

Apache Solr

Learning to Rank FTW!

Berlin Buzzwords 2017
June 12, 2017

Diego Ceccarelli
Software Engineer, News Search
dceccarelli4@bloomberg.net

Michael Nilsson
Software Engineer, Unified Search
mnilsson23@bloomberg.net

techatbloomberg.com

© 2017 Bloomberg Finance L.P. All rights reserved.

Bloomberg



News Search at Bloomberg

325K+ Subscribers

9 Million Searches PER DAY

1 Million Stories

PUBLISHED EACH DAY

INDEX OF 500 MILLION STORIES

500 Stories

PER SECOND

Available for Search

in **~100ms**

180ms

RESPONSE TIME

More. Better. Faster.

Alerts in 100ms

1.5 MILLION

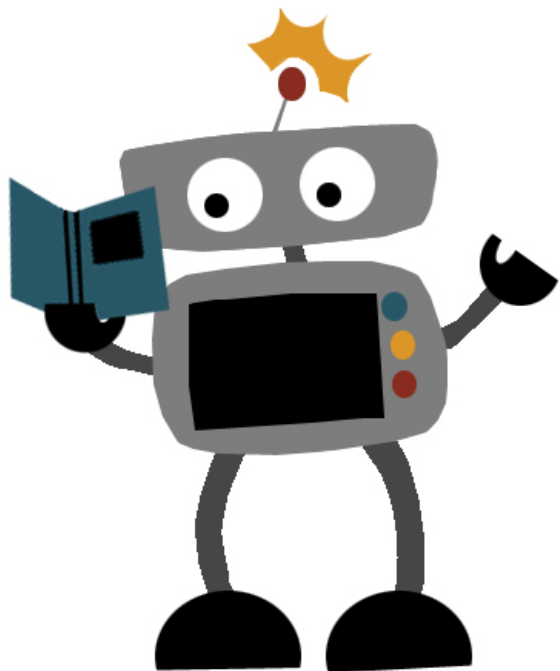
SAVED SEARCHES

What we did in the last few years...

- Bye-bye existing proprietary system
 - Inflexible, no scalable relevance sorting
- ... Enter Solr/Lucene!
 - Rich in features, extensible and actively maintained
 - Free software, we are involved and contribute back!
 - From-scratch alerting backend based on Lucene and Luwak
 - Scalable with load and data: Just add machines!
- Learning to Rank plugin upstreamed in Apache Solr 6.4
 - <https://cwiki.apache.org/confluence/display/solr/Learning+To+Rank>
 - <https://issues.apache.org/jira/browse/SOLR-8542>



Learning to Rank?



Machine Learned Ranking

Why Learning to Rank?

solr



Solr in 5 minutes - SolrTutorial.com

www.solrtutorial.com/solr-in-5-minutes.html ▼

Solr in 5 minutes. Solr makes it easy to run a full-featured search server. In fact, its so easy, I'm going to show you how in 5 minutes! Installing Solr; Starting Solr ...

PHP: Solr - Manual

php.net/manual/en/book.solr.php ▼

solr_get_version — Returns the current version of the Apache Solr extension ...
response from Solr; SolrClient::setServlet — Changes the specified servlet type ...

Spring Data Solr Tutorial - Petri Kainulainen

www.petrikainulainen.net/spring-data-solr-tutorial/ ▼

This tutorial describes how you can use Solr in your Spring powered applications.

eZ Find Demystified: Installing and configuring a multi-core ...

[share.ez.no > Learn > eZ Publish > eZ Find Demystified: Installing and...](#) ▼

eZ Find Demystified: Installing and configuring a multi-core Solr/eZ Find 2.6 instance with eZ Tika - PDF format. eZ Find Demystified - Installing and configuring a ...


Apache Solr -

lucene.apache.org/solr/ ▼

Solr is highly reliable, scalable and fault tolerant, providing distributed indexing, replication and load-balanced querying, automated failover and recovery, ...

score = 2.3 * BM25

Why Learning to Rank?

solr 

Spring Data Solr Tutorial - Petri Kainulainen
www.petrikainulainen.net/spring-data-solr-tutorial/ ▼
This tutorial describes how you can use Solr in your Spring powered applications.

