```
pip install pdfplumber pymupdf pytesseract pdf2image pillow
```

```
Collecting pdfplumber
  Downloading pdfplumber-0.11.5-py3-none-any.whl.metadata (42 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.5/42.5 kB 1.4 MB/s eta 0:00:00
Collecting pymupdf
  Downloading pymupdf-1.25.4-cp39-abi3-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (3.4 kB)
Collecting pytesseract
  Downloading pytesseract-0.3.13-py3-none-any.whl.metadata (11 kB)
Collecting pdf2image
  Downloading pdf2image-1.17.0-py3-none-any.whl.metadata (6.2 kB)
Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-packages (11.1.0)
Collecting pdfminer.six==20231228 (from pdfplumber)
  Downloading pdfminer.six-20231228-py3-none-any.whl.metadata (4.2 kB)
Collecting pypdfium2>=4.18.0 (from pdfplumber)
  Downloading pypdfium2-4.30.1-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (48 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 48.2/48.2 kB 3.1 MB/s eta 0:00:00
Requirement already satisfied: charset-normalizer>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from pdfminer.six==20231228->pd
Requirement already satisfied: cryptography>=36.0.0 in /usr/local/lib/python3.11/dist-packages (from pdfminer.six==20231228->pdfplum
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.11/dist-packages (from pytesseract) (24.2)
Requirement already satisfied: cffi>=1.12 in /usr/local/lib/python3.11/dist-packages (from cryptography>=36.0.0->pdfminer.six==20231
Requirement already satisfied: pycparser in /usr/local/lib/python3.11/dist-packages (from cffi>=1.12->cryptography>=36.0.0->pdfminer
Downloading pdfplumber-0.11.5-py3-none-any.whl (59 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 59.5/59.5 kB 4.8 MB/s eta 0:00:00
Downloading pdfminer.six-20231228-py3-none-any.whl (5.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.6/5.6 MB 49.6 MB/s eta 0:00:00
Downloading pymupdf-1.25.4-cp39-abi3-manylinux2014_x86_64.manylinux_2_17_x86_64.whl (20.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20.0/20.0 MB 24.3 MB/s eta 0:00:00
Downloading pytesseract-0.3.13-py3-none-any.whl (14 kB)
Downloading pdf2image-1.17.0-py3-none-any.whl (11 kB)
Downloading pypdfium2-4.30.1-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.9/2.9 MB 81.4 MB/s eta 0:00:00
Installing collected packages: pytesseract, pypdfium2, pymupdf, pdf2image, pdfminer.six, pdfplumber
Successfully installed pdf2image-1.17.0 pdfminer.six-20231228 pdfplumber-0.11.5 pymupdf-1.25.4 pypdfium2-4.30.1 pytesseract-0.3.13
```

```
!apt-get update
!apt-get install -y tesseract-ocr
```

```
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:3 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease [3,632 B]
Get:4 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  InRelease [1,581 B]
Get:5 https://r2u.stat.illinois.edu/ubuntu jammy InRelease [6,555 B]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:7 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Get:8 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Hit:9 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy InRelease
Hit:10 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Get:11 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [47.7 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2,698 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1,238 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [3,813 kB]
Get:15 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  Packages [1,381 kB]
Get:16 https://r2u.stat.illinois.edu/ubuntu jammy/main amd64 Packages [2,680 kB]
Get:17 https://r2u.stat.illinois.edu/ubuntu jammy/main all Packages [8,772 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [4,041 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,538 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [55.7 kB]
Get:21 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [3,041 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [35.2 kB]
