Data Management

Laboratory 01

- Students: *Romain Claret & Simon Martinez*
- Professor: *Dr. Fatemeh Borran*
- Assistant: *Gary Marigliano*

- Due-date: *Monday 15 October 2018*

Understanding the Lucene API

1. Yes, the demo uses the default stopword removal
   - QueryParser takes as argument a StandardAnalyzer() which is built by default with a default list of stopwords `STOP_WORDS_SET`
   - Proof: “frame” and “the frame” is giving the same output
2. No, the demo is not using any form of stemming
   - StandardAnalyzer doesn’t provide a stemming by default
   - We couldn’t find any stemming library into demo code
   - We didn’t find any custom/manual stemming in the demo code
   - Proof: “frame” and “frames” doesn’t give the same output
3. Yes, the demo is case insensitive
   - StandarAnalyzer use LowerCaseFilter which normalizes tokens text into lower case format.
   - Proof : “test” and “TEST” give the same output
4. Yes, it does matter
   - In case of Normalisation: taking the words “been” and “being” as example, the normalization of those words is “be”, which is part of the `STOP_WORDS_SET`. We could lose information if the stemming is done before the normalization.
   - Depends: If the stopwords are stemmed then we should stem first then apply the stopwords filter. Otherwise, we would do the inverse.

Indexing

Based on the FieldType documentation
- `fieldType.setStoreTermVectorOffsets(true);`
- Store token character offsets into the term vector for this field.
- `fieldType.setStoreTermVectorPayloads(true);`
- Store token payloads into the term vector for this field.
- `fieldType.setStoreTermVectorPositions(true);`

- Store token positions into the term vector for this field.
- `fieldType.setStoreTermVectors(true);`
- Store the indexed form into term vectors for this field.

Using different Analyzers

- **StandardAnalyzer**
  - Number of documents: 3203
    Number of terms: 27099
  -

Name	Term count	%	Decoder
summary	19.972	73,7 %	string utf8

  -

Rank	Freq	Field	Text
1	961	title	algorithm
2	657	summary	which
3	389	summary	system
4	378	summary	paper
5	375	summary	computer
6	352	summary	can
7	327	summary	described
8	323	summary	given
9	316	summary	presented
10	309	summary	time

  - Total real size of files in selected commits (or all): 2423 kB
  - Indexing time: 1294ms
- **WhitespaceAnalyzer**
  - Number of documents: 3203
    Number of terms: 34827
  -

Name	Term count	%	Decoder
summary	26.821	77,01 %	string utf8

Rank	Freq	Field	Text
1	1492	summary	of
2	1471	summary	the
3	1298	summary	is
4	1270	summary	a
5	1260	summary	and
6	1199	summary	to
7	1093	title	of
8	1077	summary	in
9	1042	summary	for
10	938	summary	The
- Total real size of files in selected commits (or all): 2870 kB
- Indexing time: 1453ms

- **EnglishAnalyzer**
  - Number of documents: 3203
    Number of terms: 23010
Name	Term count	%	Decoder
summary	16.724	72,68 %	string utf8
Rank	Freq	Field	Text
1	1008	title	algorithm
2	676	summary	us
3	657	summary	which
4	554	summary	comput
5	529	summary	program
6	511	summary	system
7	428	summary	present
8	414	summary	describ
9	392	title	comput
10	384	summary	paper
  - Total real size of files in selected commits (or all): 2319 kB
  - Indexing time: 1386ms
- **ShingleAnalyzerWrapper (shingle size 2)**

- Number of documents: 3203
  Number of terms: 103070
Name	Term count	%	Decoder
summary	85.610	83,06 %	string utf8
Rank	Freq	Field	Text
1	961	title	algorithm
2	657	summary	which
3	389	summary	system
4	378	summary	paper
5	375	summary	computer
6	352	summary	can
7	337	summary	_ paper
8	327	summary	described
9	323	summary	given
10	316	summary	presented
- Total real size of files in selected commits (or all): 5380 kB
- Indexing time: 2181ms