Apache Solr -
lucene.apache.org/solr/ ▼
Solr is highly reliable, scalable and fault tolerant, providing distributed indexing, replication and load-balanced querying, automated failover and recovery, ...

eZ Find Demystified: Installing and configuring a multi-core ...
[share.ez.no > Learn > eZ Publish > eZ Find Demystified: Installing and...](#) ▼
eZ Find Demystified: Installing and configuring a multi-core Solr/eZ Find 2.6 instance with eZ Tika - PDF format. eZ Find Demystified - Installing and configuring a ...

Solr in 5 minutes - SolrTutorial.com
www.solrtutorial.com/solr-in-5-minutes.html ▼
Solr in 5 minutes. Solr makes it easy to run a full-featured search server. In fact, its so easy, I'm going to show you how in 5 minutes! Installing Solr; Starting Solr ...

PHP: Solr - Manual
php.net/manual/en/book.solr.php ▼
solr_get_version — Returns the current version of the Apache Solr extension ...
response from Solr; SolrClient::setServlet — Changes the specified servlet type ...

$$\begin{aligned} \text{score} &= 2.3 * \text{BM25} \\ &+ 4.5 * \text{BM25}(\text{title}) \\ &+ 5.2 * \text{BM25}(\text{desc}) \end{aligned}$$

Why Learning to Rank?

solr



Solr in 5 minutes - SolrTutorial.com

www.solrtutorial.com/solr-in-5-minutes.html ▼

Solr in 5 minutes. Solr makes it easy to run a full-featured search server. In fact, its so easy, I'm going to show you how in 5 minutes! Installing Solr; Starting Solr ...

Apache Solr -

lucene.apache.org/solr/ ▼

Solr is highly reliable, scalable and fault tolerant, providing distributed indexing, replication and load-balanced querying, automated failover and recovery, ...

Spring Data Solr Tutorial - Petri Kainulainen

www.petrikainulainen.net/spring-data-solr-tutorial/ ▼

This tutorial describes how you can use Solr in your Spring powered applications.

eZ Find Demystified: Installing and configuring a multi-core ...

[share.ez.no > Learn > eZ Publish > eZ Find Demystified: Installing and...](#) ▼

eZ Find Demystified: Installing and configuring a multi-core Solr/eZ Find 2.6 instance with eZ Tika - PDF format. eZ Find Demystified - Installing and configuring a ...

PHP: Solr - Manual

php.net/manual/en/book.solr.php ▼

`solr_get_version` — Returns the current version of the Apache Solr extension ...
`response from Solr; SolrClient::setServlet` — Changes the specified servlet type ...

$$\begin{aligned} \text{score} &= 2.3 * \text{BM25} \\ &+ 4.5 * \text{BM25}(\text{title}) \\ &+ 5.2 * \text{BM25}(\text{desc}) \\ &+ 0.1 * \text{doc-length} \\ &+ 1.3 * \text{freshness} \end{aligned}$$

Problem setup

- It's hard to manually tweak the ranking
 - You must be an expert in the domain

$$\begin{aligned} \text{score} &= 2.3 * \text{BM25} \\ &+ 4.5 * \text{BM25}(\text{title}) \\ &+ 5.2 * \text{BM25}(\text{desc}) \\ &+ 0.1 * \text{doc-length} \\ &+ 1.3 * \text{freshness} \end{aligned}$$

query = solr

query = lucene

query = london

query = bloomberg

query = ...

Learning to Rank plugin: Goals

- Automatically optimize for relevancy using machine learning
- Make different machine learning models pluggable
- Access to rich internal Solr search functionality for feature building

Steps for Machine Learned Ranking

- I. Collect query-document judgments [Offline]
- II. Extract query-document features [Solr]
- III. Train model with judgments + features [Offline]
- IV. Deploy model [Solr]
- V. Re-rank results [Solr]
- VI. Evaluate results [Offline]

Steps for Machine Learned Ranking

- I. Collect query-document judgments [Offline]
- II. Extract query-document features [Solr]
- III. Train model with judgments + features [Offline]
- IV. Deploy model [Solr]
- V. Re-rank results [Solr]
- VI. Evaluate results [Offline]