Fetched 29.7 MB in 6s (5,075 kB/s)
Reading package lists... Done
W: Skipping acquire of configured file 'main/source/Sources' as repository 'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' d
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  tesseract-ocr-eng tesseract-ocr-osd
The following NEW packages will be installed:
  tesseract-ocr tesseract-ocr-eng tesseract-ocr-osd
0 upgraded, 3 newly installed, 0 to remove and 35 not upgraded.
Need to get 4,816 kB of archives.
After this operation, 15.6 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tesseract-ocr-eng all 1:4.00~git30-7274cfa-1.1 [1,591 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tesseract-ocr-osd all 1:4.00~git30-7274cfa-1.1 [2,990 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tesseract-ocr amd64 4.1.1-2.1build1 [236 kB]
Fetched 4,816 kB in 1s (4,257 kB/s)
Selecting previously unselected package tesseract-ocr-eng.
(Reading database ... 126209 files and directories currently installed.)
Preparing to unpack .../tesseract-ocr-eng_1%3a4.00~git30-7274cfa-1.1_all.deb ...
Unpacking tesseract-ocr-eng (1:4.00~git30-7274cfa-1.1) ...
Selecting previously unselected package tesseract-ocr-osd.
Preparing to unpack .../tesseract-ocr-osd_1%3a4.00~git30-7274cfa-1.1_all.deb ...
Unpacking tesseract-ocr-osd (1:4.00~git30-7274cfa-1.1) ...
Selecting previously unselected package tesseract-ocr.
```

```
Preparing to unpack .../tesseract-ocr_4.1.1-2.1build1_amd64.deb ...
Unpacking tesseract-ocr (4.1.1-2.1build1) ...
Setting up tesseract-ocr-eng (1:4.00~git30-7274cfa-1.1) ...
Setting up tesseract-ocr-osd (1:4.00~git30-7274cfa-1.1) ...
Setting up tesseract-ocr (4.1.1-2.1build1) ...
Processing triggers for man-db (2.10.2-1) ...
```

```
!apt-get install -y poppler-utils
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  poppler-utils
0 upgraded, 1 newly installed, 0 to remove and 35 not upgraded.
Need to get 186 kB of archives.
After this operation, 696 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 poppler-utils amd64 22.02.0-2ubuntu0.6 [186 kB]
Fetched 186 kB in 1s (321 kB/s)
Selecting previously unselected package poppler-utils.
(Reading database ... 126256 files and directories currently installed.)
Preparing to unpack .../poppler-utils_22.02.0-2ubuntu0.6_amd64.deb ...
Unpacking poppler-utils (22.02.0-2ubuntu0.6) ...
Setting up poppler-utils (22.02.0-2ubuntu0.6) ...
Processing triggers for man-db (2.10.2-1) ...
```

```
import os
os.listdir("/content")
```

```
['.config', 'CBT metaanalysis from prison.pdf', 'sample_data']
```

```
import pdfplumber
import pytesseract
from pdf2image import convert_from_path
from PIL import Image
import os
import pandas as pd
import textwrap
import re

# Set your PDF path
pdf_path = "/content/CBT metaanalysis from prison.pdf"
filename = os.path.basename(pdf_path)

# 🧼 Universal cleaner function
def clean_text_universal(text):
    text = re.sub(r'--- Page \d+ ---', ' ', text)
    text = re.sub(r'\S+@\S+', ' ', text)
    text = re.sub(r'http\S+|www\S+|doi\S+', ' ', text)
    text = re.sub(r'(\w+)-\s*\n\s*(\w+)', r'\1\2', text)
    text = re.sub(r'\s+', ' ', text)
    return text.strip()

# ✂️ Chunking function
def split_text_into_chunks(text, chunk_size=500):
    paragraphs = text.split("\n\n")
    chunks = []
    chunk_id = 0
    for para in paragraphs:
        if len(para.strip()) < 100:
            continue
        wrapped_chunks = textwrap.wrap(para.strip(), chunk_size)
        for chunk in wrapped_chunks:
            chunk_id += 1
            chunks.append((f"P1-C{chunk_id}", chunk))
    return chunks

# 🔍 First try pdfplumber
def extract_text_pdfplumber(pdf_path):
    full_text = ""
    try:
        with pdfplumber.open(pdf_path) as pdf:
            for i, page in enumerate(pdf.pages):
                text = page.extract_text()