- **ShingleAnalyzerWrapper (shingle size 3)**
  - Number of documents: 3203
    Number of terms: 221842
Name	Term count	%	Decoder
summary	191.471	86,31 %	string utf8

-
Rank	Freq	Field	Text
1	961	title	algorithm
2	657	summary	which
3	389	summary	system
4	378	summary	paper
5	375	summary	computer
6	352	summary	can
7	337	summary	_ paper
8	327	summary	described
9	323	summary	given
10	316	summary	presented
    - Total real size of files in selected commits (or all): 9563 kB
    - Indexing time: 2863ms
- **StopAnalyzer**
    - Number of documents: 3203
      Number of terms: 24663
Name	Term count	%	Decoder
summary	18.342	74,37 %	string utf8
Rank	Freq	Field	Text
1	963	title	algorithm
2	396	summary	system
3	383	summary	computer
4	381	summary	paper
5	334	summary	presented
6	314	summary	time
7	280	summary	method
8	277	summary	program
9	276	summary	data
10	260	title	computer
    - Total real size of files in selected commits (or all): 2213 kB
    - Indexing time: 1188ms

Reading Index

1. What is the author with the highest number of publications? How many publications does he/she have?
    - ```
      Top ranking terms for field [authors] are:
      38 Thacher Jr., H. C.
      19 Naur, P.
      16 Hill, I. D.
      ```
2. List the top 10 terms in the title field with their frequency.
    - ```
      Top ranking terms for field [title] are:
      963 algorithm
      260 computer
      172 system
      154 programming
      125 method
      110 data
      108 systems
      99 language
      93 program
      78 time
      ```

Searching

- the term "Information Retrieval"
    - Searching for **[Information Retrieval]**
    - Total Results : **188**
    - 1456 : Data Manipulation and Programming Problemsin Automatic Information Retrieval (8,651913)
    - 890 : Everyman's Information Retrieval System (8,181953)
    - 1698 : Experimental Evaluation of InformationRetrieval Through a Teletypewriter (7,574708)
    - 2306 : Dynamic Document Processing (7,358763)
    - 3133 : The Use of Normal Multiplication Tablesfor Information Storage and Retrieval (7,355752)
    - 1031 : Theoretical Considerations in Information Retrieval Systems (7,312654)
    - 1934 : Randomized Binary Search Technique (7,106321)
    - 1680 : Easy English,a Language for InformationRetrieval Through a Remote Typewriter Console (6,702070)
    - 2989 : Effective Information Retrieval Using Term Accuracy (6,702070)
    - 2518 : On the Problem of Communicating Complex Information (6,249776)
- both "Information" and "Retrieval"

- Searching for **[Information AND Retrieval]**
- Total Results : **23**
- 1456 : Data Manipulation and Programming Problemsin Automatic Information Retrieval (8,651913)
- 890 : Everyman's Information Retrieval System (8,181953)
- 1698 : Experimental Evaluation of InformationRetrieval Through a Teletypewriter (7,574708)
- 2306 : Dynamic Document Processing (7,358763)
- 3133 : The Use of Normal Multiplication Tablesfor Information Storage and Retrieval (7,355752)
- 1031 : Theoretical Considerations in Information Retrieval Systems (7,312654)
- 1934 : Randomized Binary Search Technique (7,106321)
- 1680 : Easy English,a Language for InformationRetrieval Through a Remote Typewriter Console (6,702070)
- 2989 : Effective Information Retrieval Using Term Accuracy (6,702070)
- 2518 : On the Problem of Communicating Complex Information (6,249776)

