COMP472 – Project 1 Demo

Rhina Kim – 40130779

September 22, 2022

On top of every pipeline step described in the report.pdf, this demo file walks through demo of the script.

The below scripts are a function which calls every pipeline function described in repord.pdf.

# pipline function

def pipeline(directory, output\_directory, test=False, max\_sgm\_files=-1):

    PIPELINE\_STEP = 0  # pipeline step number requied for output file name

    NUM\_SGM\_FILES = 0  # number of sgm files read from reuters21578

    # first, decide list of stop words to be used for all documents

    stop\_words = create\_stop\_words\_list()

    # iterate all the files inside the folder

    for file in os.listdir(directory):

        # [DEMO] numeber of maximum files to be retrived

        if test and NUM\_SGM\_FILES == max\_sgm\_files:

            break

        # if not sgm file then skip

        if not file.endswith(".sgm"):

            continue

        # if sgm file, count sgm files

        NUM\_SGM\_FILES += 1

        # open the file to be read

        filename = os.path.join(directory, file)

        sgm\_file = open(filename, 'r', encoding='utf-8', errors='ignore') # avoid UnicodeDecodeError with 'r'

        # start the pipeline step

        print("\nFile #(" + str(NUM\_SGM\_FILES) + ")")

        print("\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  Pipeline [1]  \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*")

        PIPELINE\_STEP = 1

        documents = extract\_documents\_from\_corpus(sgm\_file)

        # [DEMO] output the result

        if test and output\_directory:

            file\_output(documents, NUM\_SGM\_FILES, PIPELINE\_STEP, OUTPUT\_DIRECTORY)

        print("\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  Pipeline [2]  \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*")

        PIPELINE\_STEP = 2

        postings = tokenize(documents)

        # [DEMO] output the result

        if test and output\_directory:

            file\_output(postings, NUM\_SGM\_FILES, PIPELINE\_STEP, OUTPUT\_DIRECTORY)

        print("\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  Pipeline [3]  \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*")

        PIPELINE\_STEP = 3

        postings = lowercase(postings)

        # [DEMO] output the result

        if test and output\_directory:

            file\_output(postings, NUM\_SGM\_FILES, PIPELINE\_STEP, OUTPUT\_DIRECTORY)

        print("\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  Pipeline [4]  \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*")

        PIPELINE\_STEP = 4

        postings = porter\_stemmer(postings)

        # [DEMO] output the result

        if test and output\_directory:

            file\_output(postings, NUM\_SGM\_FILES, PIPELINE\_STEP, OUTPUT\_DIRECTORY)

        print("\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  Pipeline [5]  \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*")

        PIPELINE\_STEP = 5

        postings = remove\_stop\_words(postings, stop\_words)

        # [DEMO] output the result

        if test and output\_directory:

            file\_output(postings, NUM\_SGM\_FILES, PIPELINE\_STEP, OUTPUT\_DIRECTORY)

### FOR TESTING PURPOSE ###

DIRECTORY = "reuters21578\_extracted/"

OUTPUT\_DIRECTORY = "outputs\_test/"

MAX\_SGM\_FILES = 5  # [DEMO] for testing purpose

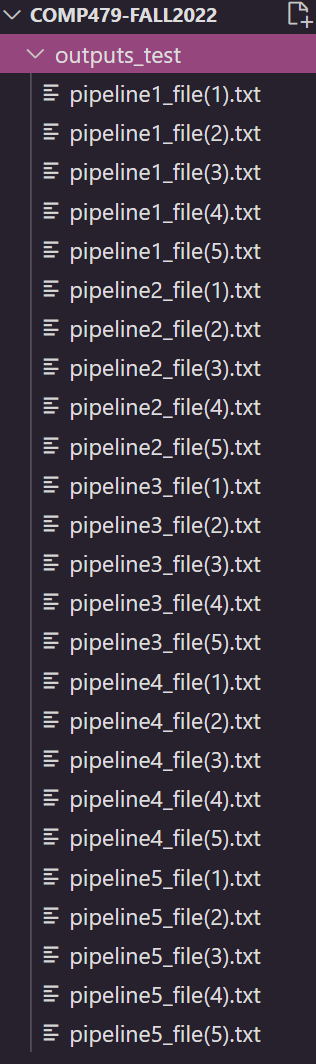
pipeline(DIRECTORY, OUTPUT\_DIRECTORY, True, MAX\_SGM\_FILES)

The function called “***file\_output()***” outputs the result list to the specified file name.

All the result outputs are stored inside the folder “***outputs\_test***”. Since we only need output files of five modules (five pipeline steps) for the first five sgm files in the collections (for testing/demo purpose), we will have total of 25 files.

The convention of the file name is as follow:

***“pipeline#-file(#).txt”*** where pipeline # describes the pipeline step, and file# describes the unique file ID for every 5 files used for testing/demo purpose.



On the console, we locate the command address to the folder name “COMP479-FALL2022”, and we type “p1.py” in the command to perform the script.

Weakness of this script:

* Too many iterations happening (we iterate postings list inside every pipeline function)
* Further analysis needed for removing special characters and punctuations (ex. You’re 🡪 youre, is that appropriate?)