DIS17 – Search Engine Technologies

06 – Fine-tuning and Ranking

Philipp Schaer, Technische Hochschule Köln, Cologne, Germany

Version: WS 2021

Technology
Arts Sciences
TH Köln

Assignment I – Feedback

- 52 registered for teams
- 51 people registered to this course over PSSO
- 50 submitted assigment I
- 45 submitted assigment I CORRECTLY
  - wrong filenames
  - wrong queries

Assignment II – Write a concept paper

- This is the **first group assignment**.
- Write a **two page concept paper** (approx. 1000 words) and describe you planned approach for the Mini Trec campaign.
- Please make sure that the **following key aspects** are included in you concept paper:
  - There is a **rough project plan** or a **project outline**, which describes the **ideas**, the **procedure** and the **goals** of the project.
  - The idea is based on a coherent **analysis** of a problem regarding CORD-19, the topics or the relevance assessments.
  - The ideas should be based on concepts from the lectures DIS08, DIS12 or any other DIS-related lecture/project. Additionally you may refer to external literature.

Assignment II – Teams

Team name	Members
First	Paramalingam, Jennert, Puddu, Nebatz
Die Powerpuff Girls	Simunovic, Waldispühl, Schmer, Fumfack
TheGitGangTheory	Bateva, Blum, Kubon, Wolf
def init(elynot,self,ish)	Celik, Tanriver, Pekarski, Romer
The Boolean Business	Seker, Schenkel, Haupt, Schüller
Shit happens	Cagferoglu, Okwus, Pelzers, Przibylla
BugBustersReloaded	Prantz, Hilbrecht, Bilko, Vetter
Gehacktes Misch	Richter, Colic, Novak, Gelgurt
Die Drei von der Tankstelle	Depois, Mihretab, Poledniok, Arda
Information_Revival_Group	Courtpozanis, Uhler, Landsiedel, Krause
DuckDuckBros	Fresi, Hardtke, Erdogan, Loose
The Internet Explorers	Szyprons, Ullrich, Schulz, Kim
Searchbusters	Akyürek, Alarslan, Disterhof, Pycha

Schedule for lectures WS 2021/22

15.10.21	Introduction	
22.10.21	IR in a Nutshell and TREC Evaluations	
29.10.21	Solr in a Nutshell	
05.11.21	Indexing	
12.11.21	Hands-on Session on Solr	
19.11.21	Queries	Assignment I due (Solr tutorial)
26.11.21	Project week (no lecture)	
03.12.21	Ranking	
10.12.21	Mini-TREC	Assignment II due (concept paper)
17.12.21	Mini-TREC	
24.12.21	Christmas (no lecture)	
31.12.21	Christmas (no lecture)	
07.01.22	Mini-TREC	
14.01.22	Results of Mini-TREC (World Café)	Assignment III 13.01.22 (run submission)
28.02.22		Assignment IV (term paper)

World Café

It‘s about an intensive and direct discussion!

The guests of the café discuss with each other and exchange their knowledge, experiences and ideas

Ideas, learning experiences and suggestions begin to connect as guests wander through the café

What is the idea behind World Cafés

The **informal atmosphere** supports the **creativity** of the participants.

- **Free exchange of ideas** and experiences is worth a lot
- Keeping results **light-weight** is enough Jamboards

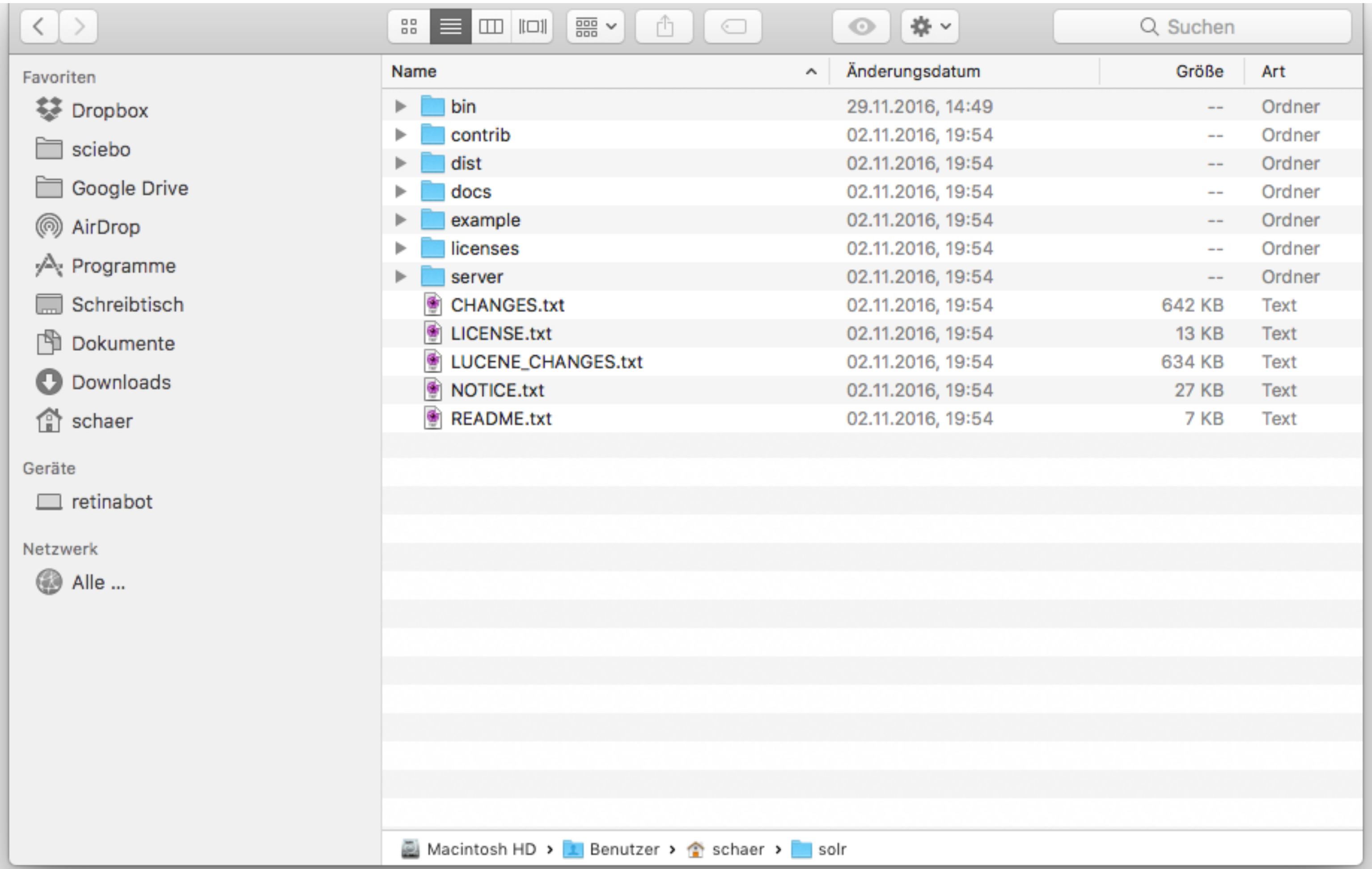
Due to the predefined questions and fixed discussion times, the participants **get straight to the topic**.