* at least the term "Retrieval" and, possibly "Information" but not "Database"
  - Searching for **[+Retrieval Information -Database]**
  - Total Results : **54**
  - 1456 : Data Manipulation and Programming Problemsin Automatic Information Retrieval (8,651913)
  - 890 : Everyman's Information Retrieval System (8,181953)
  - 1698 : Experimental Evaluation of InformationRetrieval Through a Teletypewriter (7,574708)
  - 2306 : Dynamic Document Processing (7,358763)
  - 3133 : The Use of Normal Multiplication Tablesfor Information Storage and Retrieval (7,355752)
  - 1031 : Theoretical Considerations in Information Retrieval Systems (7,312654)
  - 1934 : Randomized Binary Search Technique (7,106321)
  - 1680 : Easy English,a Language for InformationRetrieval Through a Remote Typewriter Console (6,702070)
  - 2989 : Effective Information Retrieval Using Term Accuracy (6,702070)
  - 2518 : On the Problem of Communicating Complex Information (6,249776)
* starting with "Info"
  - Searching for **[Info*]**
  - Total Results : **193**
  - 221 : Coding Isomorphisms (1,000000)
  - 271 : A Storage Allocation Scheme for ALGOL 60 (1,000000)
  - 395 : Automation of Program Debugging (1,000000)
  - 396 : A Card Format for Reference Files in Information Processing (1,000000)
  - 408 : CL-1, An Environment for a Compiler (1,000000)
  - 439 : Record Linkage (1,000000)
  - 482 : On the Nonexistence of a Phrase Structure Grammar for ALGOL 60 (1,000000)
  - 615 : An Information Algebra - Phase I Report-LanguageStructure Group of the CODASYL Development Committee (1,000000)
  - 643 : A String Language for Symbol Manipulation Based on ALGOL 60 (1,000000)
  - 654 : COMIT as an IR Language (1,000000)
  - Searching for **[Information Retrieval~5]**

- Total Results : **191**
- 1456 : Data Manipulation and Programming Problemsin Automatic Information Retrieval (7,194777)
- 890 : Everyman's Information Retrieval System (6,724819)
- 2831 : Faster Retrieval from Context Trees (Corrigendum) (6,458653)
- 3133 : The Use of Normal Multiplication Tablesfor Information Storage and Retrieval (6,334777)
- 1698 : Experimental Evaluation of InformationRetrieval Through a Teletypewriter (6,215545)
- 2306 : Dynamic Document Processing (6,202866)
- 1031 : Theoretical Considerations in Information Retrieval Systems (6,081074)
- 1934 : Randomized Binary Search Technique (5,909492)
- 2159 : Canonical Structure in Attribute Based File Organization (5,628992)
- 2518 : On the Problem of Communicating Complex Information (5,344692)

```
System.out.println("Searching for [" + q +"]");
ComplexPhraseQueryParser parser = new ComplexPhraseQueryParser("summary",this.analyzer);
try {
   Query query = parser.parse(q);
   try {
              TopDocs search = this.indexSearcher.search(query, 10);
              System.out.printf("Total Results : %d\n",search.totalHits);
              for (int i=0; i< 10 ; i++){
                  System.out.printf("%s : %s (%f)\n",search.scoreDocs[i].doc,this.indexReader.document(search.scoreDocs[i].doc).get("title"),search.scoreDocs[i].score);
              }
   }
   catch (IOException ex_search){
      System.out.println(ex_search);
   }
}
catch (ParseException ex_parse){
   System.out.println(ex_parse);
}
```