I. Collect Judgments

Curated relevance of documents per query

AAPL US



[Tim Cook - Wikipedia, the free encyclopedia](https://en.wikipedia.org/wiki/Tim_Cook)

https://en.wikipedia.org/wiki/Tim_Cook ▼

Timothy Donald "Tim" Cook (born November 1, 1960) is an American business executive, and is the chief executive officer of Apple Inc. Cook joined Apple in ...
[National Football Foundation - Auburn University - Scott Forstall](#)

[AAPL:NASDAQ GS Stock Quote - Apple Inc - Bloomberg ...](http://www.bloomberg.com/quote/AAPL:US)

www.bloomberg.com/quote/AAPL:US ▼

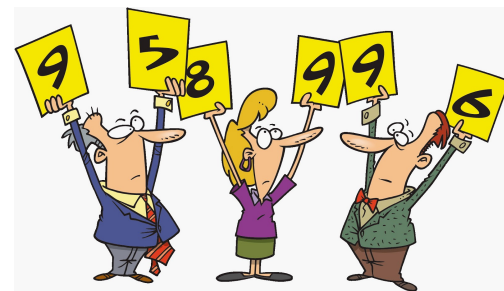
Stock analysis for Apple Inc (AAPL:NASDAQ GS) including stock price, stock chart, company news, key statistics, fundamentals and company profile.

[apple seeds](https://www.appleseedsplay.com)

<https://www.appleseedsplay.com> ▼

apple seedlings; camp; ... [New York](#). Bklyn Clinton Hill; Chelsea; ... Check your email for a notification from [apple seeds](#) containing your login credentials.

Judgement (good/bad)	Judgement (5 stars)
	3/5
	5/5
	0/5



I. Collect Judgments

- Explicit – judges assess search results manually
 - Experts
 - Crowd sourced
- Implicit – infer assessments through user behavior
 - Aggregated result clicks
 - Query reformulation
 - Dwell time

Steps for Machine Learned Ranking

- I. Collect query-document judgments [Offline]
- II. Extract query-document features [Solr]
- III. Train model with judgments + features [Offline]
- IV. Deploy model [Solr]
- V. Re-rank results [Solr]
- VI. Evaluate results [Offline]

II. Extract Features

Signals that give an indication of a result's importance

[Tim Cook - Wikipedia, the free encyclopedia](https://en.wikipedia.org/wiki/Tim_Cook)

https://en.wikipedia.org/wiki/Tim_Cook ▼

Timothy Donald "Tim" Cook (born November 1, 1960) is an American business executive, and is the chief executive officer of Apple Inc. Cook joined Apple in ...
[National Football Foundation - Auburn University - Scott Forstall](#)

[AAPL:NASDAQ GS Stock Quote - Apple Inc - Bloomberg ...](http://www.bloomberg.com/quote/AAPL:US)

www.bloomberg.com/quote/AAPL:US ▼

Stock analysis for Apple Inc (AAPL:NASDAQ GS) including stock price, stock chart, company news, key statistics, fundamentals and company profile.

[apple seeds](https://www.appleseedsplay.com)

<https://www.appleseedsplay.com> ▼

apple seedlings; camp; ... [New York](#). Bklyn Clinton Hill; Chelsea; ... Check your email for a notification from [apple seeds](#) containing your login credentials.

Query matches the title	Freshness	Is the document from Bloomberg.com?	Popularity
0	0.7	0	3583
1	0.9	1	625
0	0.1	0	129

II. Extract Features

- Define features to extract in myFeatures.json

- Deploy features definition file to Solr

```
curl -XPUT  
'http://localhost:8983/solr/myCollection/schema/feature-  
store' --data-binary "@/path/myFeatures.json" -H  
'Content-type:application/json'
```

```
[  
  {  
    "name": "matchTitle",  
    "type": "org.apache.solr.ltr.feature.SolrFeature",  
    "params": {  
      "q": "{!field f=title}${text}"  
    }  
  }, {  
    "name": "freshness",  
    "type": "org.apache.solr.ltr.feature.SolrFeature",  
    "params": {  
      "q": "{!func}recip(ms(NOW,timestamp),3.16e-11,1,1)"  
    }  
  },  
  { "name": "isFromBloomberg", ... },  
  { "name": "popularity", ... }  
]
```


