COL764: Assignment 1

Archisman Biswas, 2021MT10254

3 September 2024

Contents

Note: All time measures are given w.r.t programs run on Apple Silicon M2 chip

1 Tokenization

The full text consists of all text from the `title` and `abstract` fields of all documents. The Simple tokenizers was trained on this entire text while the BPE and WordPiece tokenizers were trained on the latter 50% documents, with the resulting vocabulary stored in the `output.dict` file. For tokenization, only English alphabets were considered. Although various types of characters were experimented with, using only English alphabets proved to be the most efficient in terms of both accuracy and time. This step generates the file `"output.dict"`. The respective tokenizers are as follows:

1.1 Simple Tokenizer

The delimiters used to tokenize are `[' ', ',', '.', ':', ';', '"', "'"]`
The size of vocabulary is `274052` and construction time is around `40s`. Disk size of `output.dict` is `3.1 MB`.

1.2 BPE Tokenizer

The delimiters used to tokenize are `['.', ' ', ':', ';', '"', "'", '.', '?', '!', ',', '\n', '~', '‘', '(', ')', '/', '#', '*', '%', '+', '-', '[', ']', '{', '}', '@', '^']`
BPE merges are an expensive process and each merge takes around `0.15s`. The vocabulary size obtained here is around `1800` hence, considering the time limit of `300s`. Disk size of `output.dict` is `10 KB`.

1.3 WordPiece Tokenizer

The delimiters used to to tokenize here are same as that of BPE. Characters have `"##"` appended before it, if it occurs in the middle of a word. The score for merging a pair (maximized for each merge) is computed as:

$$\frac{\text{frequency(merged token1 and token2)}}{\text{frequency(token1)} \times \text{frequency(token2)}}$$

Considering this to be equally expensive, the vocabulary size obtained here is around `1300`. Disk size of `output.dict` is `12 KB`. The learned merges look like this:

```
...
publically
amplifying
minimizing
##bilizing
##vailable
...
```

2 Inverted Index construction

There are two files which are created in this step: `<indexfile>.dict` to store the dictionary and `<indexfile>.idx` to store the index as posting lists along with some information.
`<indexfile>.dict` has the rows as `<term>:<offset>`, where offset is the difference from the beginning of the `.idx`, which is used to navigate to the respective postings. The `.dict` file looks as follows:

```
...
no:10310534
and:10593617
increased:12278283
...
```

In `<indexfile>.idx`, the first row is used to store the doc_ids of the documents it is indexed on (separated by a ;). After that, the posting lists are stored per row for each term.
The format is `<DF_term>;<Doc_id_1>:<TF_term_1>;<Doc_id_2>:<TF_term_2>;...`
The `.idx` file looks as follows:

```
02tnwd4m;ug7v899j;ejv2xln0;2b73a28n;...
383;02tnwd4m:2;g4puurhk:3;sn1a7ikq:2;...
mdej7nhj:1;ary5eafy:1;5s10b9zy:1;...
...
```

	`<indexfile>.dict`	`<indexfile>.idx`
Simple	5.8 MB	185.1 MB
BPE	29 KB	278.7 MB
WordPiece	17 KB	110.2 MB

Table 1: Disk size of .dict and .idx files

	Time (in seconds)
Simple	107.04
BPE	402.88
WordPiece	442.93

Table 2: Total time for construction of inverted index

3 Searching

The Vector-Space Model is used to score the relevance between a query and a document. Cosine similarity is calculated using the TDF-IDF scores:

$$tf_{ij} = \begin{cases} 1 + \log_2(f_{ij}) & \text{if, } f_{ij} \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$idf_i = \log_2\left(1 + \frac{N}{df_i}\right)$$

Note: IDF score for a query is always set to `1`
While searching, the inverted index and the dictionary is loaded to memory. The locations to those are given as inputs.

3.1 Query processing

The queries are processed in two parts:

- The `title` field string is doubled and then concatenated with the `description` field string. In this way we get to emphasize more on the keywords which the title is supposed to contain
- Stopwords in the above concatenated string is removed. A list of stopwords is hard-coded into.

The above steps help to improve the representation of the query as a single string of tokens, which are given to the model for computing the similarity scores with documents. Several other query processing methods were experimented with, like taking only the `title` field, simply concatenating `title` and `description` fields, taking only the top-k% frequent words from `description` field.

