jlnin231d

December 20, 2024

1 Ejercicio 1

- Se prepara el entorno

[51]:
```
!pip install bm25s[full]
```

```
Requirement already satisfied: bm25s[full] in /usr/local/lib/python3.10/dist-
packages (0.2.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages
(from bm25s[full]) (1.13.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(from bm25s[full]) (1.26.4)
Requirement already satisfied: orjson in /usr/local/lib/python3.10/dist-packages
(from bm25s[full]) (3.10.12)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(from bm25s[full]) (4.67.1)
Requirement already satisfied: PyStemmer in /usr/local/lib/python3.10/dist-
packages (from bm25s[full]) (2.2.0.3)
Requirement already satisfied: numba in /usr/local/lib/python3.10/dist-packages
(from bm25s[full]) (0.60.0)
Requirement already satisfied: huggingface-hub in
/usr/local/lib/python3.10/dist-packages (from bm25s[full]) (0.27.0)
Requirement already satisfied: black in /usr/local/lib/python3.10/dist-packages
(from bm25s[full]) (24.10.0)
Requirement already satisfied: jax[cpu] in /usr/local/lib/python3.10/dist-
packages (from bm25s[full]) (0.4.33)
Requirement already satisfied: pytrec-eval in /usr/local/lib/python3.10/dist-
packages (from bm25s[full]) (0.5)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.10/dist-
packages (from black->bm25s[full]) (8.1.7)
Requirement already satisfied: mypy-extensions>=0.4.3 in
/usr/local/lib/python3.10/dist-packages (from black->bm25s[full]) (1.0.0)
Requirement already satisfied: packaging>=22.0 in
/usr/local/lib/python3.10/dist-packages (from black->bm25s[full]) (24.2)
Requirement already satisfied: pathspec>=0.9.0 in
/usr/local/lib/python3.10/dist-packages (from black->bm25s[full]) (0.12.1)
Requirement already satisfied: platformdirs>=2 in
/usr/local/lib/python3.10/dist-packages (from black->bm25s[full]) (4.3.6)
```

```
Requirement already satisfied: tomli>=1.1.0 in /usr/local/lib/python3.10/dist-
packages (from black->bm25s[full]) (2.2.1)
Requirement already satisfied: typing-extensions>=4.0.1 in
/usr/local/lib/python3.10/dist-packages (from black->bm25s[full]) (4.12.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from huggingface-hub->bm25s[full]) (3.16.1)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub->bm25s[full])
(2024.10.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-
packages (from huggingface-hub->bm25s[full]) (6.0.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from huggingface-hub->bm25s[full]) (2.32.3)
Requirement already satisfied: jaxlib<=0.4.33,>=0.4.33 in
/usr/local/lib/python3.10/dist-packages (from jax[cpu]; extra ==
"full"->bm25s[full]) (0.4.33)
Requirement already satisfied: ml-dtypes>=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from jax[cpu]; extra ==
"full"->bm25s[full]) (0.4.1)
Requirement already satisfied: opt-einsum in /usr/local/lib/python3.10/dist-
packages (from jax[cpu]; extra == "full"->bm25s[full]) (3.4.0)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in
/usr/local/lib/python3.10/dist-packages (from numba->bm25s[full]) (0.43.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub->bm25s[full]) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests->huggingface-hub->bm25s[full]) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub->bm25s[full]) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub->bm25s[full]) (2024.12.14)
```

Descargamos el fichero

[52]:
```python
import gdown
import zipfile
import os

# URL de descarga directa de Google Drive
url = "https://drive.google.com/uc?
 ↪export=download&id=17hrYKedsi_5t59UWGN0bvYBrcUh5hHKI"

# Descargar el archivo .zip
output = "trec-covid-RI.zip"
gdown.download(url, output, quiet=False)
```

```
# Descomprimir el archivo .zip
with zipfile.ZipFile(output, 'r') as zip_ref:
    zip_ref.extractall("/content/trec-covid-RI")

print("Archivo descargado y descomprimido exitosamente.")
```

```
Downloading…
From (original):
https://drive.google.com/uc?export=download&id=17hrYKedsi_5t59UWGN0bvYBrcUh5hHKI
From (redirected): https://drive.google.com/uc?export=download&id=17hrYKedsi_5t5
9UWGN0bvYBrcUh5hHKI&confirm=t&uuid=07c654ac-c54d-4a3c-b7f0-a3d46a137f43
To: /content/trec-covid-RI.zip
100%|      | 71.7M/71.7M [00:00<00:00, 163MB/s]

Archivo descargado y descomprimido exitosamente.
```

