2024-05-04 Information Retrieval - Boolean Retrieval

Submitter	Yusuf Syaifudin
Student ID Number	23/525770/PPA/06617
Date of submit	20 May 2024

Code

All code in this report can be obtained through public repository:
https://github.com/yusufsyaifudin/20240504-boolean-retrieval

System Architecture

This task is to create inverted index based on the document collections (term based). For this, I create two task: Indexing and Retrieval.

For Indexing task, I separate into two sub-task: data cleaning and create the index. For Retrieval task, I create function to the index model and try two query for each boolean logical operator: `AND`, `OR`, and `NOT`.

Create Inverted Index

The process is as follow:

1. Open file CSV.
2. For each row, I get the `content` value.
3. For each `content` value, I do case-folding (transform all characters into lower case).
4. Then I split by space to get `token` (or word).
5. Each `token` is streamed into functions: `remove_tweet_special`, `remove_number`, `remove_punctuation`, `remove_whitespace_multiple`, `remove_single_char` to clean unnecessary character. If after this process the `token` text become empty string, we skip this `token` for further process.
6. Stemmer library `Sastrawi==1.0.1` is used for stemming words.
7. Also, we remove some stopwords from `nltk.corpus` Indonesian dataset.

Step 6 and 7 is optional, hence I create 3 index:

Index name	Number of unique token (word)	Size	Description
word_map_not_stemmed_all_word.json	92778	34M	All words is saved as-is, without stemming. I still save into index even if it considered as stopwords.
word_map_stemmed_all_word.json	73573	31M	Save all stemmed words, even if it contains stop words.
word_map_stemmed_not_stopword.json	73256	20M	Words are stemmed, and if it is a stopword, then it will not be saved.

All this index is saved in following JSON format to make it easier to load, read (even by human), and re-use:

```
{
    "word1": [
        {
            "documentID1": 12
        },
        {
            "documentID2": 10
        }
    ]
}
```

The `12` is the number of occurences of word `word1` in the document with ID `documentID1`. All arrays is already sorted from high score to lowest before saving.

To build this index from 14343 documents, it took 1 hour 35 minutes 58 seconds on my Apple Macbook M1 Pro CPU 8-Core, GPU 14-Core, RAM 16 GB.

You can see this process in indexed.ipynb.

Retrieving The Documents From The Index

For retrieval, I do write my own logic for boolean logical operator.

1. `OR` operator is pretty simple, if `word` in the query match with the index key, then I push all document IDs. If the same document ID is found for the next query, I pick maximum score to be saved in the list.
2. `AND` operator will only save the document IDs that match of all words in the query.
3. `NOT` operator will look for any document that not contain all word in the query.

I create some testing data for this named `word_map_test.json`, here's the example:

```
{
    "a": [
        {"doc1": 3},
        {"doc2": 2},
    ],
    "b": [
        {"doc3": 10},
        {"doc1": 4},
    ],
    "c": [
        {"doc1": 5},
        {"doc4": 10},
    ],
}
```

Logical Operator	Query	Result	Notes
AND	['a', 'b']	[('doc1', 4)]	Because only `doc1` that have word `a` and `b`.
OR	['a', 'b']	[('doc3', 10), ('doc1', 4), ('doc2', 2)]	Because either `doc1`, `doc2`, and `doc3` that have word `a` OR `b`. `doc3` is took precedence because it has highest occurence among other documents.
NOT	['a', 'b']	[('doc4', 10)]	Only `doc4` that not contain both word `a` and `b`. `doc1` contain word `a` and `b`, whereas `doc2` contain word `a`, so these docs is not match with `NOT` query.

Analysis

Computation time

To create index, it tooks: 1 hour 35 minutes 58 seconds with average 0.401673 seconds per document.

When retrive data with query `["pemerintah", "korupsi"]`, the index `word_map_stemmed_all_word.json` took longest time compared to other index. This is because it need to do stemming before searching. It still take longer than `word_map_stemmed_not_stopword.json` because the index file is larger. Also, why `word_map_not_stemmed_all_word.json` take longer time even if I am not doing stemming, I guess that is because the index file is the largest than the others.