II. Extract Features

- Add transformer to Solr config

```
<!-- Document transformer adding feature vectors with each retrieved document -->  
<transformer name="features"  
  class="org.apache.solr.ltr.response.transform.LTRFeatureLoggerTransformerFactory" />
```

- Request features for document

[http://localhost:8983/solr/myCollection/query?q=...&fl=*,\[features efi.text="APPL US"\]](http://localhost:8983/solr/myCollection/query?q=...&fl=*,[features efi.text=)

```
{  
  "title": "Tim Cook",  
  "url ": "https://en.wikipedia.org/wiki/Tim_Cook",  
  ...  
  "[features]": "matchTitle:0.0, freshness:0.7, isFromBloomberg:0.0, popularity:3583.0"  
}
```

Steps for Machine Learned Ranking

- I. Collect query-document judgments [Offline]
- II. Extract query-document features [Solr]
- III. Train model with judgments + features [Offline]**
- IV. Deploy model [Solr]
- V. Re-rank results [Solr]
- VI. Evaluate results [Offline]

III. Train Model

- Combine query-document judgments & features into training data file
- Train ranking model offline
 - RankSVM¹ [liblinear]
 - LambdaMART² [ranklib]

¹T. Joachims, *Optimizing Search Engines Using Clickthrough Data*, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2002.

²C.J.C. Burges, "*From RankNet to LambdaRank to LambdaMART: An Overview*", Microsoft Research Technical Report MSR-TR-2010-82, 2010.

Steps for Machine Learned Ranking

- I. Collect query-document judgments [Offline]
- II. Extract query-document features [Solr]
- III. Train model with judgments + features [Offline]
- IV. Deploy model [Solr]**
- V. Re-rank results [Solr]
- VI. Evaluate results [Offline]

IV. Deploy Model

- Generate trained output model in myModel.json
- Deploy model definition file to Solr

```
curl -XPUT
```

```
'http://localhost:8983/solr/techproducts/schema/mo  
del-store' --data-binary "@/path/myModel.json" -H  
'Content-type:application/json'
```

```
{  
  "class": "org.apache.solr.ltr.model.MultipleAdditiveTreesModel",  
  "name": "myModelName",  
  "features": [ { "name": "freshness"}, { "name": "matchTitle"}, ... ],  
  "params": {  
    "trees": [ {  
      "weight": 1,  
      "tree": {  
        "feature": "matchedTitle",  
        "threshold": 0.5,  
        "left": { "value": -100 },  
        "right": {  
          "feature": "freshness",  
          "threshold": 0.5,  
          "left": { "value": 50 },  
          "right": { "value": 75 }  
        }  
      }  
    }  
  ]  
}
```

Steps for Machine Learned Ranking

- I. Collect query-document judgments [Offline]
- II. Extract query-document features [Solr]
- III. Train model with judgments + features [Offline]
- IV. Deploy model [Solr]
- V. Re-rank results [Solr]**
- VI. Evaluate results [Offline]

V. Re-rank Results

- Add LTR query parser to Solr config

```
<!-- Query parser used to rerank top docs with a provided model -->  
<queryParser name="ltr" class="org.apache.solr.ltr.search.LTRQParserPlugin" />
```

- Search and re-rank results

<http://localhost:8983/solr/myCollection/query?q=...&>

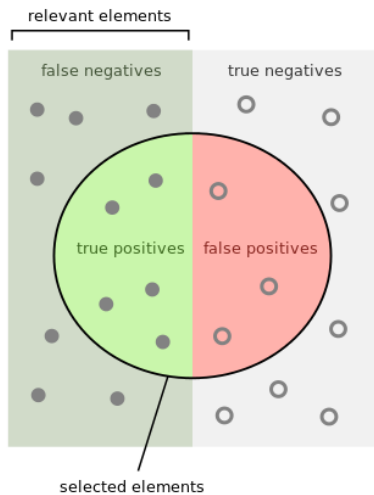
rq={!ltr model="myModelName" reRankDocs=100 efi.text="APPL US"}

