.0->chromadb) (4.9.1)
Requirement already satisfied: zipp>=3.20 in
/usr/local/lib/python3.12/dist-packages (from importlib-
metadata<8.8.0,>=6.0->opentelemetry-api>=1.2.0->chromadb) (3.23.0)
Requirement already satisfied: jsonpointer>=1.9 in
/usr/local/lib/python3.12/dist-packages (from
jsonpatch<2.0.0,>=1.33.0->langchain-core<1.0.0,>=0.3.72->langchain)
(3.0.0)
Requirement already satisfied: mdurl~=0.1 in
/usr/local/lib/python3.12/dist-packages (from markdown-it-py>=2.2.0-
>rich>=10.11.0->chromadb) (0.1.2)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.12/dist-packages (from sympy-
>onnxruntime>=1.14.1->chromadb) (1.3.0)
Requirement already satisfied: mypy-extensions>=0.3.0 in
/usr/local/lib/python3.12/dist-packages (from typing-
inspect<1,>=0.4.0->dataclasses-json<0.7.0,>=0.6.7->langchain-
community) (1.1.0)
Requirement already satisfied: sniffio>=1.1 in
/usr/local/lib/python3.12/dist-packages (from anyio->httpx>=0.27.0-
>chromadb) (1.3.1)
Requirement already satisfied: humanfriendly>=9.1 in
/usr/local/lib/python3.12/dist-packages (from coloredlogs-
>onnxruntime>=1.14.1->chromadb) (10.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.12/dist-packages (from jinja2->torch>=1.11.0-
>sentence-transformers) (3.0.3)
Requirement already satisfied: oauthlib>=3.0.0 in
/usr/local/lib/python3.12/dist-packages (from requests-oauthlib-
>kubernetes>=28.1.0->chromadb) (3.3.1)
Requirement already satisfied: pyasn1<0.7.0,>=0.6.1 in
```

```
/usr/local/lib/python3.12/dist-packages (from pyasn1-modules>=0.2.1->google-auth>=1.0.1->kubernetes>=28.1.0->chromadb) (0.6.1)
```

```
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

apple_pdf_path = "/content/drive/MyDrive/HBR_How_Apple_Is_Organized_For_Innovation-4.pdf"

pdf_loader = PyPDFLoader(apple_pdf_path)

apple = pdf_loader.load()

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

```
Page Number : 1
REPRINT R2006F
PUBLISHED IN HBR
NOVEMBER–DECEMBER 2020
ARTICLEORGANIZATIONAL CULTURE
How Apple Is
Organized
for Innovation
It’s about experts leading experts.
by Joel M. Podolny and Morten T. Hansen
This article is made available to you with compliments of Apple Inc
for your personal use. Further posting, copying or distribution is not
permitted.
Page Number : 2
2
Harvard Business Review
November–December 2020
This article is made available to you with compliments of Apple Inc
for your personal use. Further posting, copying or distribution is not
permitted.
Page Number : 3
PHOTOGRAPHER MIKAEL JANSSON
How Apple Is  Organized  for InnovationIt’s about experts  leading
experts.
ORGANIZATIONAL
```

```
CULTURE
Joel M.
Podolny
Dean, Apple
University
Morten T.
Hansen
Faculty, Apple
University
AUTHORS
FOR ARTICLE REPRINTS CALL 800-988-0886 OR 617-783-7500, OR VISIT
HBR.ORG
Harvard Business Review
November–December 2020  3
This article is made available to you with compliments of Apple Inc
for your personal use. Further posting, copying or distribution is not
permitted.
```

```
apple[5].page_content
```

```
{"type":"string"}
```

```
len(apple)
```

```
11
```

```
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)
```

```
25
```

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

```
{"type":"string"}
```

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

```
{"type":"string"}
```

```
document_chunks[-1].page_content
```

```
{"type":"string"}
```

