

Laboratory exercise 1

in Internet Search Techniques and Business Intelligence: Index and search news with Lucene search engine

In this laboratory exercise we will use Lucene, which is a text search engine, in which you can index documents and search for words and phrases. It is Java-based and free to download. In this exercise we will use Lucene's demo application, for indexing and searching.

We will use press releases from the Swedish government, written in English, as the texts to index and to search in. To download a website or a part of a website you can use the tool wget.

In the exercise we will download pages from the website, use Lucene to index the pages, and then search the pages. Thereafter we will apply stemming on the texts and see if there are any differences in the search results after that.

Follow the steps below:

1. Download and extract exercise:

Download the zip-file lab.zip from

<http://people.dsv.su.se/~mariask/lab.zip>

and extract it in a root directory, for example M:\

(To extract,

either right-click and choose 7-zip, extract files

or right-click and choose Open with 7-zip. Choose "packa upp".)

Important: Choose extract to: M:\, or another root directory, or the paths will not work correctly.

2. Open a command prompt:

Open a command window through double-clicking "Command Prompt", which you find in the downloaded directory "lab"

(You need to know the following about the command prompt:

* In order to paste text into the command window, click on the little icon in the upper left corner and choose Edit, and then paste.

* In order to move downwards in the directory tree type:

`cd directoryName`, (where `directoryName` is changed to the name of the directory you want to move down in).

* In order to move upwards in the directory tree type:

`cd ..`

* Use the up arrow and down arrow on the keyboard to retrieve previous commands so you don't need to retype everything.)

3. Set environment variables:

In the command window, type `set_variables`. This will run a script that sets the path and the java class path.

4. Download the website:

Type `cd download`, which will place you in the directory **download**.

Type (in one line):

```
wget -rE -l3 -I/sb/d/586/a/ -w1 --timeout=30
http://www.sweden.gov.se/sb/d/586/a/
```

(-l3 contains the letter "l")

to download the website This will place the website in the directory
download.

Several hundred files will be downloaded, and you will see that on your
terminal (if just one file is downloaded you have probably not typed correctly)

5. Prepare for removing HTML tags:

Copy all the html-files from the directory **download\www.sweden.gov.se\sb\d**
586a to the directory **files_to_extract**. This can for example be done like this:

Go to the directory **files_to_extract** and type:

```
copy ..\download\www.sweden.gov.se\sb\d\586\a\*.html .
(Don't forget the final dot, which means "current directory".)
```

6. Remove the HTML tags:

Standing in the same directory, **files_to_extract**, type:

```
perl ..\text-extractor-isbi.pl (This is a script that
removes the HTML tags, and places them in files with the same name, but
ending with .txt). Open one of the downloaded txt-documents with a text-editor, for
example Notepad++. Have all HTML-tags been removed? What is the disadvantage
of removing HTML-tags before indexing the pages?
```

7. Prepare for indexing, through copying the txt-files to a separate directory.

Copy all the text files from that directory to the directory **files_to_index**. This
can for example be done like this:

Go to the directory **files_to_index** and type:

```
copy ..\files_to_extract\*.txt .
(Don't forget the final dot)
```

8. Index files

Stand in the directory **files_to_index** and type:

```
java org.apache.lucene.demo.IndexFiles .
```

(Don't forget the final dot) This will index the documents.

A directory named index has been created, in which the index information is
stored.

9. Search files:

Stand in the directory **files_to_index** and type:

```
java org.apache.lucene.demo.SearchFiles
```

in order to launch a command line searcher.

a) Search for the word *minister*. How many documents are found?

b) Search for the word *opportunity*. How many documents are found?

c) Search for the word *activity*. How many documents are found?

d) Search for the phrase "*will be published*". What is Lucene really searching
for? What do you call the kind of words that are removed in the search? Are
there any disadvantages using this technique?

10. Prepare for stemming

Standing in the directory **lemmatized_files_to_index**, type:

```
copy ..\files_to_extract\*.txt .
```

11. Apply stemming on the texts

Standing in the directory **lemmatized_files_to_index**, type:

```
perl ..\lemmatizer.pl ..\cstlemma\flexrules_en
```

to apply stemming. This will generate new files with file names ending with `.lemma`. Open some of the `.lemma`-files and compare them to the corresponding `.txt` file. Can you find some examples of what the script has done with the words?

12. Delete all files in the directory `lemmatized_files_to_index`, that ends with `.txt` , since it is only the lemmatized files that are going to be indexed.

Standing in the directory **lemmatized_files_to_index**, type:

```
del *.txt
```

13. Index `.lemma` files

Standing in the directory **lemmatized_files_to_index**, type:

```
java org.apache.lucene.demo.IndexFiles .
```

14. Search `.lemma` files:

Standing in the directory **lemmatized_files_to_index**, type:

```
java org.apache.lucene.demo.SearchFiles
```

Search for the words *minister*, *opportunity* and *activity* again.

How many documents are found? Does it differ from the number of documents that are found, when searching the original documents where stemming has not been applied?

Why or why not?

Since it is just a demo, it is not very robust, therefore when searching you sometimes will get an error like: "Exception in"

If you do, just restart the search demo. To quit the demo press "Enter" when you are prompted to search, or press ctrl-C.

Written statement

Write one A4 page of what you did and your observations and reflections during the steps 1-14 in the laboratory. Questions about the laboration can be asked to the laboratory assistant during the laboration. Remember that a written statement is compulsory to be approved of the laboration.

Extra

(only if you have more time)

Try to use Lucene HTML-indexer. Standing in

```
\lab\download\www.sweden.gov.se\sb\d\586\a
```

Type:

```
java org.apache.lucene.demo.IndexHTML -create .
```

and then

```
java org.apache.lucene.demo.SearchFiles
```

Search for the word *meeting*. What are the top ten results?

Search for the word *meeting* in the indexing without HTML tags (See step 8. and 9. above.) Is it the same top ten? If it's different, do you have a suggestion to why it is different?

Do also try to search for *title: meeting*

What documents are found?

More information

###lucene documentation

#http://lucene.apache.org/java/2_9_0/

###lucene documentation

#http://lucene.apache.org/java/2_9_0/

reference: basic set of useful wget switches

-r : Recursive retrieval, i.e. follow links to a specified depth (default 5)

-k : Convert links to absolute links for local browsing

-E : Force html extension to text/html files

-w : Number of seconds to wait between requests

-l : Number of levels to retrieve (i.e. the depth to go to)

-l : Stay within the given folder path

-nv : no verbose, a little less output to the terminal window

#

and the one you should be *_very_* careful with!

-e robots=off : Ignores robot ethics, think before you use this option

#

for more switches see <http://www.gnu.org/software/wget/manual/>

wget.html

Latest change: 2009-11-09 by Maria Skeppstedt