Introdução

Este documento serve como documentação do trabalho feito para o Assingment1 da cadeira de Recuperação da Informação no ano letivo de 2022/2023.

Reader

O reader implementado traduz-se numa classe python com um dicionário (hash) para guardar a informação dos documentos a ser lidos (chave - doc ID : valor - titulo + texto do documento) bem como um byte counter que serve como ponteiro para permitir a leitura do documento não sequencial.

Foi utilizada a biblioteca gzip para ler os conteúdos dos ficheiros comprimidos. O ficheiro json é lido linha a linha e a informação sobre cada documento é introduzida no dicionário da classe.

Tokenizer

O tokenizer itera sobre o dicionário de documentos do reader e dá output duma token string que guarda cada um dos tokens como um tuplo (doc ID, token).

Através de uma expressão regular, o texto dos documentos é separado nos seus tokens. Posteriormente, dependendo do input do utilizador, os tokens são processados:

- tokens no ficheiro de stopwords são skipped;
- tokens com número de caracteres menor que aquele especificado pelo user são skipped;
- caso o utilizador especifique um stemmer, os tokens são stemmed (implementado pela biblioteca python NTLK);
- os caracteres dos tokens são forçados a lowercase.

Indexer

O processo de indexação SPIMI começa pela leitura dos documentos (através do reader), a transformação dos documentos lidos em tokens (através do tokenizer), e finalmente, com os a token stream obtida, é chamada a função SPIMI_Invert, que traduz a token stream (guardada em memória) em pequenos *inverted indexes* guardados em disco (ordenados alfabeticamente), libertando memória.

Este processo é repetido até a totalidade do ficheiro estar processada em pequenos inverted indexes guardados em disco. Passamos então para a segunda fase do SPIMI Invert: o merge destes blocos.

São lidos os X primeiros tokens de cada bloco para memória. Devido à prévia ordenação alfabética podemos selecionar o termo com precedência alfabética e dar merge aos postings de todos os blocos em que este é encontrado. Os postings de dito termo são então finais, e podem ser adicionados a um dicionário em memória. Quando o dicionário ocupa espaço suficiente em memória este é dumped para o nosso index final.

O index final é fracionado em vários ficheiros com um range alfabético indicado no nome do ficheiro (ex: alface-caneta.index).

Política de Memória

Na fase inicial de leitura-indexação-dump em blocos temporários, a memória facultada pelo utilizador é dividida entre os documentos lidos para memória e a token stream gerada pela tokenização.

Na fase de merge, um terço da memória é facultada ao dicionário do index final, e o restante repartido pelos buffers dos índices temporários guardados em disco.

Resultados

Os resultados foram obtidos a partir dos comandos encontrados no runSPIMI.sh. Foi utilizado um computador de 8Gb de RAM e

Resultados do tiny (python3 main.py indexer --indexer.class SPIMIIndexer --tk.stopwords stop_words_english.json --tk.minL 3 --indexer.memory_threshold 50 ./collections/pubmed_2022_tiny.jsonl.gz ./output/tiny)

```
--- Indexing Statistics ---
Total indexing time: 78.68223237991333
Merging time: 16.731722354888916
Number of temporary index segments written to disk: 1
Total index size on disk: 93351269
Vocabulary size: 200276
```

Resultados do small (python3 main.py indexer --indexer.class SPIMIIndexer --tk.stopwords stop_words_english.json --tk.minL 3 --indexer.memory_threshold 70 ./collections/pubmed_2022_small.jsonl.gz ./output/small)

```
Total indexing time: 1930.1399884223938
Merging time: 181.9301562309265
Number of temporary index segments written to disk: 8
Total index size on disk: 1044387846
Vocabulary size: 810292
```

Resultados do medium file (python3 main.py indexer --indexer.class SPIMIIndexer --tk.stopwords stop_words_english.json --tk.minL 3 --indexer.posting_threshold 200000 --indexer.memory_threshold 60 ./collections/pubmed_2022_medium.jsonl.gz ./output/medium)

```
--- Indexing Statistics ---
Total indexing time: 5552.725581645966
Merging time: 589.88658618927
Number of temporary index segments written to disk: 55
Total index size on disk: 3158125060
Vocabulary size: 1711538
```

Resultados do large (python3 main.py indexer --indexer.class SPIMIIndexer --tk.stopwords stop_words_english.json --tk.minL 3 --indexer.posting_threshold 230000 --indexer.memory_threshold 60 ./collections/pubmed_2022_large.jsonl.gz ./output/large)

Devido a insuficiência de espaço no disco e do elevado tempo de processamento, não foi possível recolher métricas para o ficheiro