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

```
/tmp/ipython-input-4198310515.py:1: LangChainDeprecationWarning: The
class `HuggingFaceEmbeddings` was deprecated in LangChain 0.2.2 and
will be removed in 1.0. An updated version of the class exists in
the :class:`~langchain-huggingface package and should be used instead.
To use it run `pip install -U :class:`~langchain-huggingface` and
import as `from :class:`~langchain_huggingface import
HuggingFaceEmbeddings``.
  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/tokens), set it as
secret in your Google Colab and restart your session.
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(
```


```
{"model_id":"f74ea13fbbd74c73bc61f797c557f4fb","version_major":2,"vers
ion_minor":0}

{"model_id":"29ab1c6e9b6d4c7f9c3a81fa46848c36","version_major":2,"vers
ion_minor":0}

{"model_id":"99a9894aa51b4c519acdd21f50f0e404","version_major":2,"vers
ion_minor":0}

{"model_id":"c1abf5d9c9f24fa1989f3fda283d30ce","version_major":2,"vers
ion_minor":0}

{"model_id":"20d8fbccb44b4415b746b95f0ee63c26","version_major":2,"vers
ion_minor":0}

{"model_id":"f70b6749da674716be55a23f98312683","version_major":2,"vers
ion_minor":0}

{"model_id":"965a9a3981de4270b353efe3ee5f9228","version_major":2,"vers
ion_minor":0}

{"model_id":"5d9e0a45837a49128d552eacd660db81","version_major":2,"vers
ion_minor":0}

{"model_id":"6e98a6463cb448b2b06924a6fe06f7d6","version_major":2,"vers
ion_minor":0}

{"model_id":"823668341e844a3fad81056ac34a2e12","version_major":2,"vers
ion_minor":0}
```


```
!pip install numpy==1.26.4 sentence-transformers==2.8.1 --upgrade --
force-reinstall --no-cache-dir -q
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/61.0 kB ? eta -:--:--
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 61.0/61.0 kB 8.0 MB/s eta
0:00:00
ERROR: Ignored the following yanked versions: 0.2.6
ERROR: Ignored the following versions that require a different python
version: 1.21.2 Requires-Python >=3.7,<3.11; 1.21.3 Requires-Python
>=3.7,<3.11; 1.21.4 Requires-Python >=3.7,<3.11; 1.21.5 Requires-
Python >=3.7,<3.11; 1.21.6 Requires-Python >=3.7,<3.11
ERROR: Could not find a version that satisfies the requirement
sentence-transformers==2.8.1 (from versions: 0.1.0, 0.2.0, 0.2.1,
0.2.2, 0.2.3, 0.2.4, 0.2.4.1, 0.2.5, 0.2.5.1, 0.2.6.1, 0.2.6.2, 0.3.0,
0.3.1, 0.3.2, 0.3.3, 0.3.4, 0.3.5, 0.3.5.1, 0.3.6, 0.3.7, 0.3.7.1,
0.3.7.2, 0.3.8, 0.3.9, 0.4.0, 0.4.1, 0.4.1.1, 0.4.1.2, 1.0.0, 1.0.1,
1.0.2, 1.0.3, 1.0.4, 1.1.0, 1.1.1, 1.2.0, 1.2.1, 2.0.0, 2.1.0, 2.2.0,
2.2.1, 2.2.2, 2.3.0, 2.3.1, 2.4.0, 2.5.0, 2.5.1, 2.6.0, 2.6.1, 2.7.0,
3.0.0, 3.0.1, 3.1.0, 3.1.1, 3.2.0, 3.2.1, 3.3.0, 3.3.1, 3.4.0, 3.4.1,
4.0.0, 4.0.1, 4.0.2, 4.1.0, 5.0.0, 5.1.0, 5.1.1, 5.1.2)
ERROR: No matching distribution found for sentence-transformers==2.8.1
```

```
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 0.2.9 and will be removed
in 1.0. An updated version of the class exists in
the :class:`~langchain-chroma package and should be used instead. To
use it run `pip install -U :class:`~langchain-chroma` and import as
```

```
`from :class:`~langchain_chroma import Chroma``.
  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)

[Document(metadata={'creationdate': '2020-10-05T14:18:42-04:00',
'source':
'/content/drive/MyDrive/HBR_How_Apple_Is_Organized_For_Innovation-
4.pdf', 'total_pages': 11, 'page_label': '5', 'creator': 'Adobe
InDesign 14.0 (Macintosh)', 'trapped': '/False', 'page': 4,
'producer': 'Adobe PDF Library 15.0 (via http://bfo.com/products/pdf?
version=2.23.5-r33279)', 'moddate': '2020-12-01T18:37:49+00:00'},
page_content='WHY A FUNCTIONAL ORGANIZATION?\nApple’s main purpose is
to create products that enrich \npeople’s daily lives. That involves
not only developing \nentirely new product categories such as the
iPhone and the \nApple Watch, but also continually innovating within
those \ncategories. Perhaps no product feature better reflects Apple’s
\ncommitment to continuous innovation than the iPhone cam-\nera. When
the iPhone was introduced, in 2007, Steve Jobs \ndevoted only six
seconds to its camera in the annual keynote \nevent for unveiling new
products. Since then iPhone camera \ntechnology has contributed to the
photography industry \nwith a stream of innovations: High dynamic
range imaging \n(2010), panorama photos (2012), True Tone flash
(2013), opti-\ncal image stabilization (2015), the dual-lens camera
(2016), \nportrait mode (2016), portrait lighting (2017), and night
mode \n(2019) are but a few of the improvements.\nTo create such
innovations, Apple relies on a structure \nthat centers on functional
expertise. Its fundamental belief \nis that those with the most
expertise and experience in a \ndomain should have decision rights for
that domain. This \nis based on two views: First, Apple competes in
markets \nwhere the rates of technological change and disruption are \
nhigh, so it must rely on the judgment and intuition of people \nwith
deep knowledge of the technologies responsible for \ndisruption. Long
before it can get market feedback and solid \nmarket forecasts, the
```