- Topics come from the project context and don't need introduction.
- Each of you has already dealt with it.

Each individual quickly engages in conversation with others.

- **No frontal** presentations
- **No leaders** of the word

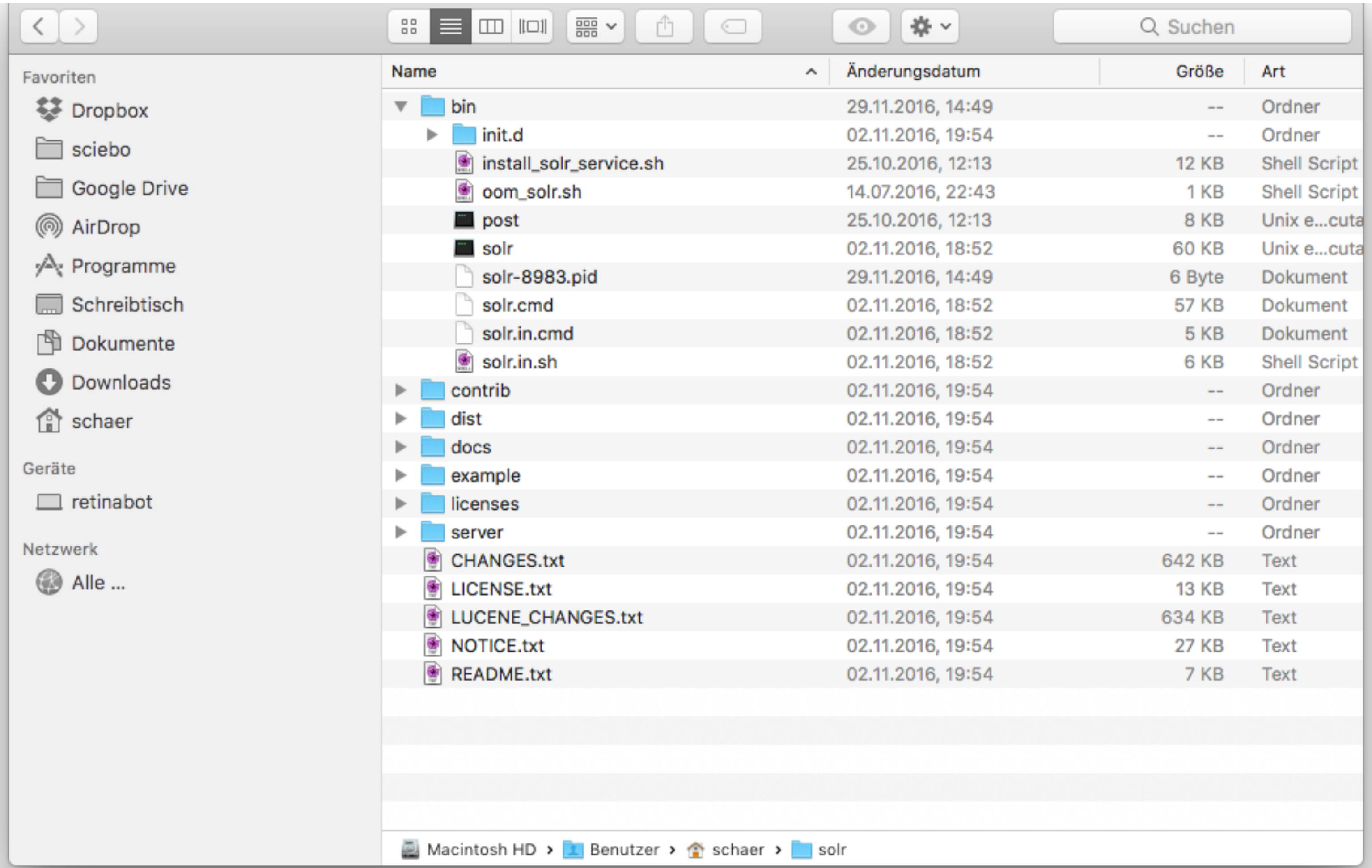
Solr‘s directory structure



Name	Änderungsdatum	Größe	Art
bin	29.11.2016, 14:49	--	Ordner
contrib	02.11.2016, 19:54	--	Ordner
dist	02.11.2016, 19:54	--	Ordner
docs	02.11.2016, 19:54	--	Ordner
example	02.11.2016, 19:54	--	Ordner
licenses	02.11.2016, 19:54	--	Ordner
server	02.11.2016, 19:54	--	Ordner
CHANGES.txt	02.11.2016, 19:54	642 KB	Text
LICENSE.txt	02.11.2016, 19:54	13 KB	Text
LUCENE_CHANGES.txt	02.11.2016, 19:54	634 KB	Text
NOTICE.txt	02.11.2016, 19:54	27 KB	Text
README.txt	02.11.2016, 19:54	7 KB	Text