For query `["jalan", "rusak"]`, the index `word_map_not_stemmed_all_word.json` is took the longest time. Same as previous query, I guess that it is because the index file is large enough to do search. My assumption is supported by the fact that in the query `["pemerintah", "korupsi"]` is also takes longer time.

Also, in general we can see that `word_map_stemmed_all_word.json` is always longer than `word_map_stemmed_not_stopword.json` because it has combo: large index and need stemming.

The time to query and it's detail is described in table below:

Query: `["pemerintah", "korupsi"]`

Logical Operator	Index file	Total time (s)
AND	word_map_not_stemmed_all_word.json	1.037159
AND	word_map_stemmed_not_stopword.json	0.776101
AND	word_map_stemmed_all_word.json	**1.101779**
OR	word_map_not_stemmed_all_word.json	1.086989
OR	word_map_stemmed_not_stopword.json	0.759185
OR	word_map_stemmed_all_word.json	**1.113027**
NOT	word_map_not_stemmed_all_word.json	2.037013
NOT	word_map_stemmed_not_stopword.json	1.340236
NOT	word_map_stemmed_all_word.json	**2.057885**

word_map_not_stemmed_all_word.json

```
Query                   : pemerintah AND korupsi

Time to search          : 0.810764 seconds
Convert to list of dicts : 0.000014 seconds
Time to get 5 docs      : 0.226191 seconds
----------------------------------------
Total time taken        : 1.037159 seconds
```

```
================================================================================

Query                    : pemerintah OR korupsi

Time to search           : 0.856955 seconds
Convert to list of dicts : 0.000478 seconds
Time to get 5 docs       : 0.229308 seconds
----------------------------------------
Total time taken         : 1.086989 seconds

================================================================================

Query                    : pemerintah NOT korupsi

Time to search           : 1.807763 seconds
Convert to list of dicts : 0.001468 seconds
Time to get 5 docs       : 0.227465 seconds
----------------------------------------
Total time taken         : 2.037013 seconds
```

word_map_stemmed_not_stopword.json

```
Query                    : pemerintah AND korupsi

Time to stemming         : 0.073376 seconds => perintah AND korupsi
Time to search           : 0.476319 seconds
Convert to list of dicts : 0.000039 seconds
Time to get 5 docs       : 0.226156 seconds
----------------------------------------
Total time taken         : 0.776101 seconds

================================================================================

Query                    : pemerintah OR korupsi

Time to stemming         : 0.069313 seconds => perintah OR korupsi
Time to search           : 0.464579 seconds
Convert to list of dicts : 0.000485 seconds
Time to get 5 docs       : 0.224536 seconds
----------------------------------------
Total time taken         : 0.759185 seconds

================================================================================

Query                    : pemerintah NOT korupsi

Time to stemming         : 0.069633 seconds => perintah NOT korupsi
Time to search           : 1.034931 seconds
Convert to list of dicts : 0.001346 seconds
Time to get 5 docs       : 0.234101 seconds
```

```
----------------------------------------
Total time taken          : 1.340236 seconds
```

word_map_stemmed_all_word.json

```
Query                     : pemerintah AND korupsi

Time to stemming          : 0.081939 seconds => perintah AND korupsi
Time to search            : 0.782484 seconds
Convert to list of dicts  : 0.000022 seconds
Time to get 5 docs        : 0.237061 seconds
----------------------------------------
Total time taken          : 1.101779 seconds

================================================================================

Query                     : pemerintah OR korupsi

Time to stemming          : 0.073133 seconds => perintah OR korupsi
Time to search            : 0.814207 seconds
Convert to list of dicts  : 0.000531 seconds
Time to get 5 docs        : 0.224878 seconds
----------------------------------------
Total time taken          : 1.113027 seconds


================================================================================

Query                     : pemerintah NOT korupsi

Time to stemming          : 0.073198 seconds => perintah NOT korupsi
Time to search            : 1.745777 seconds
Convert to list of dicts  : 0.001597 seconds
Time to get 5 docs        : 0.237045 seconds
----------------------------------------
Total time taken          : 2.057885 seconds
```

Query: `["jalan", "rusak"]`

Logical Operator	Index file	Total time (s)
AND	word_map_not_stemmed_all_word.json	**1.039380**
AND	word_map_stemmed_not_stopword.json	0.702490
AND	word_map_stemmed_all_word.json	0.978623