company must make bets about which \ntechnologies and designs are likely to succeed in smart-\nphones, computers, and so on. Relying on technical experts \nrather than general managers increases the odds that those \nbets will pay off.\nSecond, Apple’s commitment to offer the best possible \nproducts would be undercut if short-term profit and cost \nABOUT THE ART\nApple Park, Apple’s corporate headquarters in \nCupertino, California, opened in 2017.\nMikael Jansson/Trunk Archive\nFOR ARTICLE REPRINTS CALL 800-988-0886 OR 617-783-7500, OR VISIT HBR.ORG\nHarvard Business Review\nNovember–December 2020 \u20095'),
 Document(metadata={'source': '/content/drive/MyDrive/HBR_How_Apple_Is_Organized_For_Innovation-4.pdf', 'page_label': '4', 'creationdate': '2020-10-05T14:18:42-04:00', 'page': 3, 'moddate': '2020-12-01T18:37:49+00:00', 'trapped': '/False', 'producer': 'Adobe PDF Library 15.0 (via http://bfo.com/products/pdf?version=2.23.5-r33279)', 'total_pages': 11, 'creator': 'Adobe InDesign 14.0 (Macintosh)'}, page_content='WELL KNOWN FOR ITS innovations in hardware, software, \nand services. Thanks to them, it grew from some 8,000 \nemployees and $7\n \nbillion in revenue in 1997, the year Steve \nJobs returned, to 137,000 employees and $260\n \nbillion in \nrevenue in 2019. Much less well known are the organizational \ndesign and the associated leadership model that have played \na crucial role in the company’s innovation success.\nWhen Jobs arrived back at Apple, it had a conventional \nstructure for a company of its size and scope. It was divided \ninto business units, each with its own P&L responsibilities. \nGeneral managers ran the Macintosh products group, the \ninformation appliances division, and the server products \ndivision, among others. As is often the case with decentral-\nized business units, managers were inclined to fight with \none another, over transfer prices in particular. Believing that \nconventional management had stifled innovation, Jobs, in \nhis first year returning as CEO, laid off the general managers \nof all the business units (in a single day), put the entire com-\npany under one P&L, and combined the disparate functional \ndepartments of the business units into one functional organi-\nzation. (See the exhibit “ Apple’s Functional Organization. ”)\nThe adoption of a functional structure may have been \nun\nsurprisin\ng for a company of Apple’s size at the time. What is \nsurprising—in fact, remarkable—is that Apple retains it today, \neven though the company is nearly 40 times as large in terms \nof revenue and far more complex than it was in 1998. Senior \nvice presidents are in charge of functions, not products. As \nwas the case with Jobs before him, CEO Tim Cook occupies the \nonly position on the organizational chart where the design, \nengineering, operations, marketing, and retail of any of Apple’s \nmain products meet. In effect, besides the CEO, the company \noperates with no conventional general managers: people \nwho control an entire process from product development \nthrough sales and are judged according to a P&L statement.\nBusiness history and organizational theory make the case \

```
nthat as entrepreneurial firms grow large and complex, they'),
 Document(metadata={'moddate': '2020-12-01T18:37:49+00:00',
'total_pages': 11, 'page_label': '9', 'source':
'/content/drive/MyDrive/HBR_How_Apple_Is_Organized_For_Innovation-
4.pdf', 'page': 8, 'creator': 'Adobe InDesign 14.0 (Macintosh)',
'trapped': '/False', 'creationdate': '2020-10-05T14:18:42-04:00',
'producer': 'Adobe PDF Library 15.0 (via http://bfo.com/products/pdf?
version=2.23.5-r33279)'}, page_content='things, that these photos
often had blurring at the edges of a \nface but sharpness on the eyes.