Macintosh HD › Benutzer › schaer › solr

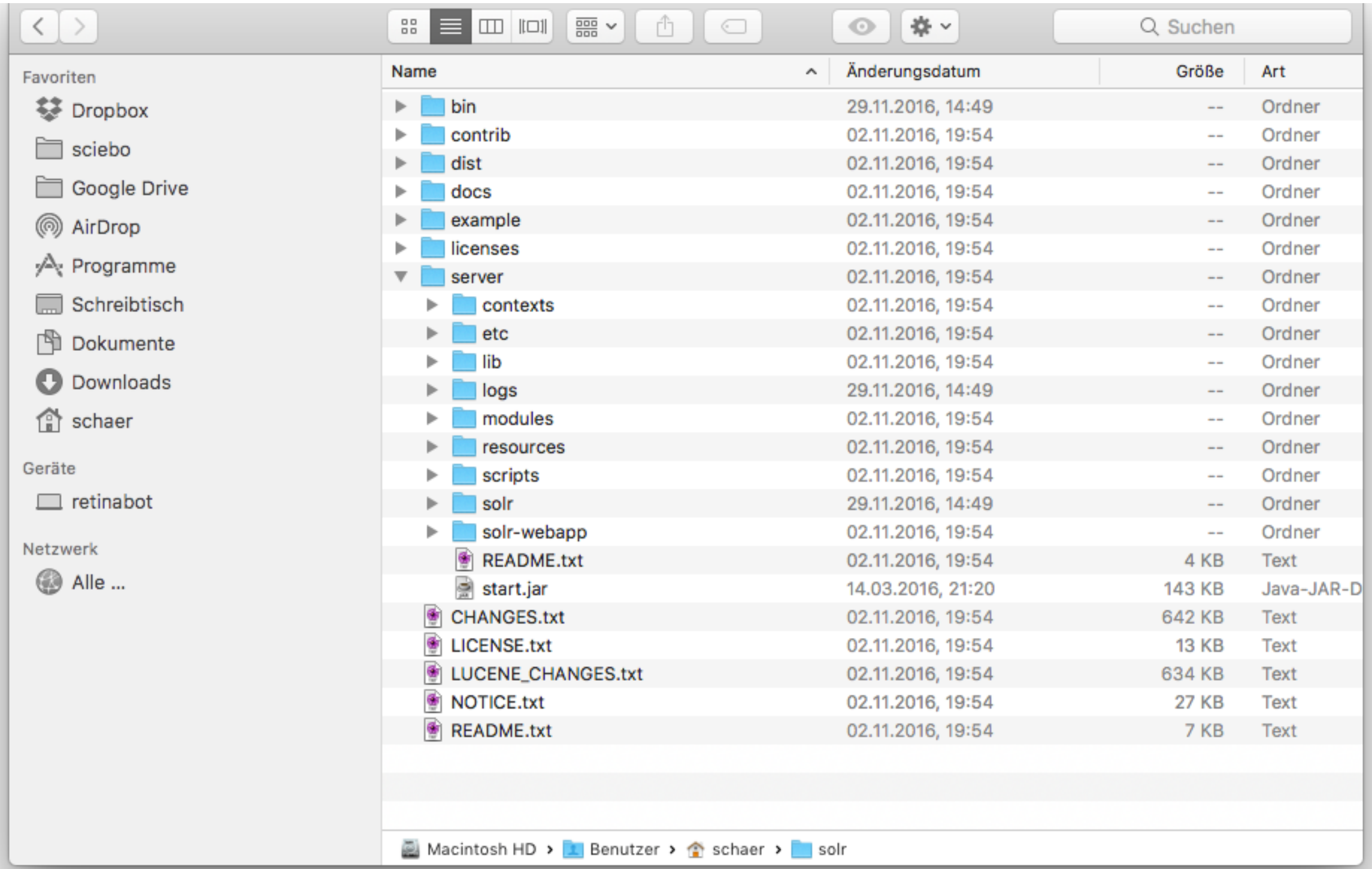
Solr‘s directory structure



Name	Änderungsdatum	Größe	Art
bin	29.11.2016, 14:49	--	Ordner
init.d	02.11.2016, 19:54	--	Ordner
install_solr_service.sh	25.10.2016, 12:13	12 KB	Shell Script
oom_solr.sh	14.07.2016, 22:43	1 KB	Shell Script
post	25.10.2016, 12:13	8 KB	Unix e...cuta
solr	02.11.2016, 18:52	60 KB	Unix e...cuta
solr-8983.pid	29.11.2016, 14:49	6 Byte	Dokument
solr.cmd	02.11.2016, 18:52	57 KB	Dokument
solr.in.cmd	02.11.2016, 18:52	5 KB	Dokument
solr.in.sh	02.11.2016, 18:52	6 KB	Shell Script
contrib	02.11.2016, 19:54	--	Ordner
dist	02.11.2016, 19:54	--	Ordner
docs	02.11.2016, 19:54	--	Ordner
example	02.11.2016, 19:54	--	Ordner
licenses	02.11.2016, 19:54	--	Ordner
server	02.11.2016, 19:54	--	Ordner
CHANGES.txt	02.11.2016, 19:54	642 KB	Text
LICENSE.txt	02.11.2016, 19:54	13 KB	Text
LUCENE_CHANGES.txt	02.11.2016, 19:54	634 KB	Text
NOTICE.txt	02.11.2016, 19:54	27 KB	Text
README.txt	02.11.2016, 19:54	7 KB	Text

