Task 1 : Structured Indexing and Retrieval in Lucene

Subtask A

Here is the MyDocument.java-class implemented

```java
public class MyDocument{

	public static Document Document (File f) throws java.io.FileNotFoundException{

		// make a new, empty document
		Document doc = new Document();

		// use the news document wrapper
		NewsDocument newsDocument = new NewsDocument(f);

		//TODO: create structured Lucene document
		Field idField = new Field("id", newsDocument.getId(), Field.Store.YES, Field.Index.ANALYZED);
		Field fromField = new Field("from", newsDocument.getFrom(), Field.Store.YES, Field.Index.ANALYZED);
		Field subjectField = new Field("subject", newsDocument.getSubject(), Field.Store.YES, Field.Index.ANALYZED);
		Field contentsField = new Field("contents", newsDocument.getContent(), Field.Store.YES, Field.Index.ANALYZED);

		doc.add(idField);
		doc.add(fromField);
		doc.add(subjectField);
		doc.add(contentsField);

		return doc;
	}
}
```

Subtask B

After downloaded Luke and used it on the indexed collection from subtask A with the term 'Vancouver' we got the following result:

Content:

Results: (Hint: Double-click on results to display all fields) | Explain | 13 doc(s) | 0-12

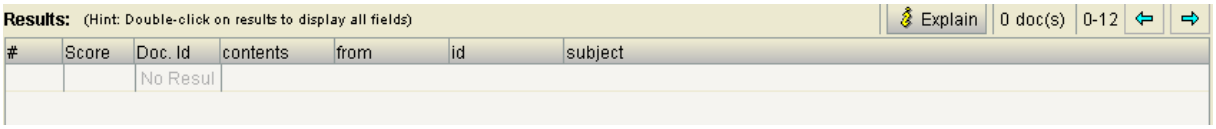
#	Score	Doc. Id	contents	from	id	subject
0	0,6534	1253	Nntp-Posting	gballent@var	54759.txt	Re: Winnipeg vs. Vancouver
1	0,5545	951	Organization:	advax@reg.tr	54277.txt	Re: How universal are (video) phones these days?
2	0,4620	160	Distribution: \	armani@edg	52114.txt	Re: Quadra 900/950
3	0,4620	624	Nntp-Posting	dwarf@bcarh	53890.txt	Re: #77's?
4	0,4620	1103	Organization:	<MWEINTR@	54545.txt	Playoff consecutive loss record?
5	0,4620	1314	Organization:	f_gautjw@cc:	54868.txt	Re: BD's did themselves--you're all paranoid freaks
6	0,3696	920	Organization:	bomr@erich.'	54246.txt	Re: multiple inputs for PC
7	0,3696	1213	Nntp-Posting	reiniger@ug.	54718.txt	Re: CBC: Canadian for ESPN.
8	0,3696	1256	Organization:	ragraca@vel:	54762.txt	Re: Wings will win
9	0,3234	1220	Organization:	kmcvay@one	54726.txt	Re: BD's did themselves--you're all paranoid freaks
10	0,2310	1071	Organization:	bks2@cbnew	54513.txt	NHL PLAYOFF RESULTS FOR GAMES PLAYED 4-21-93
11	0,1848	1747	Organization:	shell@cs.sfu	59478.txt	Great Canadian Scientists
12	0,0924	1558	Reply-To: dav	david@stat.c	59285.txt	HICN611 Medical News Part 4/4

Here every document that contains the query gets returned with a respective weight score calculated.

Luke passes the input string from the search text box to the QueryParser. Since we only inserted a single word in our input string, this is the search term Luke is using and the parser does not have to

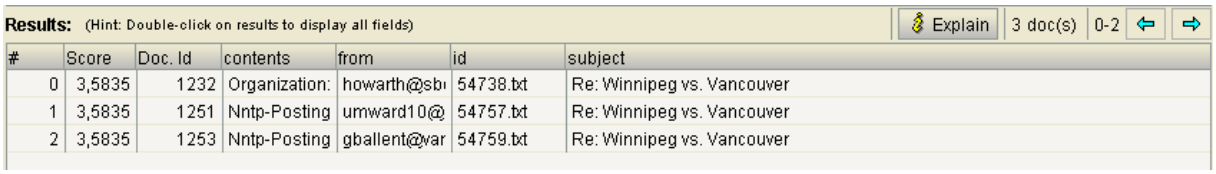
splits the search string into different search terms. The input is analyzed over the specified default field and the result of this search pulls back all documents, which contain the word Vancouver.

From and id:



None of the 'from' or 'id' fields consisted of the term 'Vancouver' and therefore none of the documents were returned.

Subject:



Here the collection where Vancouver vas in the subject gets returned. Since all of the documents containing the query have the exact same subject-description, all of the documents gets returned as their found with the same score.

Task 2 : Page rank and HITS

2.1

Compare page rank and HITS and briefly describe the main ideas of both approaches and point out their differences.

PageRank use a recursive scheme similar to HITS algorithm, but the PageRank algorithm produces a ranking independent of a user's query. The original idea behind PageRank is that a page i important if it is pointed to by other important pages. So the importance of your page is decided by your page's PageRank score, which is set by summoning the PageRank's of all pages that point to your page.

HITS was developed as an algorithm that made use of the link structure of the web in order to discover and rank pages relevant for a particular topic. The hyperlink-induced topic search algorithm was developed more by how humans analyze a search process rather than just machines searching for a topic and returning everything that matched. In example, if a human wants to a car and type in "top vehicle sellers in town", we expect to see results of the best cars and car dealers in town. However, if you search this using a query for a computer, the computer will simply count all the occurrences of the given word in a set of documents, not applying any intelligence in your search. The results will be similar of what we typed but not what we expected to them to be.

