Section 1

Monday’s Lecture Question

Pick a collection to crawl from the web. This could be something you already did in the previous assignment or (for more practice), a new collection. Either way, make sure you have 50-100 “good” documents to index and then index it using one of the three retrieval models we saw in the class. Make sure you use appropriate meta tags in your index. Provide your wget command(s) and the index statistics. [4 points]

Monday’s Lecture Answer

```
# I ran: wget -r paulgraham.com

import pyterrier as pt
if not pt.started():
  pt.init()
```

```
C:\Users\Fortu\AppData\Local\Temp\ipykernel_33888\2621541688.py:4: DeprecationWarning: Call to
deprecated function (or staticmethod) started. (use pt.java.started() instead) -- Deprecated
since version 0.11.0.
  if not pt.started():
Java started and loaded: pyterrier.java, pyterrier.terrier.java [version=5.10 (build: craigm
2024-08-22 17:33), helper_version=0.0.8]
C:\Users\Fortu\AppData\Local\Temp\ipykernel_33888\2621541688.py:5: DeprecationWarning: Call to
deprecated method pt.init(). Deprecated since version 0.11.0.
java is now started automatically with default settings. To force initialisation early, run:
pt.java.init() # optional, forces java initialisation
  pt.init()
```

```
files = pt.io.find_files('c:/Users/Fortu/Downloads/Aut2024/Info 376/paulgraham.com')
indexer = pt.FilesIndexer('c:/Users/Fortu/Downloads/Aut2024/Info 376/a_5_index',
                          verbose=True, blocks=False,
                          meta={"docno":20,"filename":1024,"title":1024},meta_tags={"title":"titl
indexref = indexer.index(files)
index = pt.IndexFactory.of(indexref)
print(index.getCollectionStatistics().toString())
```

```
Number of documents: 101
Number of terms: 6637
Number of postings: 43543
Number of fields: 0
Number of tokens: 111745
```

```
Field names: []
Positions:   false
```

```
import pandas as pd
queries = pd.DataFrame([["q1","speak"], ["q2","indulge"], ["q3", "swans"]], columns=["qid","query
queries
```

	qid	query
0	q1	speak
1	q2	indulge
2	q3	swans

```
# making sure I have the right details

print(index.getMetaIndex().getKeys())
print(index.getMetaIndex().getItem("filename", 15))
print(index.getMetaIndex().getItem("title", 15))
```

```
['docno', 'filename', 'title']
c:/Users/Fortu/Downloads/Aut2024/Info 376/paulgraham.com\books.html
Books
```

```
# Similar script as assignment 4

index = pt.IndexFactory.of("c:/Users/Fortu/Downloads/Aut2024/Info 376/a_5_index/data.properties")
BM = pt.BatchRetrieve(index, wmodel="BM25")
BM.transform(queries)
```

```
C:\Users\Fortu\AppData\Local\Temp\ipykernel_33888\3203623510.py:4: DeprecationWarning: Call to
deprecated class BatchRetrieve. (use pt.terrier.Retriever() instead) -- Deprecated since version
0.11.0.
  BM = pt.BatchRetrieve(index, wmodel="BM25")
```

	qid	docid	docno	rank	score	query
0	q1	81	d82	0	4.880851	speak
1	q1	63	d64	1	3.116216	speak
2	q1	86	d87	2	3.016180	speak
3	q1	9	d10	3	2.731277	speak
4	q1	12	d13	4	2.697123	speak
5	q1	99	d100	5	2.672519	speak
6	q1	47	d48	6	2.439291	speak

	qid	docid	docno	rank	score	query
7	q1	44	d45	7	1.944379	speak
8	q1	87	d88	8	1.934287	speak
9	q1	17	d18	9	1.930946	speak
10	q1	67	d68	10	1.614475	speak
11	q1	37	d38	11	1.539273	speak
12	q1	82	d83	12	1.334695	speak
13	q1	69	d70	13	1.247106	speak
14	q1	31	d32	14	1.001525	speak
15	q2	78	d79	0	6.239193	indulge
16	q2	10	d11	1	4.223424	indulge
17	q2	87	d88	2	3.780832	indulge
18	q2	94	d95	3	3.573318	indulge
19	q2	99	d100	4	3.537728	indulge
20	q2	47	d48	5	3.228994	indulge
21	q2	61	d62	6	3.029599	indulge
22	q2	82	d83	7	1.766793	indulge
23	q2	36	d37	8	1.292570	indulge
24	q3	85	d86	0	6.783079	swans
25	q3	4	d5	1	5.811284	swans
26	q3	9	d10	2	4.886159	swans
27	q3	23	d24	3	3.974995	swans

```
index = pt.IndexFactory.of("c:/Users/Fortu/Downloads/Aut2024/Info 376/a_5_index/data.properties")
TF_IDF = pt.BatchRetrieve(index, wmodel="TF_IDF")
TF_IDF.transform(queries)
```

```
C:\Users\Fortu\AppData\Local\Temp\ipykernel_33888\1815859450.py:2: DeprecationWarning: Call to
deprecated class BatchRetrieve. (use pt.terrier.Retriever() instead) -- Deprecated since version
0.11.0.
  TF_IDF = pt.BatchRetrieve(index, wmodel="TF_IDF")
```