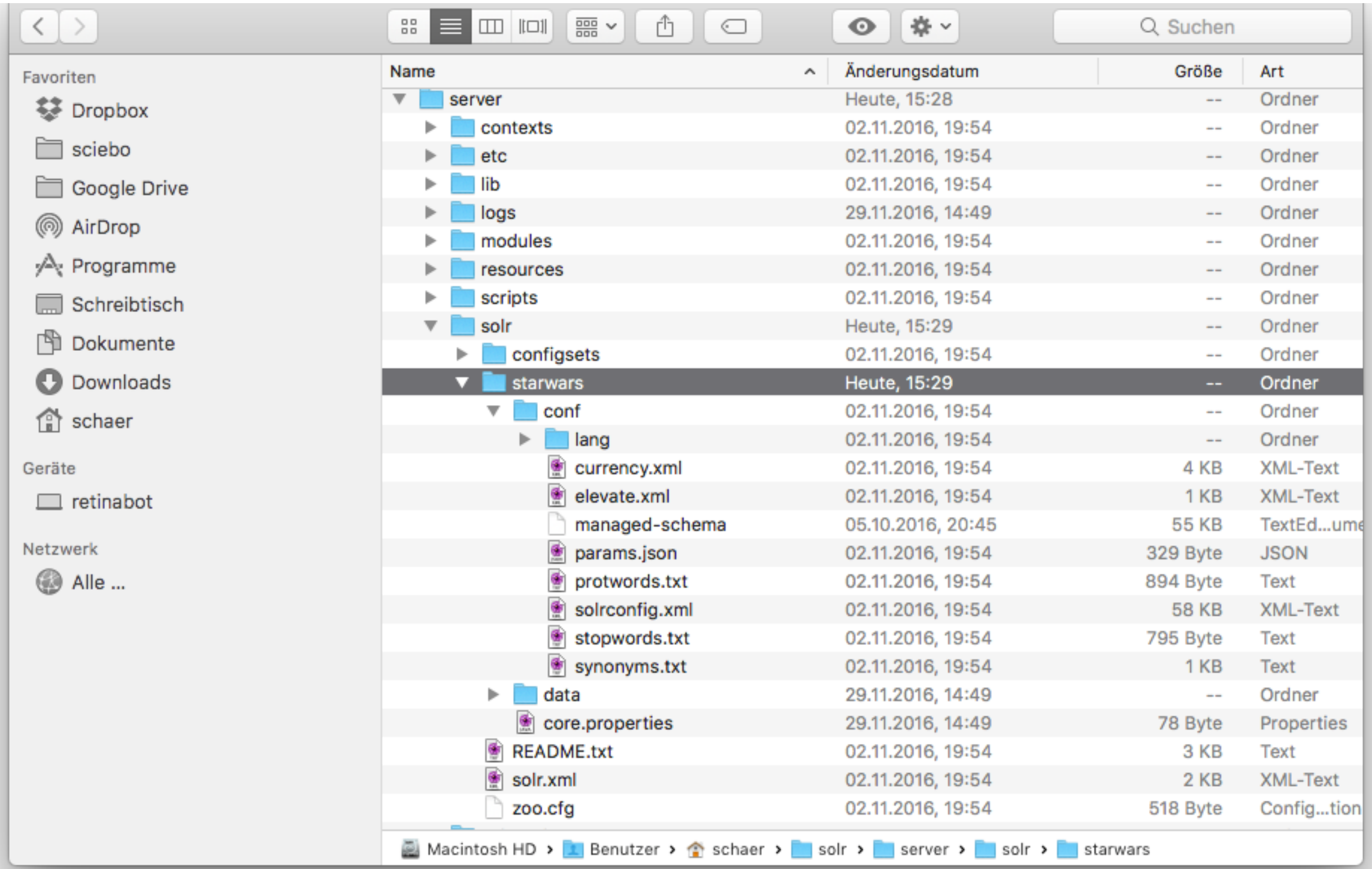
Macintosh HD › Benutzer › schaer › solr

Solr‘s directory structure – Server



Name	Änderungsdatum	Größe	Art
bin	29.11.2016, 14:49	--	Ordner
contrib	02.11.2016, 19:54	--	Ordner
dist	02.11.2016, 19:54	--	Ordner
docs	02.11.2016, 19:54	--	Ordner
example	02.11.2016, 19:54	--	Ordner
licenses	02.11.2016, 19:54	--	Ordner
server	02.11.2016, 19:54	--	Ordner
contexts	02.11.2016, 19:54	--	Ordner
etc	02.11.2016, 19:54	--	Ordner
lib	02.11.2016, 19:54	--	Ordner
logs	29.11.2016, 14:49	--	Ordner
modules	02.11.2016, 19:54	--	Ordner
resources	02.11.2016, 19:54	--	Ordner
scripts	02.11.2016, 19:54	--	Ordner
solr	29.11.2016, 14:49	--	Ordner
solr-webapp	02.11.2016, 19:54	--	Ordner
README.txt	02.11.2016, 19:54	4 KB	Text
start.jar	14.03.2016, 21:20	143 KB	Java-JAR-D
CHANGES.txt	02.11.2016, 19:54	642 KB	Text
LICENSE.txt	02.11.2016, 19:54	13 KB	Text
LUCENE_CHANGES.txt	02.11.2016, 19:54	634 KB	Text
NOTICE.txt	02.11.2016, 19:54	27 KB	Text
README.txt	02.11.2016, 19:54	7 KB	Text