                if text:
                    full_text += f"\n\n--- Page {i+1} ---\n{text}"
    except:
        pass
    return full_text.strip()

# 📸 OCR fallback for scanned PDFs
```

```
def extract_text_ocr(pdf_path, dpi=300):
    print("[INFO] Using OCR fallback for scanned PDF...")
    pages = convert_from_path(pdf_path, dpi=dpi)
    text = ""
    for i, img in enumerate(pages):
        img = img.convert("L")
        page_text = pytesseract.image_to_string(img)
        text += f"\n\n--- OCR Page {i+1} ---\n{page_text}"
    return text.strip()

# 🧠 Universal text extractor
def extract_text_universal(pdf_path):
    text = extract_text_pdfplumber(pdf_path)
    if len(text) < 100:  # not enough text, fallback to OCR
        text = extract_text_ocr(pdf_path)
    return clean_text_universal(text)

# 🚀 Run the extraction and save
full_text = extract_text_universal(pdf_path)
chunks = split_text_into_chunks(full_text)

df = pd.DataFrame(chunks, columns=["chunk_id", "chunk_text"])
df.insert(0, "filename", filename)
df.to_csv("CBTextracted_chunks.csv", index=False, encoding="utf-8")

print("✅ Done. Text extracted and saved to CBT_metaanalysis_extracted_chunks.csv")
```

```
[INFO] Using OCR fallback for scanned PDF...
✅ Done. Text extracted and saved to CBT_metaanalysis_extracted_chunks.csv
```

```
!pip install sentence-transformers chromadb
```

```
  Successfully built pypika
  Installing collected packages: pypika, monotonic, durationpy, uvloop, uvicorn, python-dotenv, pyproject_hooks, overrides, opentel
    Attempting uninstall: nvidia-nvjitlink-cu12
      Found existing installation: nvidia-nvjitlink-cu12 12.5.82
```

```
import pandas as pd
from sentence_transformers import SentenceTransformer

# Load your cleaned CSV
df = pd.read_csv("/content/CBTextracted_chunks.csv")  # ← your extracted & cleaned file

# Load embedding model (lightweight & accurate)
model = SentenceTransformer("all-MiniLM-L6-v2")
```

```
/usr/local/lib/python3.11/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 s
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%  349/349 [00:00<00:00, 28.1kB/s]
config_sentence_transformers.json: 100%  116/116 [00:00<00:00, 9.01kB/s]
README.md: 100%  10.5k/10.5k [00:00<00:00, 632kB/s]
sentence_bert_config.json: 100%  53.0/53.0 [00:00<00:00, 4.41kB/s]
config.json: 100%  612/612 [00:00<00:00, 47.4kB/s]
model.safetensors: 100%  90.9M/90.9M [00:00<00:00, 129MB/s]
tokenizer_config.json: 100%  350/350 [00:00<00:00, 28.8kB/s]
vocab.txt: 100%  232k/232k [00:00<00:00, 6.87MB/s]
tokenizer.json: 100%  466k/466k [00:00<00:00, 11.6MB/s]
special_tokens_map.json: 100%  112/112 [00:00<00:00, 7.04kB/s]
config.json: 100%  190/190 [00:00<00:00, 11.8kB/s]
```

```
import chromadb
from chromadb.config import Settings

# Initialize ChromaDB in-memory (or use persistent storage later)
chroma_client = chromadb.Client(Settings(anonymized_telemetry=False))
collection = chroma_client.create_collection(name="recidivism_chunks")

# Create documents, metadatas, and ids
documents = df["chunk_text"].tolist()
metadatas = df[["filename", "chunk_id"]].to_dict(orient="records")
ids = [f"{row['filename']}_{row['chunk_id']}" for _, row in df.iterrows()]

# Embed and add to ChromaDB
embeddings = model.encode(documents).tolist()

collection.add(
    documents=documents,
    embeddings=embeddings,
    metadatas=metadatas,
    ids=ids
)
```

```
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).
```

```
import chromadb

persist_path = "/content/drive/MyDrive/Meta Papers/Developer 1/Chormadb Storage"
chroma_client = chromadb.PersistentClient(path=persist_path)
collection = chroma_client.get_or_create_collection(name="recidivism_chunks")
```

```
import pandas as pd

csv_path = "/content/CBTextracted_chunks.csv"  # 👈 Change this if needed
df = pd.read_csv(csv_path)
```

```
# Optional: show a few rows
df.head()
```

