

## HW1 – Report

1.

- a. Precision of a search engine is a metric equal to the number of relevant items found over the total number of items returned by the search. Recall on the other hand is a metric equal to the number of relevant items found over the total number of relevant items in the collection. The two metrics are generally inversely proportional meaning that steps taken to increase one will usually decrease the other.

Both metrics give, on their own, an incomplete view of the quality of a search engine. 100% recall can be achieved by simply returning every item in the collection. However this renders the search engine useless as it returns a huge amount of mismatches. Precision can be increased by returning fewer items to the user but then they run an increased risk of not being shown what they are searching for, especially given that the same query can be used by two users searching for very different results. Therefore, to effectively evaluate a search engine, one must consider both precision and recall.

- b. A technique that would enhance precision in a library card indexing system would be to limit each book to three index cards: the obligatory title and author cards, and one card for the main topic of the book. Anyone looking for information about Salmon for example, would not find books about fish in general, fishing, cooking, or river ecosystems. They would only find books on Salmon.

A technique that would enhance recall in a library indexing system would be to make many cards for every book for every topic broached within. A book about Salmon might have cards under marine ecosystems, river currents, bear nutrition, mercury poisoning, migration patterns, water pressure.

2. At a query level, databases are very similar to search engines. You can do an advanced search (or formal query) on a search engine, structured much like a database query. However, on a results level, they are very different. Databases will always return the correct results to a query as there is no ambiguity. If you wanted lines containing the number 56, then you will get all the lines containing the number 56. Therefore, there is 100% user satisfaction. There is no concept of better or worse results. In a search engine, the user can be dissatisfied with their results as the correctness of the results is entirely dependent on the user. A user searching the number 56 in the hope of finding a new bar called “56” would be disappointed when they find “56 reasons your cat might have cancer”. There could potentially be a trade-off between speed and quality of results for a search engine and improving and measuring quality takes much effort.
3. The advantage to down casing all the terms in a search engine are that it increases recall. One person might write “MacBook” but they may want to see reviews in which the reviewers referred to it as “macbook”. Down casing increases the number of matching tuples and thus recall. However, the disadvantage of down casing is that it can reduce precision, especially in the case of acronyms or names that are also words such as “Max”.

In these cases the search engine will match with documents that the user might not want to see.

4.

- a. The IndexEngine program loops over the latimes.gz file line by line to extract information. If a line contains a tag indicating the start of a document, it begins saving each line to an array and resets all the variables containing the metadata of the previous document. As it loops through the document, when tags indicate metadata, it is parsed and passed to the appropriate variables. When the loop reaches the tag indicating the end of the document, the variables containing metadata are assigned to a dictionary with the type of data as the keys (ex: "HEADLINE"). This dictionary is then added (nested in) to another dictionary which has the metadata of each document and the docno as the keys. The id and docno are also added as one entry to a second dictionary with the id as the key to aid in the ease of retrieval for the GetDoc program. The array containing the document is written to a text file, named with the docno, which is placed in a folder named for the date of the doc (the folder is created if it does not already exist). This is done for every document and then the two dictionaries with the metadata and the id-docno conversions are pickled.

The GetDoc program accepts either the id or the docno of the desired document. If it receives the id, then it uses the pickled conversion dictionary to get the docno. The docno is then used to get the metadata from the pickled metadata dictionary. The date (from metadata) and the docno are used to complete the file path to the document which is then retrieved. The date is converted from number format to written format and the metadata is outputted. The file path is used to access the document which is then looped over and printed as well.

