Information Retrieval

Results for Assignment 3: Boolean Information Retrieval with Lucene

Patrick Schäfer (patrick.schaefer@hu-berlin.de)

Marc Bux (buxmarcn@informatik.hu-berlin.de)

Approach, Effort Invested (Monday)

Group	Index	Effort (h)
BigGoogle	Lucene, StandardAnalyzer	2
Devin&Johannes	Lucene, RAMDisk, StandardAnalyzer	7-8
getSchwifty()	Lucene, RAMDisk, StandardAnalyzer	5
Hendrik&Sinje	Lucene, StandardAnalyzer	7-8
ilGruppo		
InfoRet2000	Lucene, StandardAnalyzer	4-5
jp&bl		
Julian&Dennis		
lc&mk		
NewDimensions	Lucene, RAMDIsk, StandardAnalyzer, Multi-Threading fürs Einfügen, Merge-Faktor erhöht	10
Oceanic2		
Roland&Daniel	Lucene, RAMDisk, StandardAnalyzer	8+
TeamJA	Lucene, StandardAnalyzer	<5

Approach, Effort Invested (Tuesday)

Group	Index	Effort (h)
Bertram&Abdalla	StandardAnalyzer,	5-6
BeSharps	Lucene, StandardAnalyzer	5
Bruno&Christin	StandardAnalyzer	5
Darko&Mario		
Dinh&Feyco	RamDisk, StandardAnalyzer, Caching deaktiviert, Scoring deaktiviert (Collector), Codec gewechselt, Buffer erhöht, Multi-Threading.	15-20
HouseOfMojo		
InfoRetImWinter	StandardAnalyzer, Multi-Threading Indexing	6
jp&bl	StandardAnalyzer	
Julian&Dennis		
lc&mk		
lucy@molly	RamDisk, StandardAnalyzer, Multi-Threading Indexing	6
Oceanic2		
Paul&Peter	StandardAnalyzer	5-6
Vincent&Max	StandardAnalyzer, RamDisk	3

Student Presentations (Monday)

- Lucene Indexer: InfoRet2000 (v.a. Tobias)
- Lucene Query Parser: getSchwifty()

- Damit ergeben sich die folgenden Gruppen, die am **30. Januar** Aufgabe 4 vorstellen sollten: Roland&Daniel, TeamJA, Devin&Johannes.

Student Presentations (Tuesday)

- Lucene Indexer: Paul&Peter
- Lucene Query Parser: Bertram&Abdalla

- Damit ergeben sich die folgenden Gruppen, die am **31. Januar** Aufgabe 4 vorstellen sollten: Julian&Dennis, Lorenz aus Gruppe lc&mk.

Evaluation Setup

- One run:
  - build index (discard submission if index not built after one hour),
  - run 11 test queries (discard submission if any query is unsuccessful), and
  - report runtimes for building index and running each query.

- Ten runs per submission.
- Determine median runtimes.

- Determine ranks (across all submissions) for building the index and running each evaluation query.
- Compute average ranks across all 12 ranks.

The 11 Evaluation Queries

1. title:"game of thrones" AND type:episode AND (plot:Bastards OR (plot:Jon AND plot:Snow)) -plot:son
2. title:"Star Wars" AND type:movie AND plot:Luke AND year:[1977 TO 1987]
3. plot:Berlin AND plot:wall AND type:television
4. plot:men~1 AND plot:women~1 AND plot:love AND plot:fool AND type:movie
5. title:westworld AND type:episode AND year:2016 AND plot:Dolores
6. plot:You AND plot:never AND plot:get AND plot:A AND plot:second AND plot:chance
7. plot:Hero AND plot:Villain AND plot:destroy AND type:movie
8. (plot:lover -plot:perfect) AND plot:unfaithful* AND plot:husband AND plot:affair AND type:movie
9. (plot:Innocent OR plot:Guilty) AND plot:crime AND plot:murder AND plot:court AND plot:judge AND type:movie
10. plot:Hero AND plot:Marvel -plot:DC AND type:movie
11. plot:Hero AND plot:DC -plot:Marvel AND type:movie

Median Runtimes (21/23 submissions)