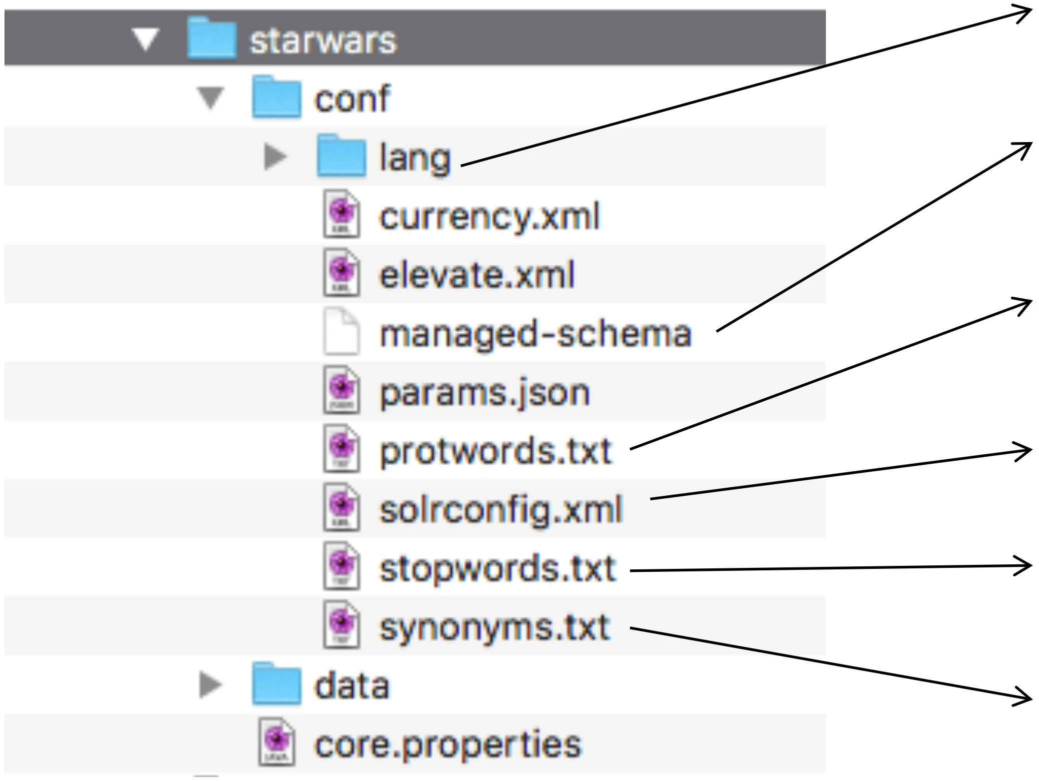
Solr‘s directory structure – Configuration



Name	Änderungsdatum	Größe	Art
server	Heute, 15:28	--	Ordner
contexts	02.11.2016, 19:54	--	Ordner
etc	02.11.2016, 19:54	--	Ordner
lib	02.11.2016, 19:54	--	Ordner
logs	29.11.2016, 14:49	--	Ordner
modules	02.11.2016, 19:54	--	Ordner
resources	02.11.2016, 19:54	--	Ordner
scripts	02.11.2016, 19:54	--	Ordner
solr	Heute, 15:29	--	Ordner
configsets	02.11.2016, 19:54	--	Ordner
starwars	Heute, 15:29	--	Ordner
conf	02.11.2016, 19:54	--	Ordner
lang	02.11.2016, 19:54	--	Ordner
currency.xml	02.11.2016, 19:54	4 KB	XML-Text
elevate.xml	02.11.2016, 19:54	1 KB	XML-Text
managed-schema	05.10.2016, 20:45	55 KB	TextEd...ume
params.json	02.11.2016, 19:54	329 Byte	JSON
protwords.txt	02.11.2016, 19:54	894 Byte	Text
solrconfig.xml	02.11.2016, 19:54	58 KB	XML-Text
stopwords.txt	02.11.2016, 19:54	795 Byte	Text
synonyms.txt	02.11.2016, 19:54	1 KB	Text
data	29.11.2016, 14:49	--	Ordner
core.properties	29.11.2016, 14:49	78 Byte	Properties
README.txt	02.11.2016, 19:54	3 KB	Text
solr.xml	02.11.2016, 19:54	2 KB	XML-Text
zoo.cfg	02.11.2016, 19:54	518 Byte	Config...tion

Macintosh HD › Benutzer › schaer › solr › server › solr › starwars

Overview on configuration files



```
▼ starwars
  ▼ conf
    ▶ lang
      currency.xml
      elevate.xml
      managed-schema
      params.json
      protwords.txt
      solrconfig.xml
      stopwords.txt
      synonyms.txt
  ▶ data
    core.properties
```

- lang → vorgefertigte **Sprachkonf**.
- managed-schema → Indexierungsschema / **schema.xml**
- protwords.txt → Ausnahmen für **Stemming**
- solrconfig.xml → Zentrale **Konfiguration**
- stopwords.txt → manuelle **Stopwortkontr**.
- synonyms.txt → **Query Expansion** mit Synonymen

managed_schema / schema.xml

- We use the schema.xml to configure our **indexing pipeline**!
- There are three (main) types of entries in schema.xml:
  - **`<fieldType>`**: Defines what types of data fields are available and how to process them, w./w.o. stop words, tokenizers, stemmers, …
  - **`<field> and <dynamicfield>`**: Are specific data fields, like „german text field“. Here we define if and how a specific field is configured, if it is a mandatory field, …
  - other entries are **`<uniqueKey>`** and **`<copyField>`**

If you would like to know more…

- http://www.solrtutorial.com/schema-xml.html
- http://wiki.apache.org/solr/SchemaXml

Dynamic fields

- The filed „id“ is predefined, but what about the rest?

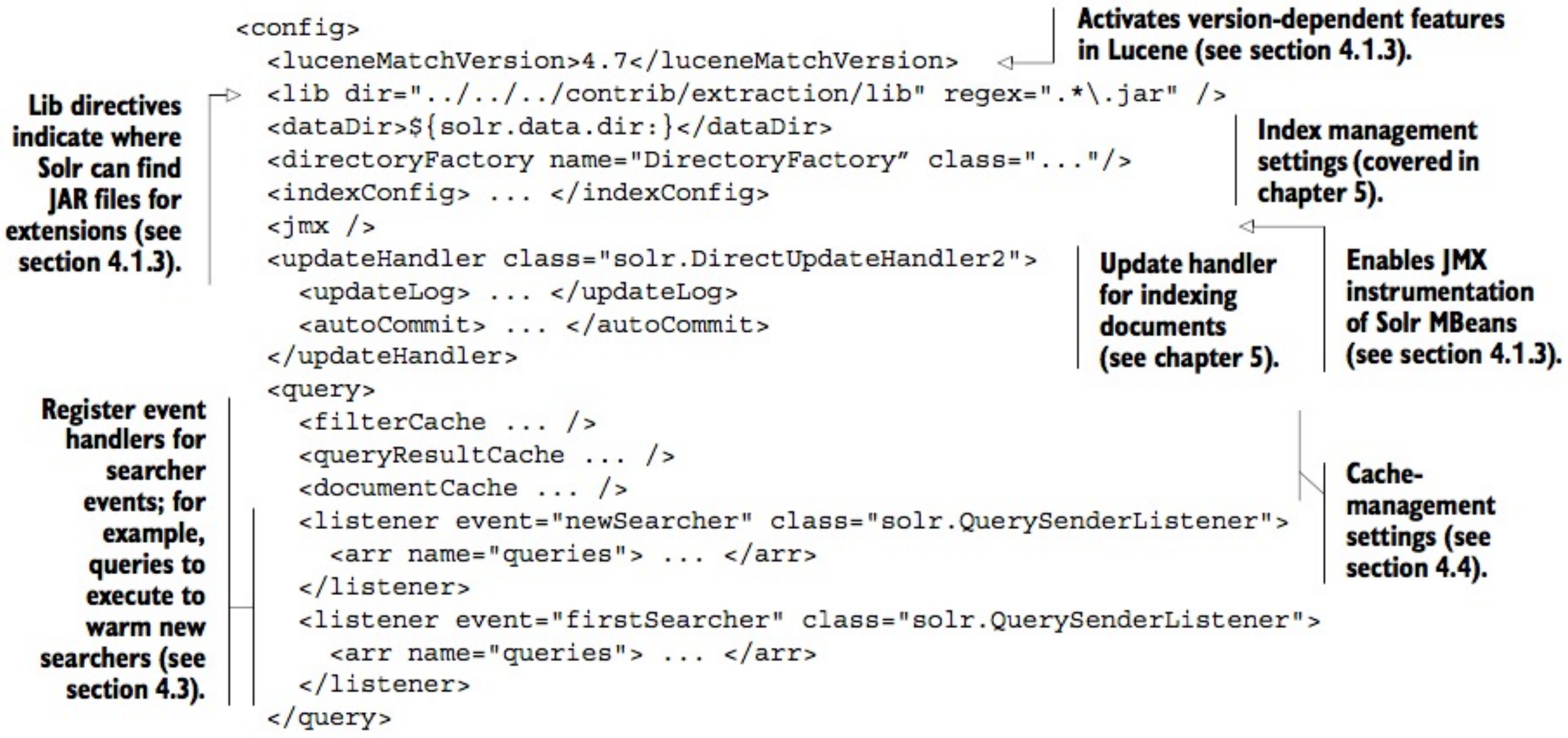
- You can map your data files and your Sorl data structure in different ways:
  - Adapt your **schema.xml**
  - Use the Schema API and/or the web interface
  - Use **dynamic fields.** Data fields that end on a specific **suffix** are processed in a special way. So, all fields that end on „_i“ will be processed as an integer.

Okay. But where is this schema.xml?

- The config file schema.xml is only available on explicit request! Since Solr 7 only the Schema API or the graphical interface can be used to alter the managed_schema.xml
  - The downside of this: The API is a mess
  - Not everything is possible via the web interface

- How can I create a schema.xml?
  - Put the following entry into your **solrconfig.xml**:
    - `<schemaFactory class="ClassicIndexSchemaFactory"/>`
  - Here is a short tutotarial:
    - https://cwiki.apache.org/confluence/display/solr/Schema+Factory+Definition+in+SolrConfig

solrconfig.xml



```
<config>
  <luceneMatchVersion>4.7</luceneMatchVersion>
  <lib dir="../../../contrib/extraction/lib" regex=".*\.jar" />
  <dataDir>${solr.data.dir:}</dataDir>
  <directoryFactory name="DirectoryFactory" class="..."/>
  <indexConfig> ... </indexConfig>
  <jmx />
  <updateHandler class="solr.DirectUpdateHandler2">
    <updateLog> ... </updateLog>
    <autoCommit> ... </autoCommit>
  </updateHandler>
  <query>
    <filterCache ... />
    <queryResultCache ... />
    <documentCache ... />
    <listener event="newSearcher" class="solr.QuerySenderListener">
      <arr name="queries"> ... </arr>
    </listener>
    <listener event="firstSearcher" class="solr.QuerySenderListener">
      <arr name="queries"> ... </arr>
    </listener>
  </query>
```

Activates version-dependent features in Lucene (see section 4.1.3).

Lib directives indicate where Solr can find JAR files for extensions (see section 4.1.3).

Index management settings (covered in chapter 5).

Enables JMX instrumentation of Solr MBeans (see section 4.1.3).

Update handler for indexing documents (see chapter 5).

Cache-management settings (see section 4.4).

Register event handlers for searcher events; for example, queries to execute to warm new searchers (see section 4.3).

Taken from: Solr in Action, Chapter 4

solrconfig.xml

THE central configuration file for your Solr instance

- Inclusion of **new filters** and **functionalities** (extensions)
- Changing the general flow of the index pipeline etc.
- Integration of e.g. **an own schema.xml** file
- Cache management / query warming / ...
- and much more.

Good news:

- It is **rarely necessary**, nevertheless:
- It helps to know that they exist and what you can theoretically do in them.

stopwords.txt

- Is to be seen in **addition to the language-specific stopwords**, which already exist for each language.
  - Language dependent stop word lists are located e.g. under lang/stopwords_de.txt or lang/stopwords_en.txt
  - The format is simple:
    - Each line start contains the stop word
    - Comments are introduced with a pipe symbol |.
- e.g.:

```
daß              |  that
derselbe         |  the same
derselben
denselben
...
```

synonyms.txt – Query Expansion

What can you do with synonyms and query expansion in Solr?

- User searches for „BRD“ and it automatically searches for „Bundesrepublik Deutschland“.
- User searches for „Star Wars“ and Solr searches for „Krieg der Sterne“ at the same time.