So they charged the algorithm \nteams with achieving the same effect.
When the teams suc-\nceeded, they knew they had an acceptable
standard.\nAnother issue that emerged was the ability to preview a \
nportrait photo with a blurred background. The camera team \nhad
designed the feature so that users could see its effect on \ntheir
photos only after they had been taken, but the human \ninterface (HI)
design team pushed back, insisting that users \nshould be able to see
a “live preview” and get some guidance \nabout how to make adjustments
before taking the photo. \nJohnnie Manzari, a member of the HI team,
gave the camera \nteam a demo. “When we saw the demo, we realized that
this \nis what we needed to do, ” Townsend told us. The members \nof
his camera hardware team weren’t sure they could do \nit, but
difficulty was not an acceptable excuse for failing to \ndeliver what
would clearly be a superior user experience. After \nmonths of
engineering effort, a key stakeholder, the video \nengineering team
(responsible for the low-level software that \ncontrols sensor and
camera operations) found a way, and the \ncollaboration paid off.
Portrait mode was central to Apple’s \nmarketing of the iPhone 7 Plus.
It proved a major reason for \nusers’ choosing to buy and delighting
in the use of the phone.\nAs this example shows, Apple’s collaborative
debate \ninvolves people from various functions who disagree, push \
nback, promote or reject ideas, and build on one another’s \nideas to
come up with the best solutions. It requires open-\n \nmindedness fr\
nom senior leaders. It also requires those \nleaders to inspire, prod,
or influence colleagues in other  \nareas to contribute toward
achieving their goals.\nWhile Townsend is accountable for how great
the camera \nis, he needed dozens of other teams—each of which had a \
nlong list of its own commitments—to contribute their time and \
neffort to the portrait mode proj\n ect. A\nt Apple that’s known as \
naccountability without control: You’re accountable for making \nthe
proj\n ect succeed ev\nen though you don’t control all the other')]

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 coordination between the
teams?")
rel_docs
```

```
/tmp/ipython-input-3586710401.py:1: LangChainDeprecationWarning: The
method `BaseRetriever.get_relevant_documents` was deprecated in
langchain-core 0.1.46 and will be removed in 1.0. Use :meth:`~invoke`
instead.
  rel_docs = retriever.get_relevant_documents("How does does Apple
develop and ship products that requires good coordination between the
teams?")

[Document(metadata={'trapped': '/False', 'page': 7, 'moddate': '2020-
12-01T18:37:49+00:00', 'creationdate': '2020-10-05T14:18:42-04:00',
'page_label': '8', 'producer': 'Adobe PDF Library 15.0 (via
http://bfo.com/products/pdf?version=2.23.5-r33279)', 'source':
'/content/drive/MyDrive/HBR_How_Apple_Is_Organized_For_Innovation-
4.pdf', 'total_pages': 11, 'creator': 'Adobe InDesign 14.0
(Macintosh)'}, page_content='40 specialist teams: silicon design,
camera software, reliabil-\nity engineering, motion sensor hardware,
video engineering, \ncore motion, and camera sensor design, to name
just a few. \nHow on earth does Apple develop and ship products that \
nrequire such coordination? The answer is collaborative \ndebate.
Because no function is responsible for a product or a \nservice on its
own, cross-functional collaboration is crucial.\nWhen debates reach an
impasse, as some inevitably do, \nhigher-level managers weigh in as
tiebreakers, including at \ntimes the CEO and the senior VPs. To do
this at speed with \nsufficient attention to detail is challenging for
even the best \nof leaders, making it all the more important that the
company \nfill many senior positions from within the ranks of its VPs,