	Index / s	Query 1 (ms)	Query 2 (ms)	Query 3 (ms)	Query 4 (ms)	Query 5 (ms)	Query 6 (ms)	Query 7 (ms)	Query 8 (ms)	Query 9 (ms)	Query 10 (ms)	Query 11 (ms)
Bertram&Abdalla	46,80	30,19	38,51	29,00	44,79	27,54	30,98	27,07	37,54	30,00	32,71	26,48
BeSharps	65,73	14,72	21,91	10,62	24,61	9,57	11,72	9,30	14,84	10,01	8,96	11,18
BigGoogle	50,25	10,20	18,67	8,08	24,12	8,57	7,88	7,64	10,70	8,07	6,25	6,37
bl_jp_1	47,01	26,16	41,67	25,39	49,08	23,70	35,01	26,21	25,95	22,80	38,44	22,80
Christin&Bruno_1	43,27	32,09	31,50	26,45	57,59	30,78	29,31	34,99	36,10	30,37	27,78	27,97
Dinh&Feyco	27,97	3,06	1,93	0,94	30,44	0,69	2,42	1,36	1,36	0,95	0,73	0,58
getSchwifty	113,01	25,26	33,81	22,13	38,00	22,10	24,36	23,07	29,79	23,22	23,52	27,53
HendrikundSinje	57,23	18,81	16,59	14,83	32,60	13,55	17,30	14,32	15,23	14,88	12,91	13,42
houseofmojo	30,07	44,50	48,25	35,71	52,42	25,82	27,97	27,10	46,19	31,17	28,31	31,86
ilGruppo	77,51	22,58	25,24	19,61	34,79	18,69	19,65	25,95	24,07	25,35	20,52	19,77
Inforet2000	57,11	30,49	23,91	22,10	30,56	20,61	20,83	20,56	22,60	21,66	25,66	20,60
InformationRetrievalImWinter	24,42	34,53	61,43	33,57	55,86	47,42	35,23	42,30	45,08	34,41	38,99	36,16
lc+mk	61,73	29,34	32,25	24,76	38,59	21,49	24,32	23,15	27,80	23,86	21,93	34,86
lucy_molly	23,32	25,27	33,81	20,34	35,99	22,56	19,99	18,77	26,38	21,62	19,44	20,13
newDimensions	11,81	59,38	63,76	81,02	138,85	46,68	56,70	56,46	65,12	53,23	88,88	47,69
Oceanic2	56,89	30,21	32,12	26,79	42,70	26,33	25,97	24,37	54,43	27,31	25,01	25,49
peterpaul2	55,71	32,94	49,22	29,22	52,88	26,73	29,51	26,63	31,81	27,70	25,97	30,83
RolandDaniel	51,51	30,51	31,75	20,98	37,51	19,95	29,46	19,20	28,80	23,70	21,68	82,68
TeamJA	60,26	30,77	35,69	30,88	50,40	25,61	28,50	25,77	34,99	26,09	23,47	25,76
vincentmax	43,54	27,79	39,50	31,28	38,52	22,47	22,30	23,65	26,57	23,20	21,75	22,71

Average Ranks (Top 10, Final Results)

	Ranks	Points
Dinh&Feyco	1,4	5
BigGoogle	2,7	3
BeSharps	4,3	2
HendrikundSinje	4,8	1
lucy_molly	6,7	1
ilGruppo	7,8	
Inforet2000	8,0	
vincentmax	9,9	
RolandDaniel	10,3	
getSchwifty()	10,6	

Old Ranking (Assignment 1 + 2)

Group	Points
Dinh&Feyco	10
lucy@molly	5
InfoRet2000	4
ilGruppo	3
NewDimensions	2
TeamJA	2
Julian&Dennis	1
jp&bl	1
BeSharps	1

Current Ranking (Assignment 1,2,3)

Group	Points
Dinh&Feyco	15
lucy@molly	6
InfoRet2000	4
ilGruppo	3
BigGoogle	3
BeSharps	3
NewDimensions	2
TeamJA	2
Julian&Dennis	1
jp&bl	1
HendrikundSinje	1

Some possible Optimizations

- Specify a ram-directory

```
Directory index = new RAMDirectory();
IndexWriterConfig iwc = new IndexWriterConfig(myAnalyzer);
iwc.setMaxBufferedDocs(600000); // 522475 docs
iwc.setRAMBufferSizeMB(IndexWriterConfig.DISABLE_AUTO_FLUSH); // unlimited
iwc.setUseCompoundFile(false);
IndexWriter writer = new IndexWriter(index, iwc);
writer.forceMerge(1, true); // better single threaded performance
```

- Watch out for tokenization of StringFields and TextFields…

```
Document doc = new Document();
doc.add(new StringField("YEAR", id, StringField.Store.NO));
doc.add(new TextField("PLOT", title, TextField.Store.YES));
```

- Limit the number of results to a reasonable number…

```
Query q = new QueryParser(null, analyzer).parse(queryString);
TopDocs hits = indexSearcher.search(q, 1000);
```