

Classes

Main.py:

Initial Class, runs the index generator and the query module

Documents.py:

Uses beautiful soup module to parse the sgm folder and returns newid and bodies and text of documents (returns text because I realise some documents did not have a body, just <text> opening and closing tag). Sending results lowercased already

preProcessor.py:

Initialize set of stop words, set of numbers and symbols to be removed then proceeds to tokenize the bodies of the document, when tokenization and removal is done, postings list is created. If the postings list gets bigger than the memory size argument passed through the command line, writes a block of term-list(docId) as a json file

Merge.py:

Merges every json file as one big inverted index json file

Query.py:

Query class initialization with reading the index from the json file. In the main creation of Query object before asking for input such that class is loaded only once. Query class stores index in class attributes and then splits the query inputted terms if there are multiple terms.

After that it searches if the queryTerms are in the dictionary, if they are, appends the postingsList to an initialize postingsList initialized array. Query class returns intersection of postings lists

	dictionary			Non-positional postings		
unfiltered	493506	Delta	cml	1844224	Delta	cml
No numbers	365372	-25%	-25%	1690158	-8%	-8%
Stop words	341163	-6%	-30%	1256396	-25%	-31%
No symbols	337427	-1%	-31%	1101397	-12%	-40%

TEST QUERIES

Jimmy carter:

Results: [12136, 13540, 17023, 18005, 19432, 20614]

Green Party:

Results: [10230]

Innovation in Telecommunications:

Results: []

WHAT I'VE LEARN:

Never give up never surrenders. This was one of the most exciting projects I've done and I'm really proud to have not given after hours and hours of errors messages! Also improving my understanding of spimi and indexing in general