TDT4117 Information Retrieval - Autumn 2021

Task 1 - Page rank and HITS

- Compare page rank and HITS and briefly describe the main ideas of both approaches and point out their differences.
- Given the graph below, compute hub and authority scores for webpages labeled as A, B, C, D and E using HITS algorithm. Perform at least 3 iterations of the algorithm and illustrate your computations by providing formulas filled with values for at least one iteration

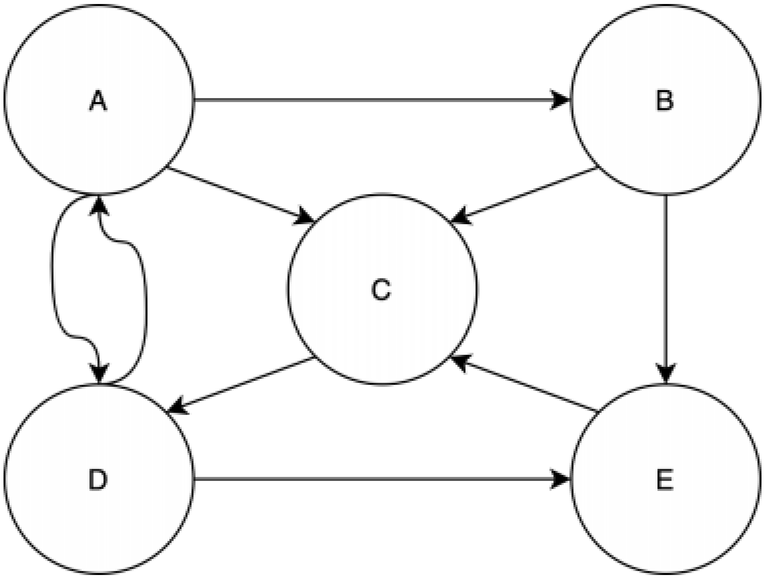


Figure 1: Graph of websites connected by links.

Answer:

Bjørn Are Odden
Ole Alexander Høyby

Task 2 - Structured Indexing and Retrieval in Lucene

Throughout this task, you will work with a subset of the 20 different newsgroups collection (original 19997 documents), which you can download from Blackboard ('20newspart.zip').

Luke

As you have maybe already noticed, is Lucene quite rich in possibilities regarding indexing of documents and searching. For example, Lucene offers the option of indexing a document in several fields (i.e., subject, body, from). If a collection is indexed, it becomes easier to search across fields or in specific fields only.
Lucene contains since version 8.1 a tool called Lucene Index Toolbox or Luke https://github.com/DmitryKey/luke. First, download the binary release https://www.apache.org/dyn/closer.lua/lucene/java/8.10.1/lucene-8.10.1.tgz and navigate into the sub-folder luke. You find two files to start Luke for Windows (.bat) and Unix/Linux (.sh) systems.