HITS is also known as hubs and authorities. A page is called an authority for a query if it contains valuable information on the topic and was linked there by many hubs and a hub if the information on

a page was not authoritative, but rather linked to many other pages. While PageRank assigns only one score to each page, HITS assigns two scores, the authority of the page that estimates the contents value, and the hub value that is estimates the value of the page's links to other pages.

The biggest strength of HITS is its ability to rank pages according to the query topic, resulting in relevant authority and hub pages. Some of the biggest weakness of HITS is that it does not detect advertisements, like sites that have commercial advertising sponsors that relates to your search. HITS can also fairly easily be spammed since people easily can add out-links on their own pages affecting the hub-score. PageRank is much more robust against spam since it's not easy for a webpage owner to add in-links to his/her page from other pages. The biggest disadvantage of PageRank is that it favors older pages since a new page, even if it is a very good page, will not have many links unless it is part of an already existing website. The PageRank can also easily be increased by the use of "link-farms", which is a group of sites that all link to every other site in the group, even if PageRank is actively looking for flaws like these while indexing.

So the main differences between PageRank and HITS are that HITS are query dependent, meaning the authority and hub scores are dependent on the search terms. HITS also sets two scores per document, while PageRank only has one, which helps HITS matching more relevant pages. HITS is also sensitive to user query, which PageRank is not. Although, PageRank is less susceptible to link spam, and PageRank is more efficient. PageRank also does computations at crawl time, while HITS does computations at query time, which makes PageRank more do-able in today's scenario.

2.2

The linking structure is defined by the following adjacency matrix *A*

$$A = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

The transposed matrix of *A* is defined as A^T

$$A^T = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Let h_i be the vector of hub scores after *i* iterations, initialized with all scores as 1

$$h_i = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Let a_i be the vector of authority scores after *i* iterations, initialized with all scores as 1

$$a_i = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

The hub and authority scores are updated according to the following algorithm[1];

1. *Start with each node having a hub score and authority score of 1*
2. *Run the Authority Update Rule*
3. *Run the Hub Update Rule*
4. *Normalize the values by dividing each Hub score by square root of the sum of the squares of all Hub scores, and dividing each Authority score by square root of the sum of the squares of all Authority scores.*
5. *Repeat from the second step as necessary.*

Where the *Authority Update Rule* is defined as:

For all p*:* $auth(p) = \sum_{i=1}^{n} hub(i)$*,* where *n* is the total number of pages connected to *p* and *i* is a page connected to *p.*

And the Hub Update Rule is defined as:
For all p, $hub(p) = \sum_{i=1}^{n} auth(i)$*,* where *n* is the total number of pages *p* connects to and *i* is a page which *p* connects to.

We can then rewrite the Authority Update Rule and Hub Update Rule to comply with our matrix notation as follows;

$$a_i = A^T h_{i-1}, \text{ and}$$

$$h_i = A a_i$$

First iteration:

We start by applying our matrix notation Authority Update Rule

$$a_1 = A^T h_0 \rightarrow \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 3 \\ 1 \end{pmatrix}$$

Next, we apply our matrix notation Hub Update Rule

$$h_1 = A a_1 \rightarrow \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \\ 1 \\ 4 \end{pmatrix}$$

[1] HITS algorithm on Wikipedia: https://en.wikipedia.org/wiki/HITS_algorithm

We then normalize a_1 by finding the sum of squares divided by the square root of the sum of squares, and dividing each element in the vector of authority scores by this number

$$\sqrt{(1^2 + 1^2 + 3^2 + 1^2)} \approx 3.464$$

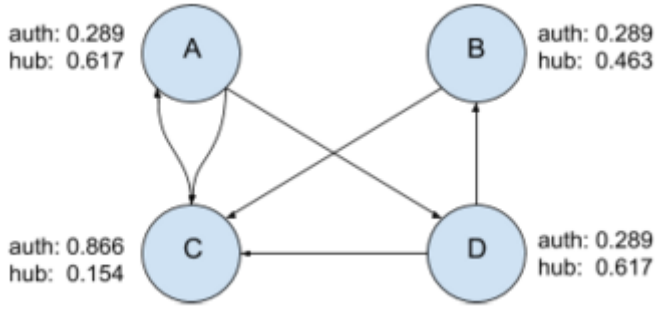
$$normalized\ a_1 = \begin{pmatrix} 0.289 \\ 0.289 \\ 0.866 \\ 0.289 \end{pmatrix}$$

Next, we normalize h_1 by applying the same procedures as above

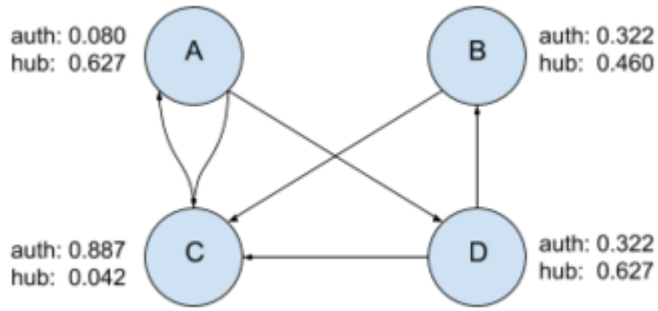
$$\sqrt{(4^2 + 3^2 + 1^2 + 4^2} \approx 6.481$$

$$normalized\ h_1 = \begin{pmatrix} 0.617 \\ 0.463 \\ 0.154 \\ 0.617 \end{pmatrix}$$

After the first iteration of the HITS algorithm, we have the following graph (with hub and authority scores shown)



Second iteration:



Third iteration: