

Statistical Language Modeling for Information Access

Practical 2: Retrieval and evaluation

Maarten de Rijke Edgar Meij Krisztian Balog

University of Amsterdam
Norwegian University of Science and Technology

August 1–4, 2011

Outline of the Course

Practical

- Day 1: Installing and Indexing
- Day 2: Retrieval and Evaluation
- Day 3: Retrieval Parameters and Indri
- Day 4: Pseudo Relevance Feedback and Some More Evaluation;
Additional bells, whistles and requests

Looking back

- Downloaded/Installed Lemur
- Downloaded the CLEF 2006 adhoc collection
- Created an IndriBuildIndex parameter file
- Ran the indexer
- Viewed the output using LemurRet.jar
- Questions?
- You can always e-mail me later: `emeij@science.uva.nl`

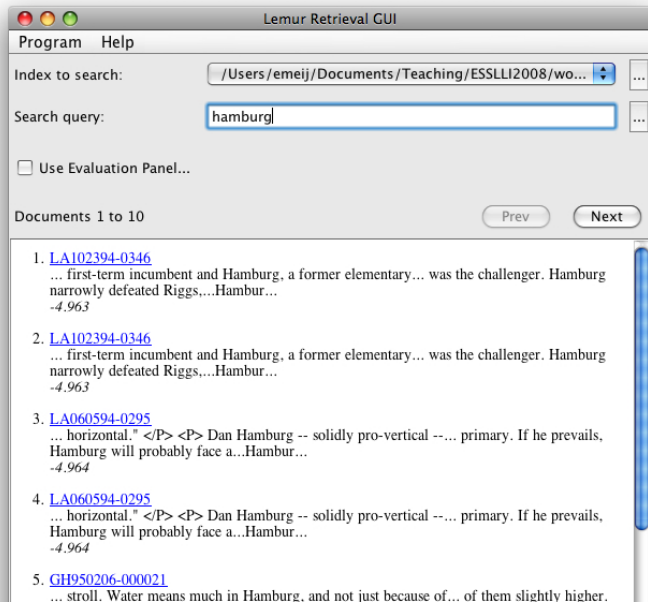
Looking back

- Downloaded/Installed Lemur
- Downloaded the CLEF 2006 adhoc collection
- Created an IndriBuildIndex parameter file
- Ran the indexer
- Viewed the output using LemurRet.jar
- Questions?
- You can always e-mail me later: emeij@science.uva.nl

Looking back

- Downloaded/Installed Lemur
- Downloaded the CLEF 2006 adhoc collection
- Created an IndriBuildIndex parameter file
- Ran the indexer
- Viewed the output using LemurRet.jar
- Questions?
- You can always e-mail me later: `emeij@science.uva.nl`