- **ltr** – name of parser in config
- **model** – name of model in myModel.json
- **reRankDocs** – number of top K documents to re-rank
- **efi.key** – list of arbitrary key values to pass in to features

Steps for Machine Learned Ranking

- I. Collect query-document judgments [Offline]
- II. Extract query-document features [Solr]
- III. Train model with judgments + features [Offline]
- IV. Deploy model [Solr]
- V. Re-rank results [Solr]
- VI. Evaluate results [Offline]

VI. Evaluate quality of search



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Precision

how many relevant results I returned divided by total number of **results returned**

Recall

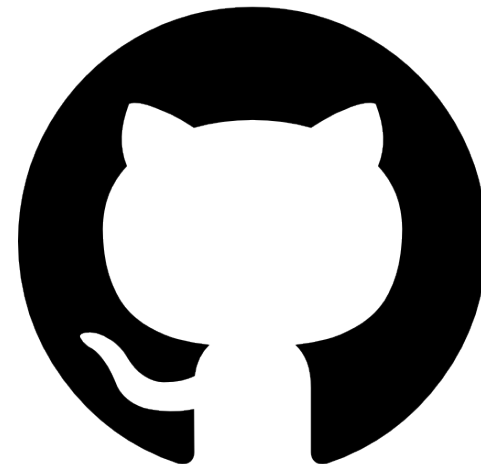
how many relevant results I returned divided by total number of **relevant results for the query**

F-Score

NDCG

How do I do this with real code?!?

- Demo time!
- Code for all steps available in GitHub



- <https://github.com/bloomberg/lucene-solr> branch:ltr-demo-lucene-solr

Configure solrconfig.xml

```
<!-- Query parser used to rerank top docs with a provided model -->  
<queryParser name="ltr" class="org.apache.solr.ltr.search.LTRQParserPlugin" />  
  
<!-- Document transformer adding feature vectors with each retrieved document -->  
<transformer name="features"  
  class="org.apache.solr.ltr.response.transform.LTRFeatureLoggerTransformerFactory" />
```

Setup

- Simple Wikipedia Json-dump (~150k articles)
- Index it into Solr
- Simple schema setting copy-field **text** containing all the text fields in the article
- The query hits **text** by default

Example of query

Document content:

<http://localhost:8983/solr/wikipedia/select?indent=on&q=berlin&wt=json>

Top 10 results:

<http://localhost:8983/solr/wikipedia/select?indent=on&q=berlin&wt=json&fl=title,score>

Collect query-document judgments

- Run `demo.py`

Write a Solr feature description file

- Provided in the demo (`features.json`)

- Example of feature:

```
{  
  "name": "freshness",  
  "type": "org.apache.solr.ltr.feature.SolrFeature",  
  "params": {  
    "q": "{!func}recip(ms(NOW,timestamp),3.16e-11,1,1)"  
  }  
}
```

- Current features: [http://localhost:8983/solr/wikipedia/schema/feature-store/ DEFAULT](http://localhost:8983/solr/wikipedia/schema/feature-store/DEFAULT)

Extract query-document features

- Using the Learning to Rank doc transformer

[http://localhost:8983/solr/wikipedia/select?indent=on&q=berlin&wt=json&fl=title,score,\[features efi.query=berlin\]](http://localhost:8983/solr/wikipedia/select?indent=on&q=berlin&wt=json&fl=title,score,[features efi.query=berlin])

Train models – Deploy – Evaluate results

- `train_linear_model.py`
- `train_tree_model.py`

Test a query outside the training set

- Rome

<http://localhost:8983/solr/wikipedia/select?indent=on&q=rome&wt=json&fl=title,score>

- LTR Rome

[http://localhost:8983/solr/wikipedia/select?indent=on&q=rome&wt=json&fl=title,score
&rq={!ltr model=... reRankDocs=30}](http://localhost:8983/solr/wikipedia/select?indent=on&q=rome&wt=json&fl=title,score&rq={!ltr model=... reRankDocs=30})

Q&A



techatbloomberg.com

Bloomberg