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

```
Mounted at /content/drive
```

```
!pip -q install pandas numpy scikit-learn tqdm wikipedia feedparser beautifulsoup4 readability-lxml requests
```

```
  Preparing metadata (setup.py) ... done
  Preparing metadata (setup.py) ... done
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 81.5/81.5 kB 3.5 MB/s eta 0:00:00
  Building wheel for wikipedia (setup.py) ... done
  Building wheel for sgmllib3k (setup.py) ... done
```

```
import os, re, json, time, hashlib
import numpy as np
import pandas as pd
from tqdm.notebook import tqdm
from sklearn.model_selection import train_test_split
```

```
tqdm.pandas()

RANDOM_SEED = 42
np.random.seed(RANDOM_SEED)

TEXT_COL = "text"
LABEL_COL = "label"     # 0=human, 1=AI

def clean_text(t: str) -> str:
    if t is None:
        return ""
    t = re.sub(r"http\S+", " ", str(t))
    t = re.sub(r"\S+@\S+", " ", t)
    t = re.sub(r"\s+", " ", t).strip()
    return t

def stable_hash(s: str) -> str:
    return hashlib.sha1(s.encode("utf-8", errors="ignore")).hexdigest()

def chunk_text_words(text: str, chunk_words=200, overlap=40, min_words=40):
    words = re.findall(r"\S+", text)
    if len(words) < min_words:
        return []
    step = max(1, chunk_words - overlap)
    chunks = []
    for i in range(0, len(words), step):
        ch = " ".join(words[i:i+chunk_words]).strip()
        if len(ch.split()) >= min_words:
            chunks.append(ch)
    return chunks

def add_len_bins(df: pd.DataFrame):
    df["len_words"] = df[TEXT_COL].str.split().str.len()
    df["len_bin"] = pd.cut(df["len_words"], bins=[0,10,25,50,100,200,400,1000,10_000],
```

```
                            labels=False, include_lowest=True)
    return df

def dedup_by_text(df: pd.DataFrame):
    df["text_hash"] = df[TEXT_COL].map(stable_hash)
    df = df.drop_duplicates(subset=["text_hash"]).drop(columns=["text_hash"])
    return df
```

```
INPUT_CSV = "/content/drive/MyDrive/AI_Human.csv"
OUTPUT_CSV = "/content/drive/MyDrive/kaggle_text_label.csv"

# def clean_text(x):
#     return re.sub(r"\s+", " ", str(x)).strip()

reader = pd.read_csv(
    INPUT_CSV,
    usecols=["text", "generated"],
    chunksize=25_000,          # safe for Colab
    engine="python",
    on_bad_lines="skip"
)

first_write = True
total_rows = 0

for chunk in reader:
    # drop bad rows
    chunk = chunk.dropna(subset=["text", "generated"])

    # label
    chunk["label"] = chunk["generated"].astype(int)

    # clean text
    chunk["text"] = chunk["text"].astype(str).apply(clean_text)

    # cheap length filter (NO split yet)
    chunk = chunk[chunk["text"].str.len() > 50]

    # keep only what you want
    chunk = chunk[["text", "label"]]

    # append to output CSV
    chunk.to_csv(
        OUTPUT_CSV,
        mode="w" if first_write else "a",
        index=False,
        header=first_write
    )

    total_rows += len(chunk)
    first_write = False
    print(f"Written so far: {total_rows}")

print("✅ DONE. Final rows:", total_rows)
print("Saved to:", OUTPUT_CSV)
```

```
Written so far: 24995
Written so far: 49989
```

```
Written so far: 74989
Written so far: 99985
Written so far: 124985
Written so far: 149985
Written so far: 174984
Written so far: 199984
Written so far: 224984
Written so far: 249984
Written so far: 274983
Written so far: 299983
Written so far: 324983
Written so far: 349983
Written so far: 374983
Written so far: 399983
Written so far: 424983
Written so far: 449983
Written so far: 474983
Written so far: 487218
✅ DONE. Final rows: 487218
Saved to: /content/drive/MyDrive/kaggle_text_label.csv
```

```
kaggle_path = "/content/drive/MyDrive/kaggle_text_label.csv"

if os.path.exists(kaggle_path):
    print(f"Found Kaggle file at: {kaggle_path}")
    # Load the file into the 'kag' variable
    kag = pd.read_csv(kaggle_path)
else:
    raise FileNotFoundError(f"Could not find {kaggle_path}. Make sure the file was actually saved there.")
```

```
Found Kaggle file at: /content/drive/MyDrive/kaggle_text_label.csv
```

```
final_output_path = "/content/drive/MyDrive/final_merged_dataset.csv"
scraped_data_path = "/content/drive/MyDrive/scraped_data_combined.csv"

# Load
kag_df = pd.read_csv(kaggle_path, low_memory=False)
scraped_df = pd.read_csv(scraped_data_path)

print("Merging datasets...")
full_df = pd.concat([kag_df, scraped_df], ignore_index=True)
# It is crucial to shuffle so the model doesn't see all "Human" samples then all "AI" samples
full_df = full_df.sample(frac=1, random_state=42).reset_index(drop=True)

print(f"Saving final dataset to {final_output_path}...")
full_df.to_csv(final_output_path, index=False)

print("\nSUCCESS! Pipeline Complete.")
print(f"Final Dataset Shape: {full_df.shape}")
print("\nSource Distribution:")
print(full_df["source"].value_counts())
print("\nLabel Distribution (0=Human, 1=AI):")
print(full_df["label"].value_counts())
```

```
Merging datasets...
Saving final dataset to /content/drive/MyDrive/final_merged_dataset.csv...

SUCCESS! Pipeline Complete.
Final Dataset Shape: (505311, 4)
```

```
Source Distribution:
source
chatgpt      12179
wikipedia     3354
news          1667
arxiv          893
Name: count, dtype: int64

Label Distribution (0=Human, 1=AI):
label
0    311711
1    193600
Name: count, dtype: int64
```