LemurRet.jar



Outline

1 Retrieval

- Queries
- Retrieval
- Retrieval Models

2 Evaluation

- qrels
- trec_eval

3 Exercises

- Exercises

Outline

1 Retrieval

- Queries
- Retrieval
- Retrieval Models

2 Evaluation

- qrels
- trec_eval

3 Exercises

- Exercises

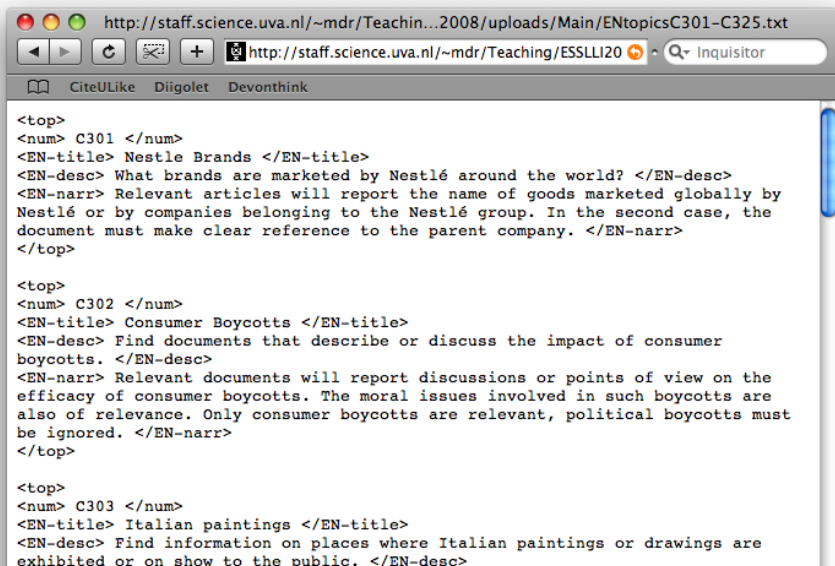
Retrieval

- General pipeline, given a set of queries:
 - ▶ Preprocess/Transform queries
 - ▶ Perform a retrieval run, using some settings
 - ▶ Evaluate
- Queries aren't usually in the “proper” format...

Retrieval

- General pipeline, given a set of queries:
 - ▶ Preprocess/Transform queries
 - ▶ Perform a retrieval run, using some settings
 - ▶ Evaluate
- Queries aren't usually in the “proper” format...

Queries — original format



The screenshot shows a web browser window with the address bar displaying `http://staff.science.uva.nl/~mdr/Teachin...2008/uploads/Main/ENTopicsC301-C325.txt`. The browser's address bar also shows a search bar with the text "Inquisitor". Below the address bar, there are several tabs: "CiteULike", "Diigolet", and "Devonthink". The main content area of the browser displays XML queries in a text editor. The queries are for topics C301, C302, and C303. Each query starts with `<top>` and `<num>`, followed by `<EN-title>`, `<EN-desc>`, and `<EN-narr>` tags. The queries are for Nestle Brands, Consumer Boycotts, and Italian paintings. The queries end with `</EN-narr>` and `</top>`.

```
<top>
<num> C301 </num>
<EN-title> Nestle Brands </EN-title>
<EN-desc> What brands are marketed by Nestlé around the world? </EN-desc>
<EN-narr> Relevant articles will report the name of goods marketed globally by
Nestlé or by companies belonging to the Nestlé group. In the second case, the
document must make clear reference to the parent company. </EN-narr>
</top>

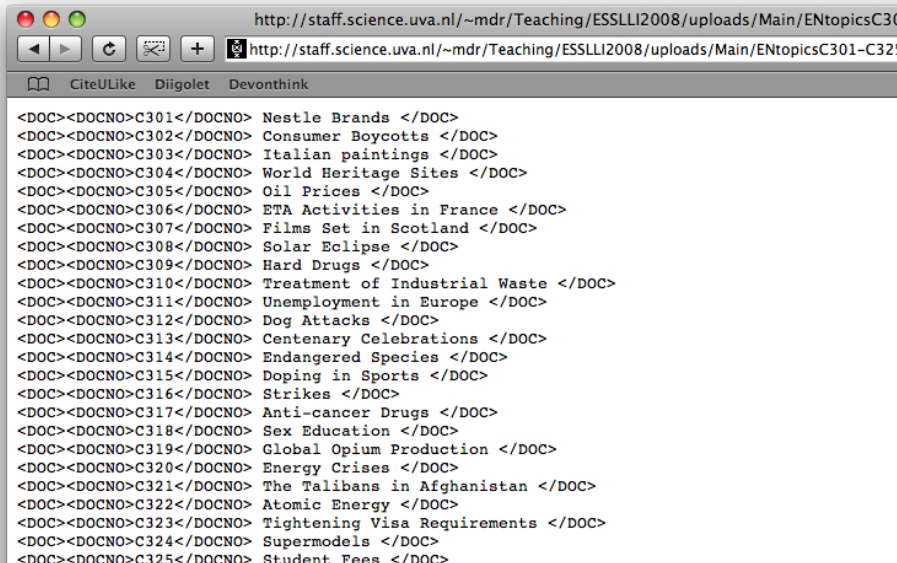
<top>
<num> C302 </num>
<EN-title> Consumer Boycotts </EN-title>
<EN-desc> Find documents that describe or discuss the impact of consumer
boycotts. </EN-desc>
<EN-narr> Relevant documents will report discussions or points of view on the
efficacy of consumer boycotts. The moral issues involved in such boycotts are
also of relevance. Only consumer boycotts are relevant, political boycotts must
be ignored. </EN-narr>
</top>

<top>
<num> C303 </num>
<EN-title> Italian paintings </EN-title>
<EN-desc> Find information on places where Italian paintings or drawings are
exhibited or on show to the public. </EN-desc>
```