Logical Operator	Index file	Total time (s)
OR	word_map_not_stemmed_all_word.json	**1.036240**
OR	word_map_stemmed_not_stopword.json	0.711646
OR	word_map_stemmed_all_word.json	0.984228
NOT	word_map_not_stemmed_all_word.json	**2.032055**
NOT	word_map_stemmed_not_stopword.json	1.380375
NOT	word_map_stemmed_all_word.json	1.923374

word_map_not_stemmed_all_word.json

```
Query                    : jalan AND rusak

Time to search           : 0.810571 seconds
Convert to list of dicts : 0.000007 seconds
Time to get 5 docs       : 0.228611 seconds
----------------------------------------
Total time taken         : 1.039380 seconds

================================================================================

Query                    : jalan OR rusak

Time to search           : 0.812092 seconds
Convert to list of dicts : 0.000226 seconds
Time to get 5 docs       : 0.223755 seconds
----------------------------------------
Total time taken         : 1.036240 seconds

================================================================================
Query                    : jalan NOT rusak

Time to search           : 1.787896 seconds
Convert to list of dicts : 0.002283 seconds
Time to get 5 docs       : 0.241436 seconds
----------------------------------------
Total time taken         : 2.032055 seconds
```

word_map_stemmed_not_stopword.json

```
Query                    : jalan AND rusak

Time to stemming         : 0.003485 seconds => jalan AND rusak
Time to search           : 0.469897 seconds
Convert to list of dicts : 0.000018 seconds
```

```
Time to get 5 docs       : 0.228920 seconds
----------------------------------------
Total time taken         : 0.702490 seconds

================================================================================

Query                    : jalan OR rusak

Time to stemming         : 0.003501 seconds => jalan OR rusak
Time to search           : 0.483797 seconds
Convert to list of dicts : 0.000503 seconds
Time to get 5 docs       : 0.223679 seconds
----------------------------------------
Total time taken         : 0.711646 seconds

================================================================================

Query                    : jalan NOT rusak

Time to stemming         : 0.003525 seconds => jalan NOT rusak
Time to search           : 1.139313 seconds
Convert to list of dicts : 0.001425 seconds
Time to get 5 docs       : 0.235887 seconds
----------------------------------------
Total time taken         : 1.380375 seconds
```

word_map_stemmed_all_word.json

```
Query                    : jalan AND rusak

Time to stemming         : 0.003571 seconds => jalan AND rusak
Time to search           : 0.737541 seconds
Convert to list of dicts : 0.000035 seconds
Time to get 5 docs       : 0.237249 seconds
----------------------------------------
Total time taken         : 0.978623 seconds

================================================================================

Query                    : jalan OR rusak

Time to stemming         : 0.003539 seconds => jalan OR rusak
Time to search           : 0.745485 seconds
Convert to list of dicts : 0.000633 seconds
Time to get 5 docs       : 0.234268 seconds
----------------------------------------
Total time taken         : 0.984228 seconds

================================================================================

Query                    : jalan NOT rusak
```

```
Time to stemming         : 0.003709 seconds => jalan NOT rusak
Time to search           : 1.688666 seconds
Convert to list of dicts : 0.001279 seconds
Time to get 5 docs       : 0.229430 seconds
----------------------------------------
Total time taken         : 1.923374 seconds
```

Are the retrieved documents relevant to the queries?

Since the query used in this trial is pretty simple, I can conclude:

- Using logical operator `AND`, both queries (`["pemerintah", "korupsi"]` and `["jalan", "rusak"]`) return good relevance of documents (using all index file). This is because the stemming index is consistently applied during indexing and retrieval process. Also, since `AND` query means that all `token` or `word` must exist in the same document, it is easily for inverted index to match the desired documents.

- `OR` queries return somewhat less relevant document. My guess is because it simply push any document with minimum one matches of `token` or `word` to the results array. Then it only sort the documents based on the `word` occurences.

- `NOT` query is clear and predictable to have most less relevant result. Actually we cannot say it is irrelevant, because the nature of the query is "we want a document that NOT contain all of the words in the array". So, since the returned document already return the documents that not contain any single word or token from the query list, then we can say that it is work well and can return results as expected.