Solr offers **two variants** of how terms can be inserted into the TokenStream:

- At indexing time
- At query time

Check the Solr Reference Guide on the „Synonym Graph Filter“

https://solr.apache.org/guide/8_1/filter-descriptions.html#synonym-graph-filter

synonyms.txt – Query Expansion

```
couch,sofa,divan
teh => the
huge,ginormous,humungous => large
small => tiny,teeny,weeny
```

Simple file format for synonyms

- **A comma-separated list of words**.
  If one of the words matches, all words in the list (including the original one) are replaced.
- **Two comma-separated lists, separated by a =>.**
  As soon as a word on the left side matches, it is replaced by the words on the right side. The original word is not included, unless it is also on the right side!

synonyms.txt – Query Expansion

Application at indexing time

- Synonyms are written into the index! They run through the indexing pipeline incl. all stemmers, filters etc. depending on the configuration.
- Bloats the index

Application at query time

- Synonyms are added to the query. Again, depending on the configuration, they run through the indexing pipeline incl. all stemmers, filters, etc. (but of course only those configured for this query).
- Index remains smaller, but can cause problems e.g. with search term suggestions or "related searches".

synonyms.txt – configuration

```
<fieldType name="text_general" class="solr.TextField"
 positionIncrementGap="100"      multiValued="true">
      <analyzer type="index">
        <tokenizer class="solr.StandardTokenizerFactory"/>
       <!-- in this example, we will only use synonyms at query time
        <filter class="solr.SynonymGraphFilterFactory"
            synonyms="index_synonyms.txt" ignoreCase="true“ />
         -->
     </analyzer>
      <analyzer type="query">
        <tokenizer class="solr.StandardTokenizerFactory"/>
       <filter class=" solr.SynonymGraphFilterFactory"
            synonyms="synonyms.txt“ ignoreCase="true" />
     </analyzer>
</fieldType>
```

Make up synonyms?

- For simple corrections, which are often found in log files, this certainly makes sense, but not for 100s of synonyms that you want to include.
- This is a job for specialized thesauri
  - **Thesaurus Sozialwissenschaften**
    - Interactive http://lod.gesis.org/thesoz/de.html
    - Download http://lod.gesis.org/download-thesoz.html
  - **Standard-Thesaurus Wirtschaft**
    - Interactive http://zbw.eu/stw/version/latest/
    - Download http://zbw.eu/stw/version/latest/about.rdf
  - **Wordnet**
    - Generic thesaurus for English
    - Download http://wordnetcode.princeton.edu/2.0/

Wordnet

- `s(100072647,1,'purchase',n,1,8).`
- **`s(100072841,1,'redemption',n,3,0).`**
- **`s(100072841,2,'repurchase',n,1,0).`**
- **`s(100072841,3,'buyback',n,1,0).`**
- `s(100073027,1,'trading',n,1,2).`
- `s(100073232,1,'bond_trading',n,1,0).`

- Can be automatically converted into the Solr format with the following program: https://issues.apache.org/jira/browse/LUCENE-2347

- There is now also a corresponding Solr parser: https://lucene.apache.org/core/8_11_0/analyzers-common/org/apache/lucene/analysis/synonym/WordnetSynonymParser.html

protwords.txt

Stemming sometimes produces funny results:

wedding	wednesday, wedges, ...
university	univers universum, universal, ...

To avoid such over-stemming errors, certain words can be excluded from stemming, or "protected".

To do this, simply add them to the **protwords.txt** file.

Getting down to business... Ranking!

Solr already offers a very powerful ranking (internally, Solr calls this “similarity”):

- **BM25** - a classic probabilistic ranking method has been the default ranking since version 6.0.
- In versions prior to 6.0, however, Solr used the **vector space model with TFIDF weighting as default**
- Additionally there are among others
  - Language Models with **Jelinek Mercer Smoothing**
  - Language Models with **Dirichlet Smoothing**
  - and much more...

- https://solr.apache.org/docs/8_1_0/solr-core/org/apache/solr/schema/SimilarityFactory.html

Change the default ranking

You have to decide whether you want to change the ranking **globally** or **per field**.

- Example: You want to change the ranking from BM25 to vector space model
- Add the following line(s) to your schema.xml file:
  - **`<similarity class="ClassicSimilarityFactory" />`**
  - Thus the "classical" vector space model is again the standard.
- You can also implement this for individual fields:

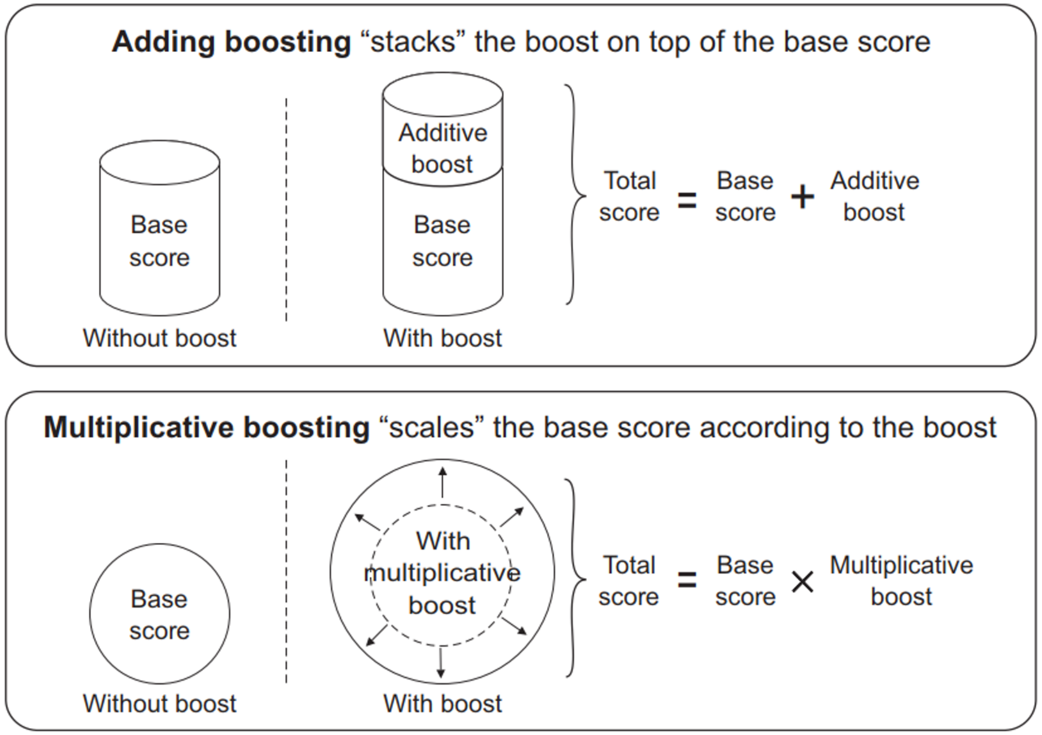
```
<fieldType name="text_classic" class="solr.TextField">
  <analyzer ... />
  <similarity class="ClassicSimilarityFactory" />
</fieldType>
```