- b. For the purposes of brevity and clarity, the screenshot corresponding to each test will be embedded in the test plan below the corresponding test. In each screenshot with the command line, the test is the most recent command, unless otherwise specified

### IndexEngine tests:

After calling: `python IndexEngine.py`

`C:\Users\thoma\Documents\3Bterm\MSCI541\latimes.gz`

`C:\Users\thoma\Documents\3Bterm\MSCI541\testdir`

- i. Integrity of the metadata is tested by using a (not yet complete) version of the GetDoc program to print the metadata of a document:

```
14 #Retrieved from: https://stackoverflow.com/questions/1118477/how-can-i-use-pickle-to-save-a-dict-on-a
15 with open("idconvert.pickle", "rb") as convert:
16     idconvert = pickle.load(convert)
17     if docInputUpper.find("DOCNO") != -1:
18         docNo = idNo
19     elif docInputUpper.find("ID") != -1:
20         idNo = str(idNo)
21         docNo = idconvert[idNo]
22     else:
23         sys.exit("Enter either a DOCNO or an id")
24 print(docNo)
25 with open("metaData.pickle", "rb") as mData:
26     metaData = pickle.load(mData)
27     docMetaData = metaData[docNo]
28     print(docMetaData)
29
```

TERMINAL    RUNNER    PROBLEMS    OUTPUT    DEBUG CONSOLE

```
PS C:\Users\thoma\Documents\3Bterm\MSCI541\testdir> python GetDoc.py C:\Users\thoma\Documents\3Bterm\MSCI541\testdir docno LA123190-0134
{"DOCNO": "LA123190-0134", "DOCID": "329701", "DATE": "123190", "HEADLINE": "SHORT TAKES; TAMMY SEES COUNTRY'S REBIRTH "}
```

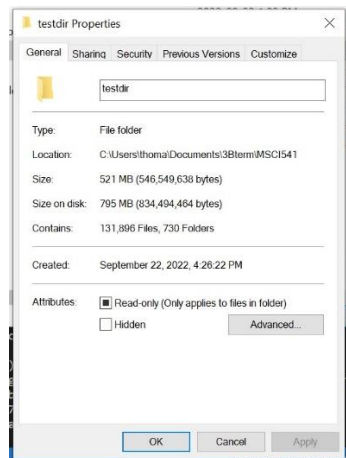
PS C:\Users\thoma\Documents\3Bterm\MSCI541\testdir>

- ii. The integrity of the pickled id conversion is tested by using a not yet complete version of the GetDoc program calling the id and printing the docno.

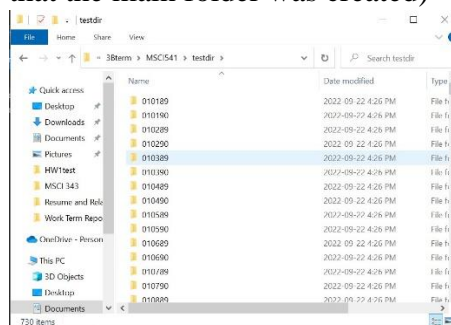
```
GetDoc.py > ...
5
6 if not len(sys.argv) > 2:
7     sys.exit("Enter correct arguments please")
8 filePath = sys.argv[1]
9 docInput = sys.argv[2]
10 idNo = sys.argv[3]
11 docInputUpper = docInput.upper()
12 docNo = ""
13 with open("idConvert.pickle", "rb") as convert:
14     idConvert = pickle.load(convert)
15     if docInputUpper.find("DOCNO") != -1:
16         docNo = idNo
17     elif docInputUpper.find("ID") != -1:
18         idNo = str(idNo)
19         docNo = idConvert[idNo]
20     else:
21         sys.exit("Enter either a DOCNO or an id")
22 print(docNo)
23
```

docNo = idConvert[idNo]  
KeyError: '329701'  
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> python IndexEngine.py C:\Users\thoma\Documents\3Bterm\MSCI541\latimes.gz C:\Users\thoma\Documents\3Bterm\MSCI541\testdir  
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> python GetDoc.py C:\Users\thoma\Documents\3Bterm\MSCI541\testdir id 329701  
LA123190-0134  
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> []

- iii. The correctness of the doc tag logic and looping is tested by checking that there are indeed 131 896 documents created