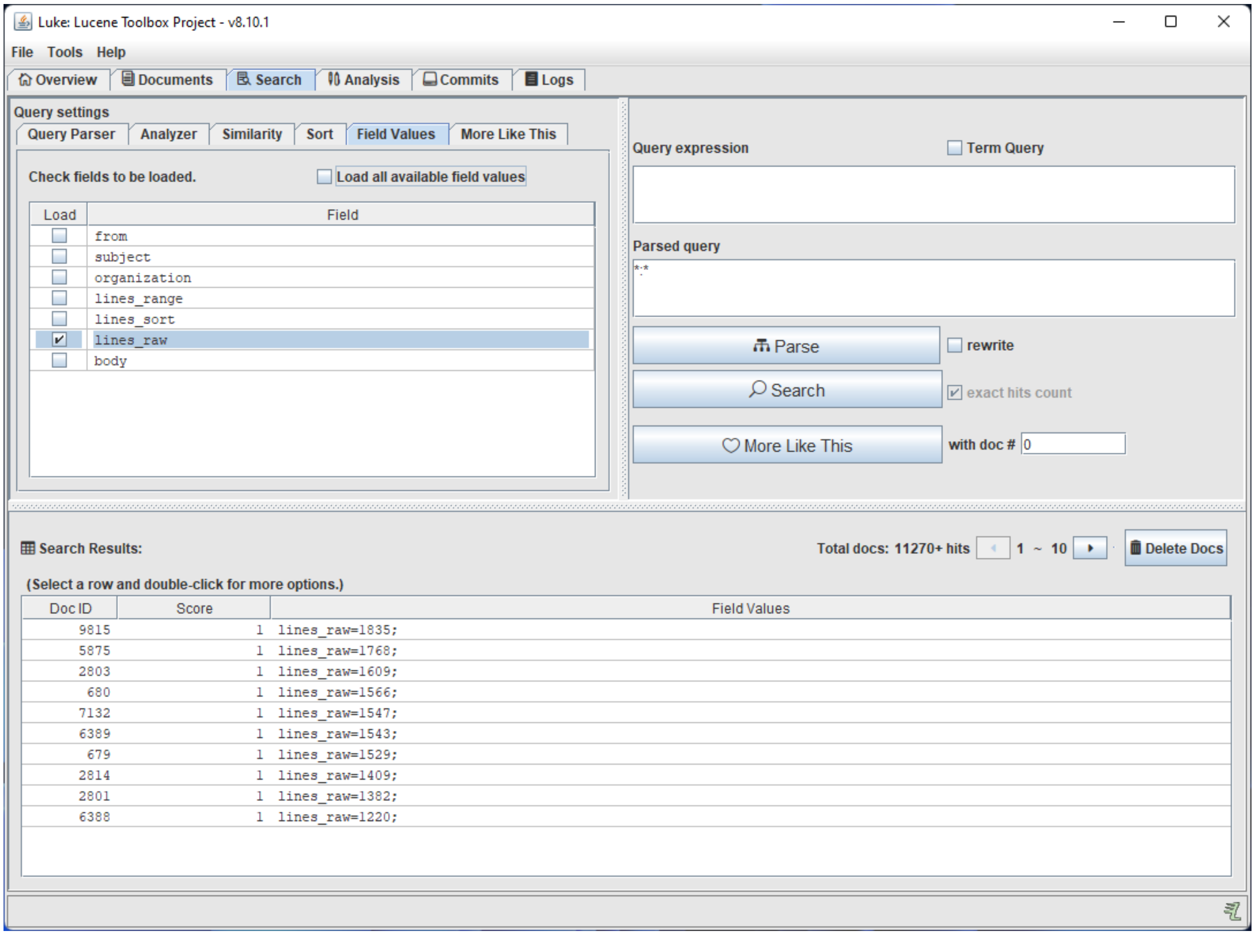
The task is to create an index based on the given collection of documents and perform an analysis.

Show screenshots of the results and explain the behavior of the system.

Subtask A

Once you start the application, create a new index. Choose a path to store the index, provide the path to the documents and let Luke create the index. Does Luke create a structure for different fields? Have a look at a document; do you miss any additional fields to search in or filter by that? If ’not’ why?

Answer:
At first glance, it seems like lucene automatically creates indexes over fields present in the documents on the form **“fieldname:fieldcontent”**, allowing one to search for terms in those fields.



Check fields to be loaded. □ Load all available field values

Load	Field
□	from
□	subject
□	organization
□	lines_range
□	lines_sort
☑	lines_raw
□	body

Parsed query: `*:*`

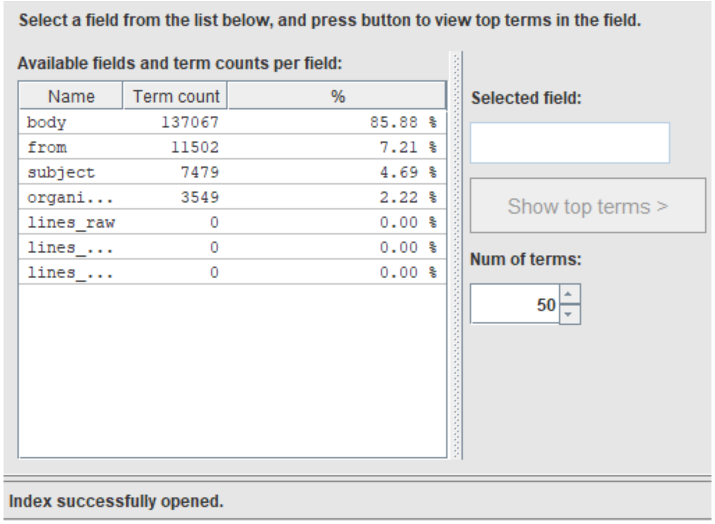
Total docs: 11270+ hits 1 ~ 10

(Select a row and double-click for more options.)