3.2 Results

3.2.1 Simple Tokenizer

Time taken: `67.06` seconds
Efficiency: `2.68` seconds

Query	Precision	Recall	F1@100	F1@50	F1@20	F1@10
1	0.39	0.0558	0.0976	0.0401	0.0167	0.0056
2	0.01	0.0030	0.0046	0.0000	0.0000	0.0000
3	0.27	0.0414	0.0718	0.0399	0.0268	0.0151
4	0.11	0.0194	0.0330	0.0194	0.0000	0.0000
5	0.12	0.0186	0.0322	0.0287	0.0210	0.0122
6	0.46	0.0463	0.0841	0.0498	0.0217	0.0199
7	0.48	0.0916	0.1538	0.0871	0.0368	0.0262
8	0.06	0.0093	0.0160	0.0172	0.0060	0.0030
9	0.21	0.1005	0.1359	0.0695	0.0262	0.0000
10	0.05	0.0101	0.0168	0.0073	0.0000	0.0000
11	0.06	0.0136	0.0221	0.0163	0.0130	0.0133
12	0.06	0.0093	0.0160	0.0057	0.0030	0.0000
13	0.21	0.0228	0.0412	0.0268	0.0149	0.0086
14	0.16	0.0586	0.0858	0.0991	0.0683	0.0636
15	0.11	0.0247	0.0403	0.0202	0.0086	0.0000
16	0.22	0.0537	0.0863	0.0957	0.0651	0.0333
17	0.24	0.0335	0.0588	0.0365	0.0109	0.0028
18	0.32	0.0480	0.0836	0.0391	0.0058	0.0000
19	0.10	0.0855	0.0922	0.0958	0.0146	0.0157
20	0.63	0.0832	0.1470	0.0743	0.0335	0.0209
21	0.08	0.0122	0.0211	0.0141	0.0089	0.0030
22	0.41	0.0689	0.1180	0.0682	0.0358	0.0165
23	0.18	0.0456	0.0727	0.0360	0.0145	0.0099
24	0.13	0.0289	0.0473	0.0200	0.0043	0.0043
25	0.38	0.0661	0.1126	0.0512	0.0403	0.0171
Average	0.218	0.042	0.0676	0.0423	0.0198	0.0116

Table 3: Precision, Recall, and F1-scores for each query

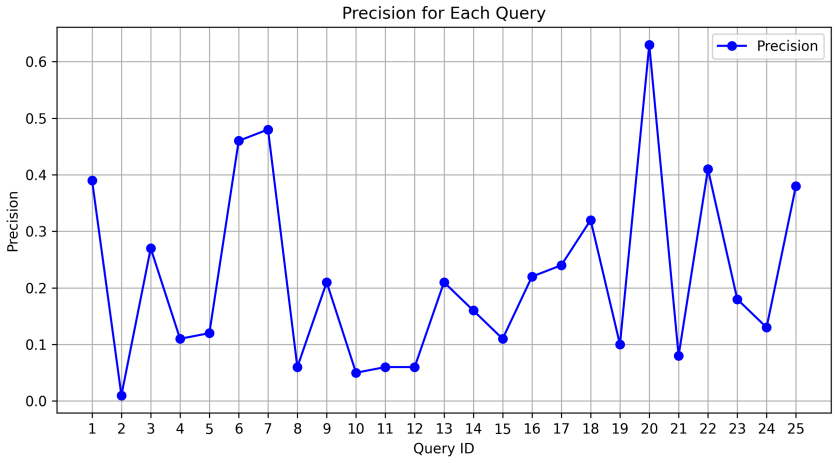

Figure 1: Fraction of relevant results among the top-100 (Precision)

3.2.2 BPE Tokenizer

Time taken: `94.36` seconds
Efficiency: `3.77` seconds

Query	Precision	Recall	F1@100	F1@50	F1@20	F1@10
1	0.15	0.0215	0.0375	0.0214	0.0056	0.0028
2	0.06	0.0179	0.0276	0.0000	0.0000	0.0000
3	0.31	0.0475	0.0824	0.0427	0.0357	0.0151
4	0.00	0.0000	0.0000	0.0000	0.0000	0.0000
5	0.43	0.0666	0.1153	0.0402	0.0210	0.0122
6	0.44	0.0443	0.0804	0.0498	0.0237	0.0199
7	0.46	0.0878	0.1474	0.0836	0.0404	0.0225
8	0.07	0.0108	0.0187	0.0172	0.0060	0.0061
9	0.00	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.04	0.0080	0.0134	0.0037	0.0000	0.0000
11	0.03	0.0068	0.0111	0.0000	0.0000	0.0000
12	0.02	0.0031	0.0053	0.0029	0.0000	0.0000
13	0.00	0.0000	0.0000	0.0000	0.0000	0.0000
14	0.14	0.0513	0.0751	0.0248	0.0068	0.0000
15	0.08	0.0179	0.0293	0.0242	0.0172	0.0044
16	0.09	0.0220	0.0353	0.0261	0.0140	0.0095
17	0.26	0.0363	0.0636	0.0339	0.0136	0.0083
18	0.25	0.0375	0.0653	0.0419	0.0204	0.0148
19	0.01	0.0085	0.0092	0.0120	0.0146	0.0000
20	0.48	0.0634	0.1120	0.0545	0.0206	0.0052
21	0.01	0.0015	0.0026	0.0028	0.0030	0.0030
22	0.06	0.0101	0.0173	0.0155	0.0000	0.0000
23	0.31	0.0785	0.1253	0.0719	0.0386	0.0099
24	0.18	0.0400	0.0655	0.0440	0.0128	0.0043
25	0.37	0.0643	0.1096	0.0736	0.0235	0.0171
Average	0.17	0.0298	0.05	0.0274	0.0127	0.0062