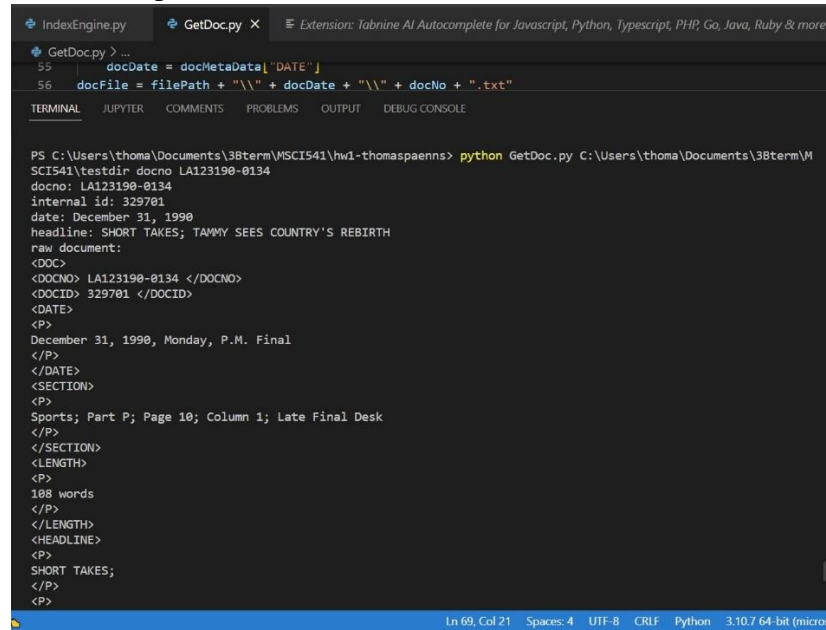
- iv. The folder creation and correct file paths are tested by ensuring that the date folders are created inside the main folder (this screenshot also shows that the main folder was created)





**GetDoc tests** (These tests also indirectly test the parsing of the IndexEngine program as all output is dependent on the IndexEngine program having run correctly. Thus these tests also test metadata and document parsing.)

