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)
```

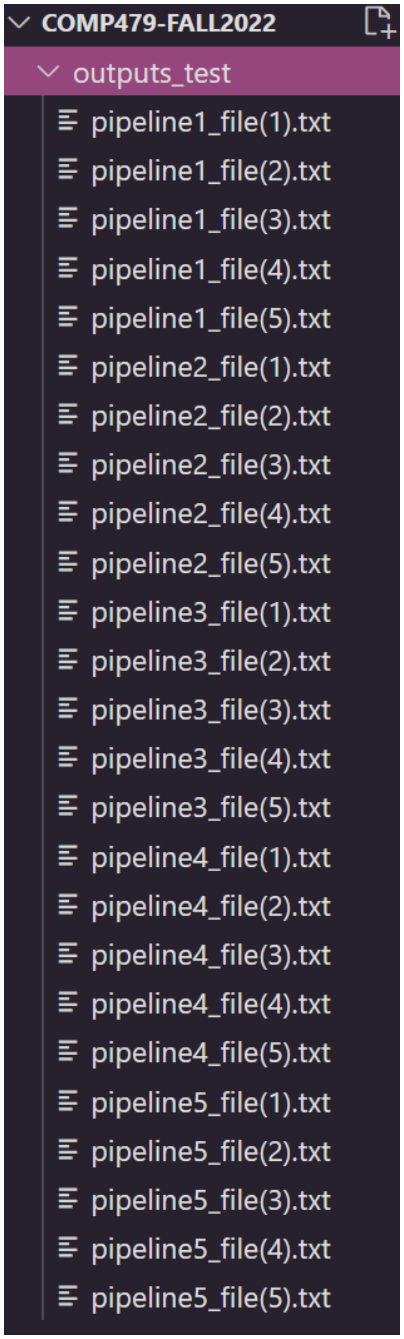
```
        # [DEMO] output the result
        if test and output_directory:
            file_output(postings, NUM_SGM_FILES, PIPELINE_STEP, OUTPUT_DIRECTORY)

        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?)