Preprocessing

- Need to perform some kind of preprocessing to create TREC Text/Web format
- Use Perl/bash/awk/...
- Example topics and script on the wiki

Queries — TREC Web format



A screenshot of a web browser window. The address bar shows the URL: `http://staff.science.uva.nl/~mdr/Teaching/ESSLI2008/uploads/Main/ENTopicsC301-C325`. The browser has tabs for 'CiteULike', 'Diigolet', and 'Devonthink'. The main content area displays a list of documents in TREC Web format, each consisting of a document ID in angle brackets followed by the document title. The list includes:

- <DOC><DOCNO>C301</DOCNO> Nestle Brands </DOC>
- <DOC><DOCNO>C302</DOCNO> Consumer Boycotts </DOC>
- <DOC><DOCNO>C303</DOCNO> Italian paintings </DOC>
- <DOC><DOCNO>C304</DOCNO> World Heritage Sites </DOC>
- <DOC><DOCNO>C305</DOCNO> Oil Prices </DOC>
- <DOC><DOCNO>C306</DOCNO> ETA Activities in France </DOC>
- <DOC><DOCNO>C307</DOCNO> Films Set in Scotland </DOC>
- <DOC><DOCNO>C308</DOCNO> Solar Eclipse </DOC>
- <DOC><DOCNO>C309</DOCNO> Hard Drugs </DOC>
- <DOC><DOCNO>C310</DOCNO> Treatment of Industrial Waste </DOC>
- <DOC><DOCNO>C311</DOCNO> Unemployment in Europe </DOC>
- <DOC><DOCNO>C312</DOCNO> Dog Attacks </DOC>
- <DOC><DOCNO>C313</DOCNO> Centenary Celebrations </DOC>
- <DOC><DOCNO>C314</DOCNO> Endangered Species </DOC>
- <DOC><DOCNO>C315</DOCNO> Doping in Sports </DOC>
- <DOC><DOCNO>C316</DOCNO> Strikes </DOC>
- <DOC><DOCNO>C317</DOCNO> Anti-cancer Drugs </DOC>
- <DOC><DOCNO>C318</DOCNO> Sex Education </DOC>
- <DOC><DOCNO>C319</DOCNO> Global Opium Production </DOC>
- <DOC><DOCNO>C320</DOCNO> Energy Crises </DOC>
- <DOC><DOCNO>C321</DOCNO> The Talibans in Afghanistan </DOC>
- <DOC><DOCNO>C322</DOCNO> Atomic Energy </DOC>
- <DOC><DOCNO>C323</DOCNO> Tightening Visa Requirements </DOC>
- <DOC><DOCNO>C324</DOCNO> Supermodels </DOC>
- <DOC><DOCNO>C325</DOCNO> Student Fees </DOC>

ParseToFile

- But then we're still not there yet
- Lemur only understands Basic Doc Format (LDF)
- Use ParseToFile to parse an inputfile containing queries (TREC Text/Web format) into LDF format
- Steps:
 - ▶ Create ParseToFile parameter file (similar to the indexer's config file), e.g.:

```
<parameters>  
<docFormat>web</docFormat>  
<outputFile>path/to/outputfile.ldf</outputFile>  
</parameters>
```
 - ▶ Run: `ParseToFile [parse_param] [query-file]`