- viii. The document retrieval with docno and printing as well as the metadata retrieval and printing is tested by calling the program with a docno. Note that the output matches the document shown in test “V” (see above)



```
IndexEngine.py  GetDoc.py x  Extension: Tabnine AI Autocomplete for Javascript, Python, Typescript, PHP, Go, Java, Ruby & more

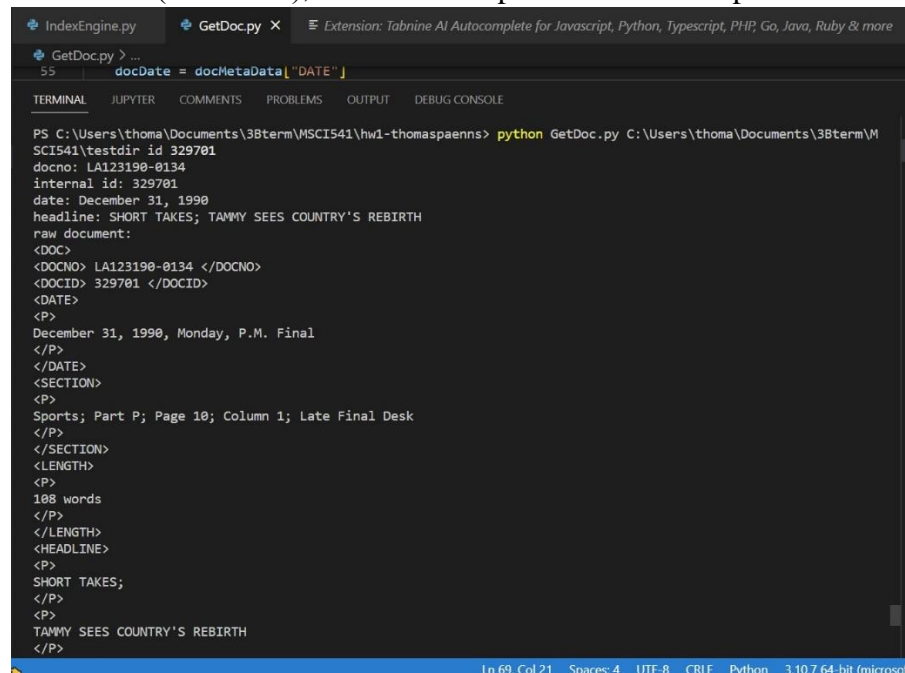
GetDoc.py > ...
55 docDate = docMetadata["DATE"]
56 docFile = filePath + "\\\" + docDate + "\\\" + docNo + ".txt"

TERMINAL  JUPYTER  COMMENTS  PROBLEMS  OUTPUT  DEBUG CONSOLE

PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> python GetDoc.py C:\Users\thoma\Documents\3Bterm\MSCI541\testdir docno LA123190-0134
docno: LA123190-0134
internal id: 329701
date: December 31, 1990
headline: SHORT TAKES; TAMMY SEES COUNTRY'S REBIRTH
raw document:
<DOC>
<DOCNO> LA123190-0134 </DOCNO>
<DOCID> 329701 </DOCID>
<DATE>
<P>
December 31, 1990, Monday, P.M. Final
</P>
</DATE>
<SECTION>
<P>
Sports; Part P; Page 10; Column 1; Late Final Desk
</P>
</SECTION>
<LENGTH>
<P>
188 words
</P>
</LENGTH>
<HEADLINE>
<P>
SHORT TAKES;
</P>
<P>
TAMMY SEES COUNTRY'S REBIRTH
</P>

Ln 69, Col 21  Spaces: 4  UTF-8  CRLF  Python  3.10.7 64-bit (microso
```

- ix. The document retrieval with id, id to docno conversion conversion, printing, as well as the metadata retrieval and printing is tested by calling the program with an id. Note that the output matches the document shown in test “V” (see above), as well as the previous test’s output



```
IndexEngine.py  GetDoc.py x  Extension: Tabnine AI Autocomplete for Javascript, Python, Typescript, PHP, Go, Java, Ruby & more

GetDoc.py > ...
55 docDate = docMetadata["DATE"]

TERMINAL  JUPYTER  COMMENTS  PROBLEMS  OUTPUT  DEBUG CONSOLE

PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> python GetDoc.py C:\Users\thoma\Documents\3Bterm\MSCI541\testdir id 329701
docno: LA123190-0134
internal id: 329701
date: December 31, 1990
headline: SHORT TAKES; TAMMY SEES COUNTRY'S REBIRTH
raw document:
<DOC>
<DOCNO> LA123190-0134 </DOCNO>
<DOCID> 329701 </DOCID>
<DATE>
<P>
December 31, 1990, Monday, P.M. Final
</P>
</DATE>
<SECTION>
<P>
Sports; Part P; Page 10; Column 1; Late Final Desk
</P>
</SECTION>
<LENGTH>
<P>
188 words
</P>
</LENGTH>
<HEADLINE>
<P>
SHORT TAKES;
</P>
<P>
TAMMY SEES COUNTRY'S REBIRTH
</P>

Ln 69, Col 21  Spaces: 4  UTF-8  CRLF  Python  3.10.7 64-bit (microso
```