\nwho have experience in Apple’s way of operating.\nHowever, given
Apple’s size and scope, even the executive \nteam can resolve only a
limited number of stalemates. The \nmany horizontal dependencies mean
that ineffective peer \nrelationships at the VP and director levels
have the potential \nto undermine not only particular proj\n ec\nts
but the entire \ncompany. Consequently, for people to attain and
remain in \na leadership position within a function, they must be
highly \neffective collaborators.\nThat doesn’t mean people can’t
express their points of \nview. Leaders are expected to hold strong,
well-grounded \nviews and advocate forcefully for them, yet also be
willing  \nto change their minds when presented with evidence \nthat
others’ views are better. Doing so is not always \neasy, of course. A
leader’s ability to be both partisan and \nopen-minded is facilitated
by two things: deep understand-\ning of and devotion to the company’s
values and common \npurpose, and a commitment to separating how right
from \nhow hard a particular path is so that the difficulty of
execut-\ning a decision doesn’t prevent its being selected.\nThe
development of the iPhone’s portrait mode illustrates \na fanatical
attention to detail at the leadership level, intense \ncollaborative
debate among teams, and the power of a shared \npurpose to shape and
ultimately resolve debates. In 2009 \nHubel had the idea of developing
an iPhone feature that \nwould allow people to take portrait photos
with bokeh—'),
 Document(metadata={'producer': 'Adobe PDF Library 15.0 (via
```

```
http://bfo.com/products/pdf?version=2.23.5-r33279)', 'creator': 'Adobe InDesign 14.0 (Macintosh)', 'source': '/content/drive/MyDrive/HBR_How_Apple_Is_Organized_For_Innovation-4.pdf', 'creationdate': '2020-10-05T14:18:42-04:00', 'page': 6, 'page_label': '7', 'trapped': '/False', 'total_pages': 11, 'moddate': '2020-12-01T18:37:49+00:00'}, page_content='Apple is run. Leaders can push, probe, and “smell” an issue. \nThey know which details are important and where to focus \ntheir attention. Many people at Apple see it as liberating, \neven exhilarating, to work for experts, who provide better \nguidance and mentoring than a general manager would. \nTogether, all can strive to do the best work of their lives in \ntheir chosen area.\nWillingness to collaboratively debate. Apple has \nhundreds of specialist teams across the company, dozens of \nwhich may be needed for even one key component of a new \nproduct offering. For example, the dual-lens camera with \nportrait mode required the collaboration of no fewer than  \nApple leaders are expected to possess deep expertise, be immersed \nin the details of their functions, and engage in collaborative debate.\nORGANIZATIONAL \nCULTURE\nFOR ARTICLE REPRINTS CALL 800-988-0886 OR 617-783-7500, OR VISIT HBR.ORG\nHarvard Business Review\nNovember–December 2020 \u20097\nThis article is made available to you with compliments of Apple Inc for your personal use. Further posting, copying or distribution is not permitted.')]

from huggingface_hub import hf_hub_download

!pip install llama-cpp-python==0.2.28 --force-reinstall --upgrade --no-cache-dir -q


Usage:
  pip3 install [options] <requirement specifier> [package-index-options] ...
  pip3 install [options] -r <requirements file> [package-index-options] ...
  pip3 install [options] [-e] <vcs project url> ...
  pip3 install [options] [-e] <local project path> ...
  pip3 install [options] <archive url/path> ...

no such option: --no-cache-dir -q

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
)

{"model_id":"5d33bbab75ae49ec8858c9a34f8f0b42","version_major":2,"version_minor":0}
```

```
from llama_cpp import Llama

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