	filename	chunk_id	chunk_text
0	CBT metaanalysis from prison.pdf	P1-C1	--- OCR Page 1 --- Articles i. k® Effectivenes...
1	CBT metaanalysis from prison.pdf	P1-C2	however, there is little evidence about their ...
2	CBT metaanalysis from prison.pdf	P1-C3	and Google Scholar for articles published from...
3	CBT metaanalysis from prison.pdf	P1-C4	delivered outside of the prison setting. We ex...
4	CBT metaanalysis from prison.pdf	P1-C5	in a random-effects meta-analysis as pooled od...

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
from sentence_transformers import SentenceTransformer

# Load embedding model
model = SentenceTransformer("all-MiniLM-L6-v2")

# Extract data
documents = df["chunk_text"].tolist()
metadatas = df[["filename", "chunk_id"]].to_dict(orient="records")
ids = [f"{row['filename']}_{row['chunk_id']}" for _, row in df.iterrows()]

# Generate embeddings
embeddings = model.encode(documents).tolist()
```

```
/usr/local/lib/python3.11/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 s
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(
```

```
# Add all to ChromaDB
collection.add(
    documents=documents,
    embeddings=embeddings,
    metadatas=metadatas,
    ids=ids
)

print("✅ All chunks added to ChromaDB and stored in Google Drive!")
print("🧠 Total chunks stored:", collection.count())
```

```
✅ All chunks added to ChromaDB and stored in Google Drive!
🧠 Total chunks stored: 144
```

```
import numpy as np # to save the embedded csv

df['embedding'] = df['chunk_text'].apply(lambda x: model.encode(x).tolist())
df.to_csv("paper1_with_embeddings.csv", index=False)
```

```
from IPython.display import display, Markdown

query = "What does CBT aim to do in prison studies?"
results = collection.query(query_texts=[query], n_results=1)

top_chunk = results["documents"][0][0]
metadata = results["metadatas"][0][0]

display(Markdown(f"""
### 🔍 **Query:**
> `{query}`

---

### 📄 **Top Retrieved Chunk:**

> {top_chunk}

---

### 🧬 **Metadata:**
```

```
- **Filename**: `{metadata['filename']}`
- **Chunk ID**: `{metadata['chunk_id']}`
"""))
```

🔍 Query:

What does CBT aim to do in prison studies?

📄 Top Retrieved Chunk:

is different to evidence from some Vol8 September 2021 reviews (including one published by the Campbell Collaboration"), which have suggested that CBT is one of the most effective forms of treatment for people in prison.'" However, these previous reviews combined RCTs with less than rigorous study designs and the current new findings question the widespread roll-out of these treatment approaches in prisons. Only one® of the six CBT studies"" in our systematic review reported significant

---

📎 Metadata:

- **Filename**: CBT metaanalysis from prison.pdf
- **Chunk ID**: P1-C95

```
from IPython.display import display, Markdown

# 📥 New Query
query = "What is the role of CBT in reducing recidivism?"
results = collection.query(query_texts=[query], n_results=1)

top_chunk = results["documents"][0][0]
metadata = results["metadatas"][0][0]

# 🖼️ Styled Display for Presentation
display(Markdown(f"""
### 🔍 **Query:**
> `{query}`

---

### 📄 **Top Retrieved Chunk:**

> {top_chunk}

---

### 📎 **Metadata:**
- **Filename**: `{metadata['filename']}`
- **Chunk ID**: `{metadata['chunk_id']}`
"""))
```

🔍 Query:

What is the role of CBT in reducing recidivism?

📄 Top Retrieved Chunk:

reductions in reoffending. Other research, in selected populations of all people who have offended and also use drugs, also found little support for CBT." Another implication of our review is that the effects of in-prison psychological interventions on recidivism appear to be smaller than those reported in previous meta-analyses, which have been estimated to be around 0-65 (95% CI 0-57—-0-75).% This difference is probably because the previous reviews included studies using weak research designs,

---

📎 Metadata:

- **Filename**: CBT metaanalysis from prison.pdf
- **Chunk ID**: P1-C96