Doc ID	Score	Field Values
9815	1	lines_raw=1835;
5875	1	lines_raw=1768;
2803	1	lines_raw=1609;
680	1	lines_raw=1566;
7132	1	lines_raw=1547;
6389	1	lines_raw=1543;
679	1	lines_raw=1529;
2814	1	lines_raw=1409;
2801	1	lines_raw=1382;
6388	1	lines_raw=1220;

Fig. 1: Using the lines field to sort returned documents.

One exception is the “Lines” field, where Lucene doesn’t seem to parse the values as terms (or store it as a term/value, deduced by looking at fig 1 and 2, but still getting the correct result when applying the sort), but rather allows you to sort the documents returned according to the value.



Select a field from the list below, and press button to view top terms in the field.

Available fields and term counts per field:

Name	Term count	%
body	137067	85.88 %
from	11502	7.21 %
subject	7479	4.69 %
organi...	3549	2.22 %
lines_raw	0	0.00 %
lines_...	0	0.00 %
lines_...	0	0.00 %

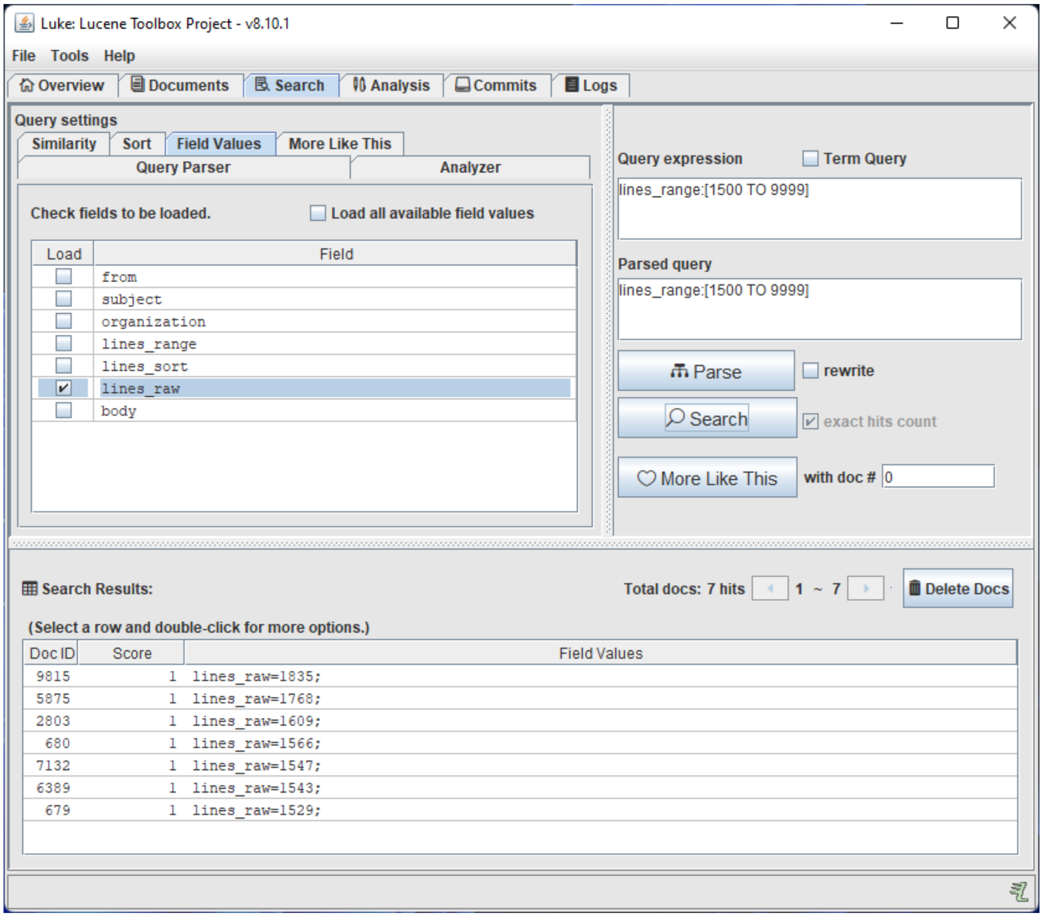
Selected field:

