COL 764: Information Retrieval System Assignment-1

Shashank G (2022AIB2684)

August 31, 2023

1 Important Note

- All experiments were conducted using my personal laptop, equipped with an AMD 5800H processor. It is important to note that no multi-threading was employed at any point in the assignment.
- I also claim to have my index construction speed evaluated for bonus marks.
- All index and dictionary files are saved on hpc */home/scai/mtech/aib222684/scratch/irs1* with the format *indexfile_[compression-type]_[tokenizer-type].*{*dict* | *idx*}
  Eg. index file with simple tokenizer and with compression - *indexfile_1_0.idx*

2 Index Construction

I executed the **block sort-based indexing** algorithm to construct the index, as taught in our class. This particular algorithm entails sorting term-docid pairs and subsequently storing them on the disk before the merging process. It's essential to highlight that a substantial amount of disk space (around $6GB$) is necessary for this operation. The amount of memory(RAM) required for running these process is less than $< 1.5GB$.
No compression in the below tables indicate no compression along with delta encoding.

Below are the results for evaluation with **Simple** tokenizer.

Compression type	**Construction speed**	**Disk size**
No compression	$3m10s$	874 MB
Variable byte compression	$3m43s$	252 MB

Below are the results for evaluation with **BPE** tokenizer.
The indicated construction speed encompasses both the training time for BPE **AND** the index construction speed. The BPE is trained over $20,000$ iterations, a process that approximately takes 360 seconds. It also generates a ”merge_order” binary file which will be used to tokenize text.

Compression type	**Construction speed**	**Disk size**
No compression	$16m10s$	958 MB
Variable byte compression	$16m34s$	265 MB

3 Query Evaluation

I utilized term-at-a-time processing to assess all queries, involving the calculation of scores for individual documents across each term in the query. These scores were accumulated and subsequently normalized with the logarithm of the document's length. All queries were evaluated using metrics provided by Github page.

Below are the results for evaluation with **Simple** tokenizer.

Compression type	**Evaluation Time**	**Efficiency**
No compression	18s	0.18s
Variable byte compression	55s	0.55s

Measure	**Precision**	**Recall**	**F1-score**
@10	0.4000	0.1293	0.1728
@20	0.3280	0.2034	0.2155
@50	0.2196	0.3113	0.2192
@100	0.1561	0.4085	0.1944

Below are the results for evaluation with **BPE** tokenizer.

Compression type	**Evaluation Time**	**Efficiency**
No compression	20s	0.20s
Variable byte compression	59s	0.59s

Measure	**Precision**	**Recall**	**F1-score**
@10	0.3360	0.1096	0.1453
@20	0.2720	0.1683	0.1773
@50	0.1856	0.2658	0.1833
@100	0.1316	0.3431	0.1621