ParseToFile

- But then we're still not there yet
- Lemur only understands Basic Doc Format (LDF)
- Use ParseToFile to parse an inputfile containing queries (TREC Text/Web format) into LDF format
- Steps:
 - ▶ Create ParseToFile parameter file (similar to the indexer's config file), e.g.:

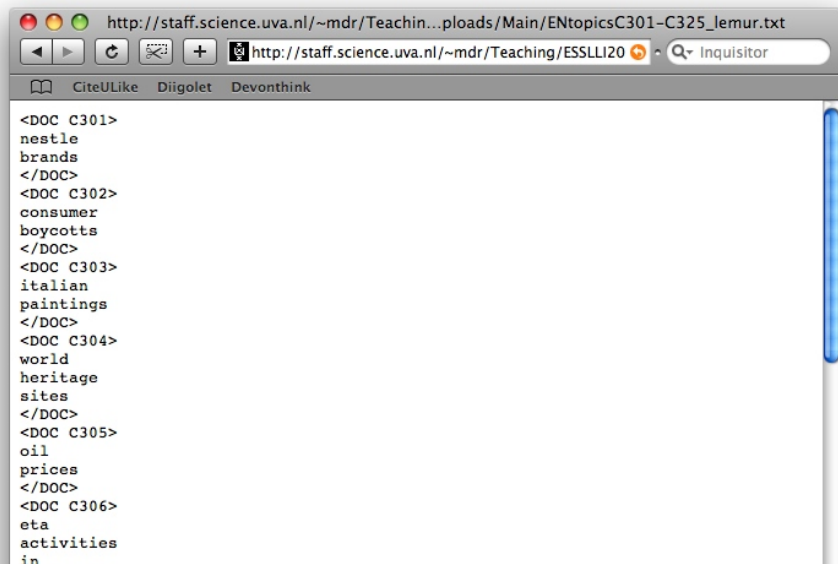
```
<parameters>  
<docFormat>web</docFormat>  
<outputFile>path/to/outputfile.ldf</outputFile>  
</parameters>
```
 - ▶ Run: `ParseToFile [parse_param] [query-file]`

ParseToFile

- But then we're still not there yet
- Lemur only understands Basic Doc Format (LDF)
- Use ParseToFile to parse an inputfile containing queries (TREC Text/Web format) into LDF format
- Steps:
 - ▶ Create ParseToFile parameter file (similar to the indexer's config file), e.g.:

```
<parameters>  
<docFormat>web</docFormat>  
<outputFile>path/to/outputfile.ldf</outputFile>  
</parameters>
```
 - ▶ Run: `ParseToFile [parse-param] [query-file]`

Queries — LDF format



A screenshot of a web browser window. The address bar shows the URL `http://staff.science.uva.nl/~mdr/Teaching/ESLLI20`. The browser's search bar contains the text "Inquisitor". Below the address bar, there are several tabs: "CiteULike", "Diigolet", and "Devonthink". The main content area of the browser displays a list of LDF format queries. Each query is enclosed in angle brackets and followed by a list of terms. The queries are: `<DOC C301>` with terms "nestle" and "brands"; `<DOC C302>` with terms "consumer" and "boycotts"; `<DOC C303>` with terms "italian" and "paintings"; `<DOC C304>` with terms "world", "heritage", and "sites"; `<DOC C305>` with terms "oil" and "prices"; and `<DOC C306>` with terms "eta" and "activities". The list is partially cut off at the bottom with "in".

```
<DOC C301>
nestle
brands
</DOC>
<DOC C302>
consumer
boycotts
</DOC>
<DOC C303>
italian
paintings
</DOC>
<DOC C304>
world
heritage
sites
</DOC>
<DOC C305>
oil
prices
</DOC>
<DOC C306>
eta
activities
in
```