Tuning the Lucene Score

- ClassicSimilarity

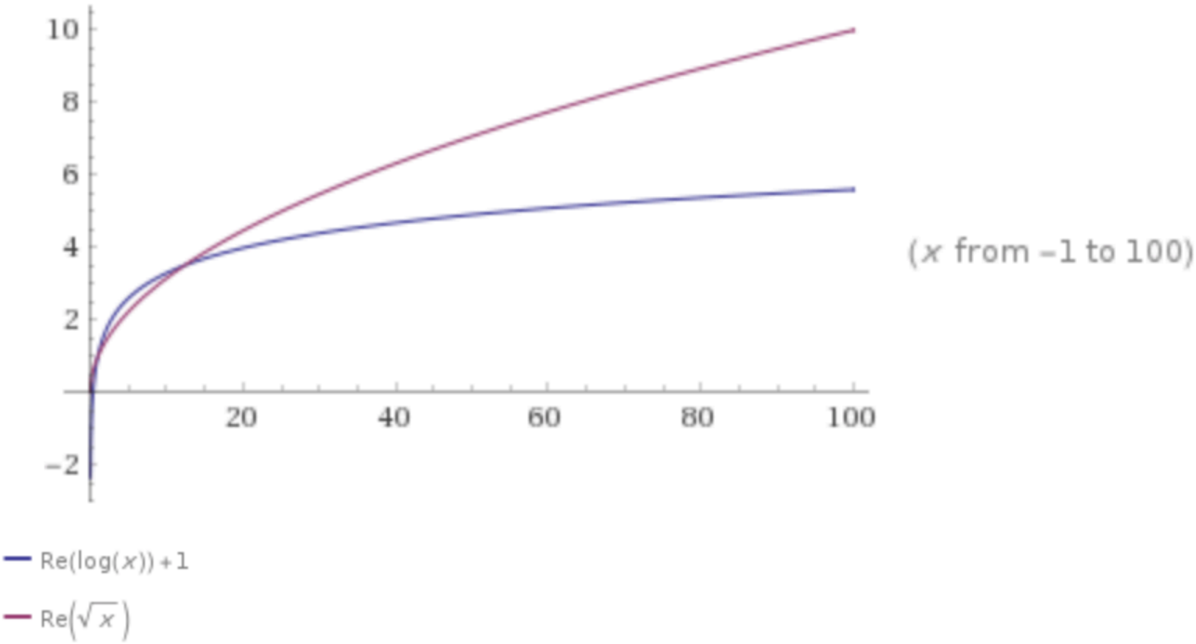
```
Searching for [compiler program]
Total Results : 578
3188 : An Algebraic Compiler for the FORTRAN Assembly Program (1,036766)
1458 : Requirements for Real-Time Languages (0,942752)
2651 : Reduction of Compilation Costs Through Language Contraction (0,937798)
1182 : A Note on the Use of a Digital Computerfor Doing Tedious Algebra and Programming (0,879980)
1464 : Program Translation Viewed as a General Data Processing Problem (0,829413)
1987 : A Formalism for Translator Interactions (0,829413)
1646 : WATFOR-The University of Waterloo FORTRAN IV Compiler (0,808241)
1236 : Conversion of Decision Tables To Computer Programs (0,754201)
2943 : Shifting Garbage Collection Overhead to Compile Time (0,754201)
636 : A NELIAC-Generated 7090-1401 Compiler (0,743363)
```

- MySimilarity

```
Searching for [compiler program]
Total Results : 578
2533 : Design and Implementation of a Diagnostic Compiler for PL/I (9,992505)
636 : A NELIAC-Generated 7090-1401 Compiler (9,239830)
2922 : High-Level Data Flow Analysis (8,913862)
2651 : Reduction of Compilation Costs Through Language Contraction (8,752361)
1646 : WATFOR-The University of Waterloo FORTRAN IV Compiler (8,554615)
1464 : Program Translation Viewed as a General Data Processing Problem (7,781996)
1987 : A Formalism for Translator Interactions (7,781996)
3188 : An Algebraic Compiler for the FORTRAN Assembly Program (7,781996)
1134 : A General Business-Oriented Language Based on Decision Expressions* (6,901810)
1236 : Conversion of Decision Tables To Computer Programs (6,901810)
```

- Describe the effect of using the new parameters.

  Plot:



The effect comes from the different between the tf=$(freq)^{1}/2$ from ClassicSimilarity and our custom tf=log(freq)+1.

The graph shows that MySimilarity is normilizing the documents frequency based on the overall term frequency. While the ClassicSimilarity is favorising documents with higher term frequency.

```java
//tf : 1+log(freq)
@Override
public float tf(float freq){
    return (float) (1 + Math.log(freq));
}

//idf : log(numDocs/docFreq+1)+1
@Override
public float idf(long docFreq, long numDocs){
    return (float) (Math.log(numDocs / (docFreq + 1)) + 1);
}

//lengthNorm : 1
@Override
```

```java
public float lengthNorm(FieldInvertState fieldInvertState) {
    return (float) 1;
}

//coord : squart(overlap/maxOverlap)
@Override
public float coord(int overlap, int maxOverlap){
    return (float) Math.sqrt(overlap / maxOverlap);
}
```