	qid	docid	docno	rank	score	query
0	q1	81	d82	0	3.167447	speak
1	q1	63	d64	1	2.022280	speak
2	q1	86	d87	2	1.957362	speak
3	q1	9	d10	3	1.772473	speak
4	q1	12	d13	4	1.750308	speak

	qid	docid	docno	rank	score	query
5	q1	99	d100	5	1.734342	speak
6	q1	47	d48	6	1.582987	speak
7	q1	44	d45	7	1.261812	speak
8	q1	87	d88	8	1.255263	speak
9	q1	17	d18	9	1.253095	speak
10	q1	67	d68	10	1.047720	speak
11	q1	37	d38	11	0.998917	speak
12	q1	82	d83	12	0.866155	speak
13	q1	69	d70	13	0.809314	speak
14	q1	31	d32	14	0.649943	speak
15	q2	78	d79	0	3.743138	indulge
16	q2	10	d11	1	2.533798	indulge
17	q2	87	d88	2	2.268270	indulge
18	q2	94	d95	3	2.143774	indulge
19	q2	99	d100	4	2.122423	indulge
20	q2	47	d48	5	1.937201	indulge
21	q2	61	d62	6	1.817576	indulge
22	q2	82	d83	7	1.059968	indulge
23	q2	36	d37	8	0.775463	indulge
24	q3	85	d86	0	3.930688	swans
25	q3	4	d5	1	3.367548	swans
26	q3	9	d10	2	2.831452	swans
27	q3	23	d24	3	2.303447	swans

```
index = pt.IndexFactory.of("c:/Users/Fortu/Downloads/Aut2024/Info 376/a_5_index/data.properties")
Hiem = pt.BatchRetrieve(index, wmodel="Hiemstra_LM")
Hiem.transform(queries)
```

```
C:\Users\Fortu\AppData\Local\Temp\ipykernel_33888\2466155938.py:2: DeprecationWarning: Call to
deprecated class BatchRetrieve. (use pt.terrier.Retriever() instead) -- Deprecated since version
0.11.0.
  Hiem = pt.BatchRetrieve(index, wmodel="Hiemstra_LM")
```

	qid	docid	docno	rank	score	query
0	q1	81	d82	0	3.217159	speak
1	q1	86	d87	1	1.209483	speak
2	q1	63	d64	2	1.078505	speak

	qid	docid	docno	rank	score	query
3	q1	9	d10	3	0.969115	speak
4	q1	99	d100	4	0.926711	speak
5	q1	12	d13	5	0.819993	speak
6	q1	47	d48	6	0.776729	speak
7	q1	44	d45	7	0.529729	speak
8	q1	87	d88	8	0.525455	speak
9	q1	17	d18	9	0.524046	speak
10	q1	67	d68	10	0.401997	speak
11	q1	37	d38	11	0.375958	speak
12	q1	82	d83	12	0.309978	speak
13	q1	69	d70	13	0.283698	speak
14	q1	31	d32	14	0.215503	speak
15	q2	78	d79	0	3.492314	indulge
16	q2	10	d11	1	1.401946	indulge
17	q2	94	d95	2	1.256705	indulge
18	q2	99	d100	3	1.233523	indulge
19	q2	87	d88	4	1.212981	indulge
20	q2	47	d48	5	1.049535	indulge
21	q2	61	d62	6	0.944553	indulge
22	q2	82	d83	7	0.443112	indulge
23	q2	36	d37	8	0.303084	indulge
24	q3	85	d86	0	3.150816	swans
25	q3	4	d5	1	3.006787	swans
26	q3	9	d10	2	2.272189	swans
27	q3	23	d24	3	1.728091	swans

Section 2

Wednesday’s Lecture Question

Build a simple web-based UI to interact with the index you built. Provide your link to that site. On the site, make sure you write one or two sentences about the collection that you have indexed and a couple of sample queries to try. [6 points]

Wednesday’s Lecture Answer

I would try searching with the queries mentioned above in this document. I have uploaded the same index. You should get the same results as the BM25.

Queries I recommend:

addiction, life, age

http://is-searchrec.ischool.uw.edu/swas/search.php