Outline

1 Retrieval

- Queries
- Retrieval
- Retrieval Models

2 Evaluation

- qrels
- trec_eval

3 Exercises

- Exercises

Retrieval

- Works in the same way as indexing
 - ▶ Create parameter file
 - ▶ Basic parameters:
 - ★ index - the index to use
 - ★ retModel - the model to use
 - ★ textQuery - the query file to use
 - ★ resultCount - number of results
 - ★ resultFile - where to store the output
 - ★ **TRECResultFormat** - to use TREC-style output
 - ▶ `run RetEval [param_file]`

Example RetEval parameter file

```
<parameters>
<index>/path/to/your/index</index>
<retModel>kl</retModel>
<textQuery>path/to/queries.1df</textQuery>
<resultCount>1000</resultCount>
<resultFile>queries.res</resultFile>
<TRECResultFormat>1</TRECResultFormat>
</parameters>
```

Outline

1 Retrieval

- Queries
- Retrieval
- Retrieval Models

2 Evaluation

- qrels
- trec_eval

3 Exercises

- Exercises

Retrieval Models

- Lemur supports a number of retrieval models:
 - ▶ **kl** — KL-divergence (query-likelihood), with
 - ★ Jelinek-Mercer smoothing
 - ★ Dirichlet smoothing
 - ★ Absolute discount
 - ★ Two-stage smoothing
 - ▶ **tfidf** — TF.IDF
 - ▶ **cos** — Cosine
 - ▶ **okapi** — Okapi (BM25)
- You can (easily) implement your own!
- More on these tomorrow...