```
AVX = 1 | AVX_VNNI = 0 | AVX2 = 1 | AVX512 = 1 | AVX512_VBMI = 0 |
AVX512_VNNI = 0 | FMA = 1 | NEON = 0 | ARM_FMA = 0 | F16C = 1 |
FP16_VA = 0 | WASM_SIMD = 0 | BLAS = 1 | SSE3 = 1 | SSSE3 = 1 | VSX =
0 |
```

```
llm("How does Apple develop and ship products that requires good
coordination between the teams?")['choices'][0]['text']
```

```
{"type":"string"}
```

```
qna_system_message = """
You are an assistant whose work is to review the report and provide
the appropriate answers from the context.
User input will have the context required by you to answer user
questions.
This context will begin with the token: ###Context.
The context contains references to specific portions of a document
relevant to the user query.

User questions will begin with the token: ###Question.

Please answer only using the context provided in the input. Do not
mention anything about the context in your final answer.

If the answer is not found in the context, respond "I don't know".
"""

qna_user_message_template = """
###Context
Here are some documents that are relevant to the question mentioned
below.
{context}

###Question
{question}
"""

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

llm("Who are the authors of this article and who published this
article ?")['choices'][0]['text']
```

```
Llama.generate: prefix-match hit
```

```
{"type":"string"}
```

```
user_input = "Who are the authors of this article and who published
this article ?"
print(generate_rag_response(user_input))
```

```
Llama.generate: prefix-match hit
```

```
Answer:
Morten T. Hansen and Joel M. Podolny are the authors of the article.
Harvard Business Review published it.
```

```
llm("List down the three leadership characteristics in bulleted points
and explain each one of the characteristics under two lines.")
['choices'][0]['text']
```

```
Llama.generate: prefix-match hit
```

```
{"type":"string"}
```

```
user_input_2 = "List down the three leadership characteristics in
bulleted points and explain each one of the characteristics under two
lines."
generate_rag_response(user_input_2)
```

```
Llama.generate: prefix-match hit
```

```
{"type":"string"}
```

```
user_input_3 = "Can you explain specific examples from the article
where Apple's approach to leadership has led to successful
innovations?"
generate_rag_response(user_input_3)
```

```
Llama.generate: prefix-match hit
```

```
{"type":"string"}
```

```
user_input = "Who are the authors of this article and who published
this article ?"
generate_rag_response(user_input, max_tokens=100)
```

```
Llama.generate: prefix-match hit
```

```
{"type":"string"}
```

```
user_input_2 = "List down the three leadership characteristics in
bulleted points and explain each one of the characteristics under two
lines."
generate_rag_response(user_input_2, temperature=0.1, max_tokens=350)
```

```
Llama.generate: prefix-match hit
```

```
{"type":"string"}
```

```
user_input_3 = "Can you explain specific examples from the article
where Apple's approach to leadership has led to successful
innovations?"
generate_rag_response(user_input_3, top_p=0.98, top_k=20,
max_tokens=256)
```

```
Llama.generate: prefix-match hit
```

```
{"type":"string"}
```

```
groundedness_rater_system_message = """
You are tasked with rating AI generated answers to questions posed by
users.
You will be presented a question, context used by the AI system to
generate the answer and an AI generated answer to the question.
In the input, the question will begin with ###Question, the context
will begin with ###Context while the AI generated answer will begin
with ###Answer.
```

```
Evaluation criteria:
The task is to judge the extent to which the metric is followed by the
answer.
1 - The metric is not followed at all
2 - The metric is followed only to a limited extent
3 - The metric is followed to a good extent
4 - The metric is followed mostly
5 - The metric is followed completely

Metric:
The answer should be derived only from the information presented in
the context

Instructions:
1. First write down the steps that are needed to evaluate the answer
as per the metric.
2. Give a step-by-step explanation if the answer adheres to the metric
considering the question and context as the input.
3. Next, evaluate the extent to which the metric is followed.
4. Use the previous information to rate the answer using the evaluaton
criteria and assign a score.
"""

relevance_rater_system_message = """
You are tasked with rating AI generated answers to questions posed by
users.
You will be presented a question, context used by the AI system to
generate the answer and an AI generated answer to the question.
In the input, the question will begin with ###Question, the context
will begin with ###Context while the AI generated answer will begin
with ###Answer.

Evaluation criteria:
The task is to judge the extent to which the metric is followed by the
answer.
1 - The metric is not followed at all
2 - The metric is followed only to a limited extent
3 - The metric is followed to a good extent
4 - The metric is followed mostly
5 - The metric is followed completely

Metric:
Relevance measures how well the answer addresses the main aspects of
the question, based on the context.
Consider whether all and only the important aspects are contained in
the answer when evaluating relevance.