- x. The invalid docno and invalid id exceptions are tested by calling the program with each. Note the last two calls in the below screenshot

```
41 docMetaData = {}
42 #Retrieved from: https://stackoverflow.com/questions/11218477/how-can-i-use-pickle-to-save-a-dict-or-an
43 with open("idConvert.pickle", "rb") as convert:
44     idConvert = pickle.load(convert)
45     if docInputUpper.find("DOCNO") != -1:
46         docNo = idNo
47     elif docInputUpper.find("ID") != -1:
48         idNo = str(idNo)
49         if not idNo in idConvert.keys():
50             sys.exit("Enter a valid ID")
51         docNo = idConvert[idNo]
52     else:
53         sys.exit("Enter either a DOCNO or an id")
54
55 with open("metaData.pickle", "rb") as mData:
56     metaData = pickle.load(mData)
57     if not docNo in metaData.keys():
58         sys.exit("Enter a valid DOCNO")
59     docMetaData = metaData[docNo]
60     docDate = docMetaData["DATE"]
```

TERMINAL JUPYTER COMMENTS PROBLEMS OUTPUT DEBUG CONSOLE

```
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> python GetDoc.py C:\Users\thoma\Documents\3Bterm\M
SCI541\testdir docname LA123190-0134
Enter either a DOCNO or an id
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> python GetDoc.py C:\Users\thoma\Documents\3Bterm\M
SCI541\testdir docno LA123190-0138
Enter a valid DOCNO
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> python GetDoc.py C:\Users\thoma\Documents\3Bterm\M
SCI541\testdir id 32970123
Enter a valid ID
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns>
```

- xi. The invalid filepath exception is tested by passing an invalid filepath

```
41 docMetaData = {}
42 #Retrieved from: https://stackoverflow.com/questions/11218477/how-can-i-use-pickle-to-save-a-dict-or-an
43 with open("idConvert.pickle", "rb") as convert:
44     idConvert = pickle.load(convert)
45     if docInputUpper.find("DOCNO") != -1:
46         docNo = idNo
47     elif docInputUpper.find("ID") != -1:
48         idNo = str(idNo)
49         if not idNo in idConvert.keys():
50             sys.exit("Enter a valid ID")
51         docNo = idConvert[idNo]
52     else:
53         sys.exit("Enter either a DOCNO or an id")
54
55 with open("metaData.pickle", "rb") as mData:
56     metaData = pickle.load(mData)
57     if not docNo in metaData.keys():
58         sys.exit("Enter a valid DOCNO")
59     docMetaData = metaData[docNo]
60     docDate = docMetaData["DATE"]
```

TERMINAL JUPYTER COMMENTS PROBLEMS OUTPUT DEBUG CONSOLE

```
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> python GetDoc.py C:\Users\thoma\Documents\3Bterm\M
SCI541\testdir docno LA123190-0138
Enter a valid DOCNO
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> python GetDoc.py C:\Users\thoma\Documents\3Bterm\M
SCI541\testdir id 32970123
Enter a valid ID
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> python GetDoc.py C:\Users\thoma\Documents\3Bterm\M
SCI541\testdirectory docno LA123190-0134
File does not exist
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns>
```

- xii. The too few arguments exception is tested by passing too few arguments (help message is returned)

```
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> python IndexEngine.py C:\Users\thoma\Documents\3Bterm\M
SCI541\latimes.gz
Enter Filepaths Please
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> python IndexEngine.py
Enter Filepaths Please
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> python GetDoc.py C:\Users\thoma\Documents\3Bterm\MSCI54
1\testdirectory docno
Enter correct arguments please. The correct arguments are: *FilePath to the documents* "docno"/"id" *DOCNO"/"ID"
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns>
```

- xiii. The wrong string exception (passing a string arg that is neither “id” nor “docno”) is tested by passing an invalid string.

```
TERMINAL  JUPYTER  COMMENTS  PROBLEMS  OUTPUT  DEBUG CONSOLE
<P>
Brief; Wire
</P>
</TYPE>
</DOC>
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> python GetDoc.py C:\Users\thoma\Documents\3Bterm\MSCI541\testdir docname LA123190-0134
Enter either a DOCNO or an id
PS C:\Users\thoma\Documents\3Bterm\MSCI541\hw1-thomaspenns> 
```