Table 4: Precision, Recall, and F1-scores for each query

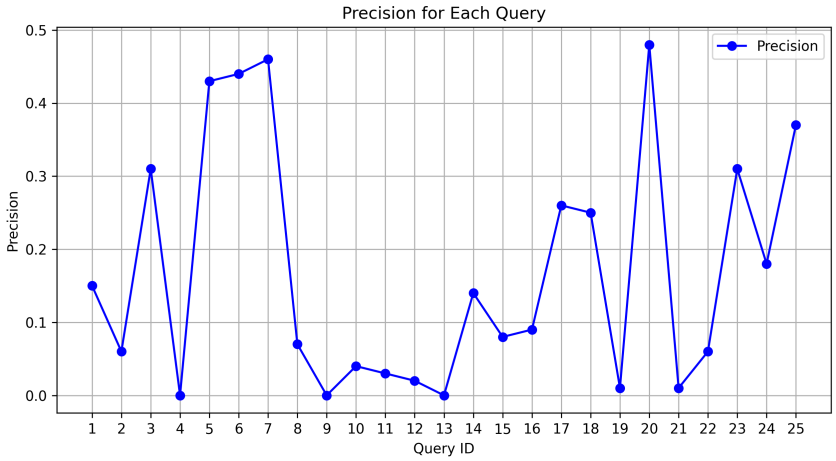

Figure 2: Fraction of relevant results among the top-100 (Precision)

3.2.3 WordPiece Tokenizer

Time taken: `209.68` seconds Efficiency: `8.38` seconds

Query	Precision	Recall	F1@100	F1@50	F1@20	F1@10
1	0.07	0.0100	0.0175	0.0080	0.0028	0.0000
2	0.00	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.07	0.0107	0.0186	0.0057	0.0000	0.0000
4	0.02	0.0035	0.0060	0.0065	0.0000	0.0000
5	0.00	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.05	0.0050	0.0091	0.0096	0.0099	0.0080
7	0.07	0.0134	0.0224	0.0209	0.0110	0.0075
8	0.09	0.0139	0.0241	0.0201	0.0120	0.0091
9	0.00	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.09	0.0181	0.0302	0.0183	0.0077	0.0039
11	0.02	0.0045	0.0074	0.0081	0.0000	0.0000
12	0.05	0.0077	0.0134	0.0029	0.0000	0.0000
13	0.04	0.0043	0.0078	0.0082	0.0021	0.0000
14	0.06	0.0220	0.0322	0.0310	0.0273	0.0212
15	0.01	0.0022	0.0037	0.0000	0.0000	0.0000
16	0.03	0.0073	0.0118	0.0087	0.0047	0.0048
17	0.04	0.0056	0.0098	0.0000	0.0000	0.0000
18	0.05	0.0075	0.0131	0.0112	0.0058	0.0000
19	0.01	0.0085	0.0092	0.0000	0.0000	0.0000
20	0.19	0.0251	0.0443	0.0297	0.0103	0.0078
21	0.02	0.0030	0.0053	0.0000	0.0000	0.0000
22	0.05	0.0084	0.0144	0.0062	0.0000	0.0000
23	0.05	0.0127	0.0202	0.0180	0.0193	0.0198
24	0.10	0.0222	0.0364	0.0360	0.0255	0.0130
25	0.01	0.0017	0.0030	0.0032	0.0034	0.0000
Average	0.0476	0.0087	0.0144	0.0101	0.0057	0.0038

Table 5: Precision, Recall, and F1-scores for each query

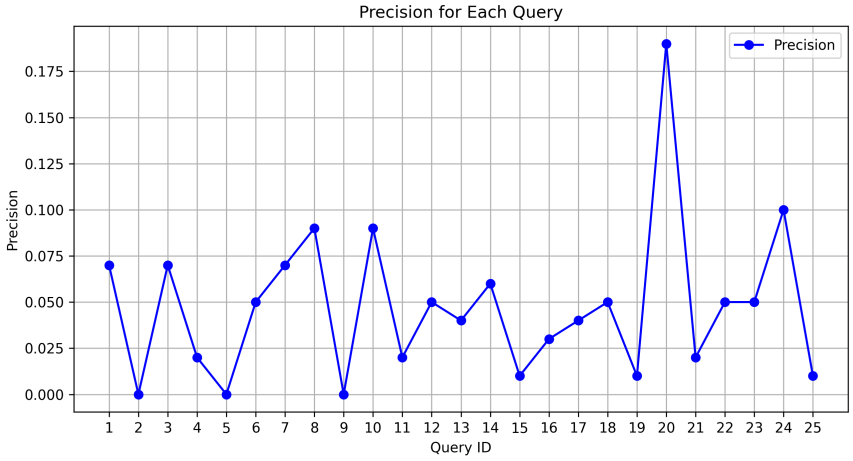

Figure 3: Fraction of relevant results among the top-100 (Precision)