Num of terms: 50

Index successfully opened.

Fig 2: Screencap from the Luke's Overview tab

In addition to sorting, the "Lines" field also allows adding terms to the query to specify that the returned documents must have a value within a certain range for the field, as shown in fig 3.



Check fields to be loaded. □ Load all available field values

Load	Field
□	from
□	subject
□	organization
□	lines_range
□	lines_sort
☑	lines_raw
□	body

Query expression □ Term Query

lines_range:[1500 TO 9999]

Parsed query

lines_range:[1500 TO 9999]

□ rewrite

☑ exact hits count

with doc # 0

Search Results:

Total docs: 7 hits 1 ~ 7

(Select a row and double-click for more options.)

Doc ID	Score	Field Values
9815	1	lines_raw=1835;
5875	1	lines_raw=1768;
2803	1	lines_raw=1609;
680	1	lines_raw=1566;
7132	1	lines_raw=1547;
6389	1	lines_raw=1543;
679	1	lines_raw=1529;

Fig 3: Returning documents with over 1500 lines.

Further, by looking at some random documents, there seems to be other fields present in *some* documents which Lucene hasn't created fields to search by in the index, e.g. "Distribution:", "X-Newsreader:" and so on. Our guess is that only fields present in all files at index creation are set up and created to be searched by specifically, as we were not able to find documentation confirming this or another approach.

And last, there's no explicit "body"-field in the documents, so we assume body is all content that isn't part of another field/not declared explicitly.