Weighted Scoring Function

DocID	Feature 1		Feature 2		Feature n		Score	Rank
Doc 1	**a1** * f1	+	**a2** * f2	+	**an** * fn	=	x ∊ ℝ	r ∊ {1,...,k}
Doc 2	**a1** * f1	+	**a2** * f2	+	**an** * fn	=	x ∊ ℝ	r ∊ {1,...,k}
Doc 3	**a1** * f1	+	**a2** * f2	+	**an** * fn	=	x ∊ ℝ	r ∊ {1,...,k}
Doc 4	**a1** * f1	+	**a2** * f2	+	**an** * fn	=	x ∊ ℝ	r ∊ {1,...,k}
...	...		...		...		...	...
Doc k	**a1** * f1	+	**a2** * f2	+	**an** * fn	=	x ∊ ℝ	r ∊ {1,...,k}

Use weights to fine-tune the importance of the features

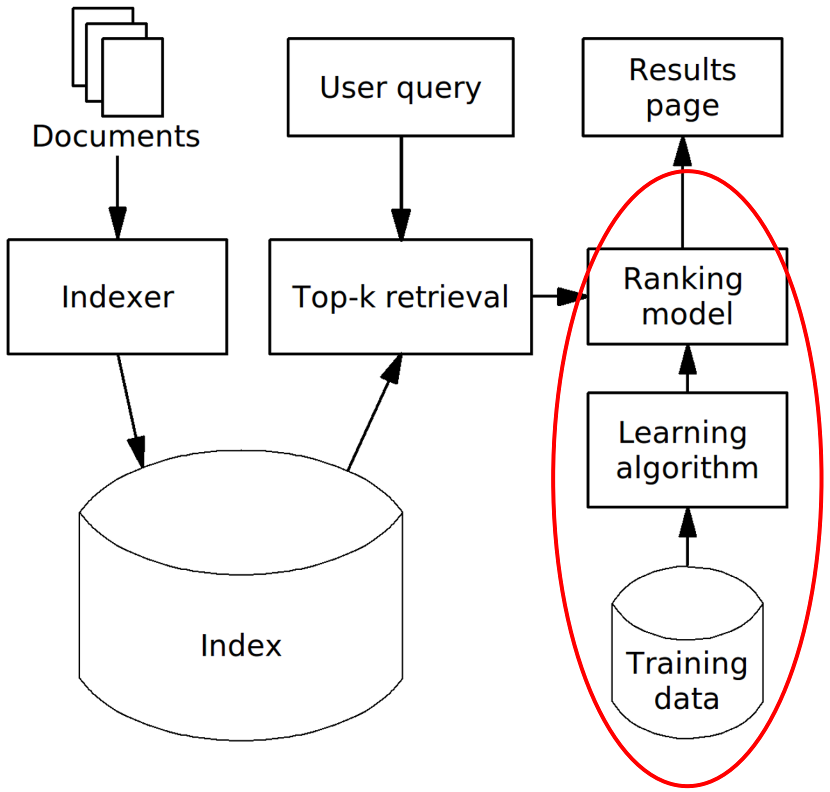
Use of boosting to get weighted features


Adding boosting "stacks" the boost on top of the base score
Base score
Without boost
Additive boost
Base score
With boost
Total score = Base score + Additive boost
Multiplicative boosting "scales" the base score according to the boost
Base score
Without boost
With multiplicative boost
With boost
Total score = Base score × Multiplicative boost

- With a **Boolean query**, you boost via an additional Boolean clause on top of the base query, using Lucene's bool query
- With a **function query**, you boost by directly modifying the ranking function by using Lucene's function_score query.

https://solr.apache.org/guide/8_1/function-queries.html

Learning To Rank



- Learns a ranking model designed to fit a specific use-case
  - **Point-wise**: learns based on single scores of the retrieved documents
  - **Pair-wise**: learns by comparing 2 neighboring documents
  - **List-wise**: learns by comparing the entire result list to a gold standard
  - In Cord-19, we have the topics, the documents, and relevance values, **such that we can apply Learning To Rank in our assignment**!

Learning To Rank in Solr/Elastic

Solr integrates Learning To Rank modules

- https://solr.apache.org/guide/8_1/learning-to-rank.html
- https://opensource.com/article/17/11/learning-rank-apache-solr

- But, we can also implement our own Learning To Rank (rather to re-rank) outside of Solr

Re-Ranking outside of Solr/Elastic

Python/Java L2R-modules

- With built in functionalities in Scikit-Learn https://towardsdatascience.com/learning-to-rank-with-python-scikit-learn-327a5cfd81f
- RankLib: A Java library of learning to rank algorithms https://github.com/codelibs/ranklib

Of course, we can also adjust a returned ranked list by re-ranking based on further features

- use the accompanying embeddings to re-rank the list of retrieved documents
- create a citation network from the full-texts and use network features for re-ranking, e.g., degree-centrality or betweenness centrality → e.g., use PageRank on the citation network

Happy hacking!

"IMG_1603" by teezeh is licensed under CC BY-SA 2.0