Instructions:
1. First write down the steps that are needed to evaluate the context
```

```
as per the metric.
2. Give a step-by-step explanation if the context adheres to the
metric considering the question as the input.
3. Next, evaluate the extent to which the metric is followed.
4. Use the previous information to rate the context using the
evaluaton criteria and assign a score.
"""

user_message_template = """
###Question
{question}

###Context
{context}

###Answer
{answer}
"""

def
generate_ground_relevance_response(user_input,k=3,max_tokens=128,tempe
rature=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=answer)}
```

```
                [/INST]"""

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

user_input = "Who are the authors of this article and who published
this article ?"
ground,rel =
generate_ground_relevance_response(user_input,max_tokens=350)

print(ground,end="\n\n")
print(rel)
```

```
Llama.generate: prefix-match hit
Llama.generate: prefix-match hit
Llama.generate: prefix-match hit

 Steps to evaluate the answer:
1. Identify the key information in the context related to the
question.
2. Check if the answer is derived only from the identified information
in the context.
3. Evaluate the extent to which the metric is followed.

Explanation:
The question asks for the authors of the article and the publisher.
The context provides the names of the authors (Morten T. Hansen and
```

```
Joel M. Podolny) and the name of the publisher (Harvard Business
Review). The answer correctly identifies both the authors and the
publisher from the information given in the context. Therefore, the
answer is derived only from the information presented in the context.

Evaluation:
The metric is followed completely as the answer is derived solely from
the context without any additional information or assumptions.

Rating:
Based on the evaluation criteria, I would rate the answer a 5 for
following the metric completely.

 Steps to evaluate the context as per the metric:
1. Identify the main aspects of the question: In this case, the main
aspects of the question are identifying the authors and the publisher
of the article.
2. Determine if the context contains all and only the important
aspects: The context provides the names of the authors (Morten T.
Hansen and Joel M. Podolny) and the name of the publisher (Harvard
Business Review). Therefore, it adheres to the metric as it contains
all the necessary information to answer the question.

The extent to which the metric is followed:
The context follows the metric completely as it provides all the
important aspects of the question in the answer.

Rating the context using the evaluation criteria and assigning a
score:
Since the context follows the metric completely, I would rate it a 5
on the evaluation criteria scale.
```

```
user_input_2 = "List down the three leadership characteristics in
bulleted points and explain each one of the characteristics under two
lines."
ground,rel =
generate_ground_relevance_response(user_input_2,max_tokens=500)

print(ground,end="\n\n")
print(rel)
```

```
Llama.generate: prefix-match hit
Llama.generate: prefix-match hit
Llama.generate: prefix-match hit
```

```
 Steps to evaluate the answer:
1. Identify the leadership characteristics mentioned in the question
and context.
2. Determine if each line of the AI generated answer is derived
directly from the information presented in the context.
3. Check if the explanation for each characteristic adheres to the
```

```
metric by ensuring that it only uses information from the context.

The first characteristic, "Deep expertise," is explained as Apple's
managers being expected to possess deep expertise in their individual
functions and experts leading other experts. This directly aligns with
the context which states, "Apple’s managers at every level, from
senior vice president on down, have been expected to possess three key
leadership characteristics:

 Steps to evaluate context as per relevance metric:
1. Identify the main aspects of the question: In this case, the
question asks for three leadership characteristics at Apple and an
explanation of each one under two lines.
2. Determine if the context provides information on the main aspects:
The context discusses Apple's functional organization and the
leadership model underlying it, specifically focusing on the three
leadership characteristics: deep expertise and immersion in the
details.
3. Check if the context explains each characteristic: The context not
only lists the characteristics but also provides an
```

```
user_input_3 = "Can you explain specific examples from the article
where Apple's approach to leadership has led to successful
innovations?"
ground,rel =
generate_ground_relevance_response(user_input_3,max_tokens=500)

print(ground,end="\n\n")
print(rel)
```

```
Llama.generate: prefix-match hit
Llama.generate: prefix-match hit
Llama.generate: prefix-match hit

 Steps to evaluate the answer:
1. Identify the specific examples mentioned in the article regarding
Apple's approach to leadership leading to successful innovations.

 Steps to evaluate context
```