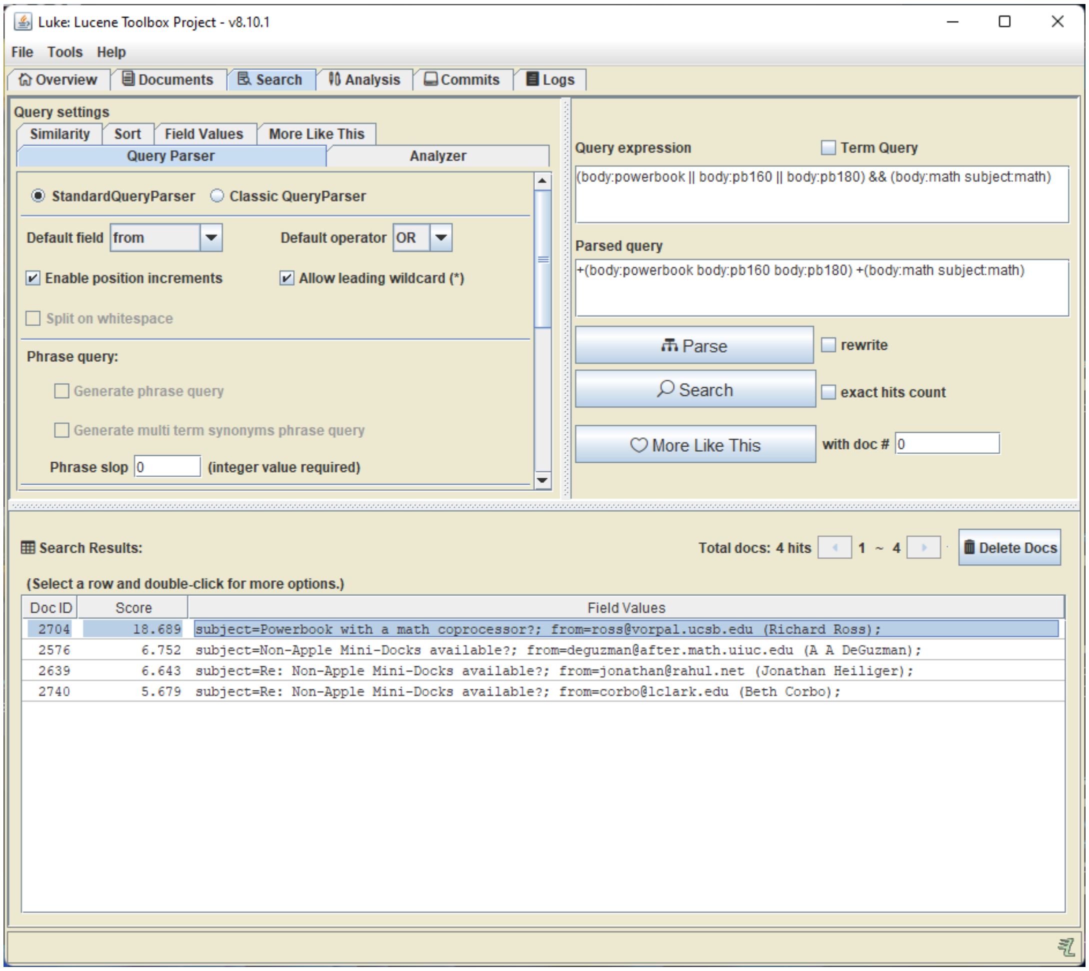
Subtask B

- Check if there is a document in the collection that someone from the University of California Santa Barbara sent. How many documents were sent? In the results, is there a specific subject that dominates the result?

Answer:
We found out that the domain name for UCSB is ucsb.edu and enabled the "allow leading wildcard" option in the Query Parser settings, which then allowed us to search for the term "from:*ucsb.edu". This results in 11 returned documents, where 6 of them have the subject: "Re: Amusing atheists and anarchists", which seem to include, according to one participant, hot takes about the Mongols and Stalin. The rest seem unrelated, but all involving hardware and/or software in some form.

- One of the documents from the University of California Santa Barbara is about a Powerbook. Why the need of a Powerbook and what is specific about that Powerbook? Search for other documents that contain issues regarding the need for a Powerbook and document the findings.

Answer:
The document in question is from ross@vorpal.ucsb.edu, who needs a Powerbook capable of running Mathematica, which apparently needs a math coprocessor (mcp) which can be found in the PB180 model, but not the PB160.



● StandardQueryParser ○ Classic QueryParser

Default field: from

Default operator: OR

☑ Enable position increments

☑ Allow leading wildcard (*)

□ Split on whitespace

Phrase query:

□ Generate phrase query

□ Generate multi term synonyms phrase query

Phrase slop: 0 (integer value required)

Query expression □ Term Query

```
(body:powerbook || body:pb160 || body:pb180) && (body:math subject:math)
```

Parsed query

```
+(body:powerbook body:pb160 body:pb180) +(body:math subject:math)
```

□ rewrite

□ exact hits count

with doc # 0

Search Results:

Total docs: 4 hits 1 ~ 4

(Select a row and double-click for more options.)

Doc ID	Score	Field Values
2704	18.689	subject=Powerbook with a math coprocessor?; from=ross@vorpal.ucsb.edu (Richard Ross);
2576	6.752	subject=Non-Apple Mini-Docks available?; from=deguzman@after.math.uiuc.edu (A A DeGuzman);
2639	6.643	subject=Re: Non-Apple Mini-Docks available?; from=jonathan@rahul.net (Jonathan Heiliger);
2740	5.679	subject=Re: Non-Apple Mini-Docks available?; from=corbo@lclark.edu (Beth Corbo);

Fig 4: settings and terms used for task 2b

Using some specific terms to search, we find more people, more specifically from deguzman@after.math.uiuc.edu, looking at computers with the same capability for running the software which requires an mcp for his boss. He receives a reply from jonathan@rahul.net , suggesting to look at E-Machines, a manufacturer that might have what he's looking for. In another followup, corbo@lclark.edu, provides some additional specs for the suggested computer.