Small demo

Outline

1 Retrieval

- Queries
- Retrieval
- Retrieval Models

2 Evaluation

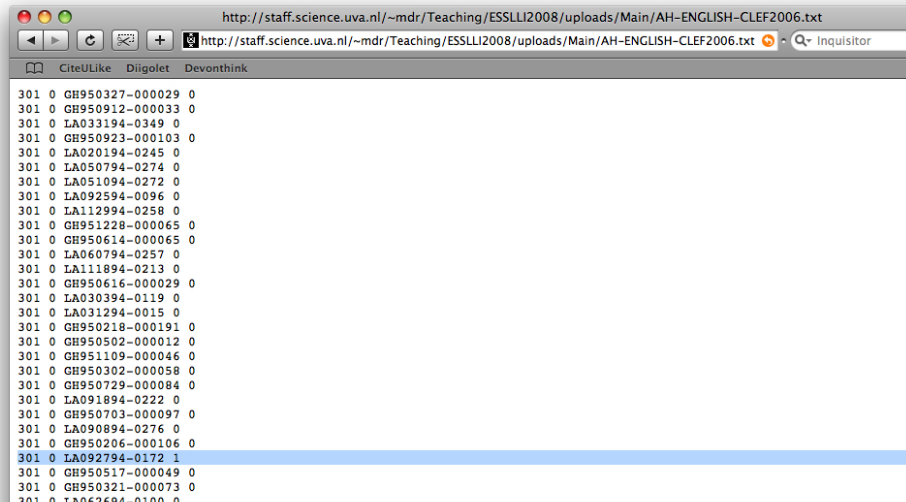
- qrels
- trec_eval

3 Exercises

- Exercises

qrels

A “qrel-file” contains the judgments on a document collection and set of queries



```
301 0 GH950327-000029 0
301 0 GH950912-000033 0
301 0 LA033194-0349 0
301 0 GH950923-000103 0
301 0 LA020194-0245 0
301 0 LA050794-0274 0
301 0 LA051094-0272 0
301 0 LA092594-0096 0
301 0 LA112994-0258 0
301 0 GH951228-000065 0
301 0 GH950614-000065 0
301 0 LA060794-0257 0
301 0 LA111894-0213 0
301 0 GH950616-000029 0
301 0 LA030394-0119 0
301 0 LA031294-0015 0
301 0 GH950218-000191 0
301 0 GH950502-000012 0
301 0 GH951109-000046 0
301 0 GH950302-000058 0
301 0 GH950729-000084 0
301 0 LA091894-0222 0
301 0 GH950703-000097 0
301 0 LA090894-0276 0
301 0 GH950206-000106 0
301 0 LA092794-0172 1
301 0 GH950517-000049 0
301 0 GH950321-000073 0
301 0 LA062684-0100 0
```

Outline

1 Retrieval

- Queries
- Retrieval
- Retrieval Models

2 Evaluation

- qrels
- trec_eval

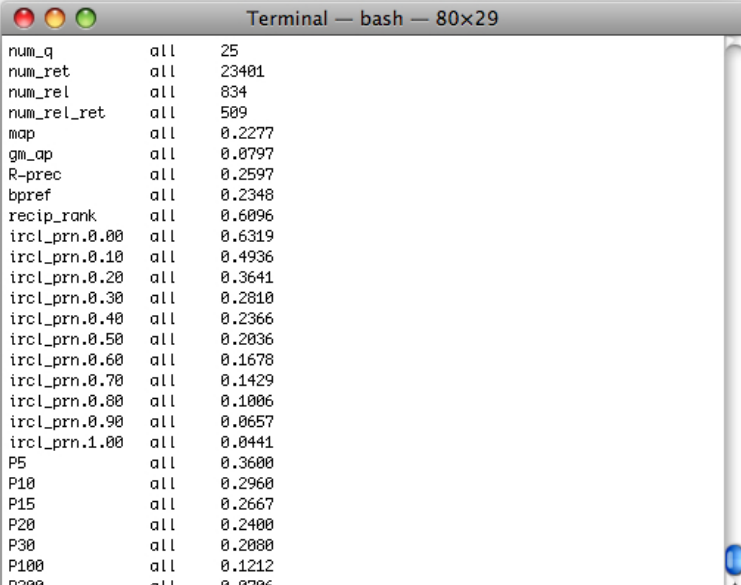
3 Exercises

- Exercises

trec_eval

- Written by Chris Buckley for the TREC evaluations
- Outputs evaluation measures
 - ▶ precision
 - ▶ recall
 - ▶ MRR
 - ▶ MAP
 - ▶ And many many more...
- Both for runs and individual topics (use the `-q` flag)
- Usage:
`trec_eval [path/to/qrels/] [path/to/resultfile]`

output



Terminal — bash — 80x29

num_q	all	25
num_ret	all	23401
num_rel	all	834
num_rel_ret	all	509
map	all	0.2277
gm_ap	all	0.0797
R-prec	all	0.2597
bpref	all	0.2348
recip_rank	all	0.6096
ircl_prn.0.00	all	0.6319
ircl_prn.0.10	all	0.4936
ircl_prn.0.20	all	0.3641
ircl_prn.0.30	all	0.2810
ircl_prn.0.40	all	0.2366
ircl_prn.0.50	all	0.2036
ircl_prn.0.60	all	0.1678
ircl_prn.0.70	all	0.1429
ircl_prn.0.80	all	0.1006
ircl_prn.0.90	all	0.0657
ircl_prn.1.00	all	0.0441
P5	all	0.3600
P10	all	0.2960
P15	all	0.2667
P20	all	0.2400
P30	all	0.2080
P100	all	0.1212
P200	all	0.0706

Outline

1 Retrieval

- Queries
- Retrieval
- Retrieval Models

2 Evaluation

- qrels
- trec_eval

3 Exercises

- Exercises

Exercises

- Compare results of two or more different retrieval models on the CLEF collection
- Report on (interesting) differences
- Where does another model help? hurt? In terms of precision, recall or some average?
- Why?