Se carga el archivo corpus y se prepara para poder ser indexado

[56]:
```
import bm25s
import json
import Stemmer # Para estematizar términos (Opcional para este ejercicio)


# Cargar la colección TREC-COVID desde el archivo corpus.jsonl
with open('/content/trec-covid-RI/corpus.jsonl', 'r', encoding='utf-8') as f:
    corpus_content = [json.loads(line) for line in f]

# Preparar el corpus para BM25S en dos formatos:
# 1. Una versión literal que mantenga la estructura original de los documentos.
# 2. Una versión de texto plano para tokenización e indexación.
corpus_verbatim = list()
corpus_plaintext = list()

for entry in corpus_content:
    document = {
        "id": entry["_id"],
        "title": entry["title"].lower(),
        "text": entry["text"].lower()
    }
    corpus_verbatim.append(document)
    # Incluimos tanto el título como el texto en la versión de texto plano
    corpus_plaintext.append(f"{entry['title'].lower()} {entry['text'].lower()}")

# Procesar el texto aplicando un stemmer para el idioma inglés y eliminando␣
 ↪palabras vacías durante la tokenización
stemmer = Stemmer.Stemmer("english")
```

```
corpus_tokenized = bm25s.tokenize(corpus_plaintext, stopwords="en",␣
 ↪stemmer=stemmer, show_progress=True)
```

```
Split strings:   0%|          | 0/171332 [00:00<?, ?it/s]
```

```
Stem Tokens:   0%|          | 0/171332 [00:00<?, ?it/s]
```

Se crea el índice BM25

[54]:
```
# Crear el retriever BM25 e indexar el corpus tokenizado
bm25_flavor = "lucene"
idf_flavor = "lucene"
retriever = bm25s.BM25(corpus=corpus_verbatim, method=bm25_flavor,␣
 ↪idf_method=idf_flavor)
retriever.index(corpus_tokenized, show_progress=True)

# Guardar el índice en una carpeta local
index_folder = "TREC-COVID_index"
retriever.save(index_folder, corpus=corpus_verbatim)

print(f"Indexación completada y guardada como '{index_folder}'.")
```

```
DEBUG:bm25s:Building index from IDs objects
```

```
BM25S Count Tokens:   0%|          | 0/171332 [00:00<?, ?it/s]
```

```
BM25S Compute Scores:   0%|          | 0/171332 [00:00<?, ?it/s]
```

```
Finding newlines for mmindex:   0%|          | 0.00/199M [00:00<?, ?B/s]
```

```
Indexación completada y guardada como 'TREC-COVID_index'.
```

Guardamos el índice (Opción para guardar en mi espacio personal) > **RECOMENDADIÓN**: NO ejecutar si no eres el autor del notebook

[55]:
```
import shutil # copia de archivos

# Guardar el índice en mi carpeta de OneDrive
onedrive_folder = "/content/drive/MyDrive/Colab Notebooks/
 ↪RecuperacionInformacion/TREC-COVID_index"
if not os.path.exists(onedrive_folder):
    os.makedirs(onedrive_folder)

for file_name in os.listdir(index_folder):
    full_file_name = os.path.join(index_folder, file_name)
    if os.path.isfile(full_file_name):
        shutil.copy(full_file_name, onedrive_folder)

print(f"Indexación completada y guardada como '{index_folder}' y en OneDrive␣
 ↪'{onedrive_folder}'.")
```

```
Indexación completada y guardada como 'TREC-COVID_index' y en OneDrive
'/content/drive/MyDrive/Colab Notebooks/RecuperacionInformacion/TREC-
COVID_index'.
```