

# How to do an Information Retrieval Experiment with Indri

Ben Carterette

CMPSCI646

9/25/2007

## IR Experiments in Indri

### Steps:

1. Obtain a corpus of documents.
2. Download and compile Indri.
3. Create an indexing parameter file.
4. Run the indexer.
5. Create a querying parameter file.
6. Query your index.
7. Evaluate results.

## IR Experiments in Indri

### Steps:

1. Obtain a corpus of documents.
2. Download and compile Indri.
3. Create an indexing parameter file.
4. Run the indexer.
5. Create a querying parameter file.
6. Query your index.
7. Evaluate results.

## Obtain a Corpus

- A collection of documents to index and search.
- CIIR has many corpora available.
  - TREC discs 1-5, TDT, GALE, WT10G, GOV2, ...
- Indri indexes text files (also PDF, doc, ppt).
- A single text file can contain multiple documents.
  - SGML tags separate them.

## Example

- ft92.dat on TREC disc 4.

```
<DOC>
<DOCNO>FT921-1</DOCNO>
<DATE>920331</DATE>
<HEADLINE>FT 31 MAR 92 / International Company News:
  Rhone-Poulenc and SNIA in European link</HEADLINE>
<BYLINE>by REUTER</BYLINE>
<TEXT>RHONE-POULENC, the French chemicals and
  pharmaceuticals group, ...</TEXT>
</DOC>
<DOC>
<DOCNO>FT921-2</DOCNO>
...
</DOC>
...
```

## IR Experiments in Indri

### Steps:

1. Obtain a corpus of documents.
2. Download and compile Indri.
3. Create an indexing parameter file.
4. Run the indexer.
5. Create a querying parameter file.
6. Query your index.
7. Evaluate results.

## Download and Compile Indri

- Download from [www.lemurproject.org/indri](http://www.lemurproject.org/indri)
- Under UNIX-like OSes:
  - ./configure
  - make
  - make install
- Binaries available for Windows.
- Executables:
  - `buildindex` is used to index a corpus.
  - `runquery` is used to query a corpus.

## Running an Experiment Using Indri

### Steps:

1. Obtain a corpus of documents.
2. Download and compile Indri.
3. Create an indexing parameter file.
4. Run the indexer.
5. Create a querying parameter file.
6. Query your index.
7. Evaluate results.

## Indexing Parameters

- buildindex reads parameters from an XML file.
- Example:

```
<parameters>
  <corpus>
    <path>/work2/collections/trec/v4</path>
    <class>trectext</class>
  </corpus>
  <corpus>
    <path>/work2/collections/trec/v5</path>
    <class>trectext</class>
  </corpus>
  <stemmer>
    <name>krovetz</name>
  </stemmer>
  <index>/work/carteret/indexes/trec_vols_45</index>
</parameters>
```

## Corpus Classes

- Value of <class> parameter is determined by document format.
  - trectext
 

```
<DOC>
<DOCNO>APW19990102.0443</DOCNO>
<TEXT>...</TEXT>
</DOC>
```
  - trecweb
 

```
<DOC>
<DOCNO>WTX001-B01-10</DOCNO>
<DOCHDR>http://URL/</DOCHDR>
...
</DOC>
```
  - html
  - txt
  - pdf (Adobe PDF files)
  - doc (Word documents—requires Microsoft Office)
- If you have a different format, you will either have to:
  - Modify the corpus to fit one of these formats.
  - Write a parser for the format you have.

## Stopwords

- Prevent certain terms from being indexed.
- Add stopwords to the parameter file, or create a second parameter file.

```
<parameters>
  <stopper>
    <word>a</word>
    <word>the</word>
    <word>and</word>
    <word>or</word>
    ...
  </stopper>
</parameters>
```

## Incremental Indexing

- buildindex indexes documents incrementally.
- If run a second time with same `<index>` parameter, it will add documents to the existing index.
- It will not delete or overwrite an index.

## Running an Experiment Using Indri

### Steps:

1. Obtain a corpus of documents.
2. Download and compile Indri.
3. Create an indexing parameter file.
4. Run the indexer.
5. Create a querying parameter file.
6. Query your index.
7. Evaluate results.

## Run the Indexer

- `buildindex indexParamFile stopwordFile`
- Took 27 minutes to index TREC discs 4 and 5 (556,000 documents; 2 GB) on my desktop.

## Running an Experiment Using Indri

### Steps:

1. Obtain a corpus of documents.
2. Download and compile Indri.
3. Create an indexing parameter file.
4. Run the indexer.
5. Create a querying parameter file.
6. Query the index.
7. Evaluate results.

## Query Parameters

- The query parameter file specifies the index to query, the number of documents to retrieve, and the queries themselves.
- Example: TREC topics 301-350:

```
<parameters>
  <index>/work/carteret/indexes/trec_vols_45</index>
  <count>100</count>
  <trecFormat>true</trecFormat>
  <query>
    <number>301</number>
    <text>international organized crime</text>
  </query>
  <query>
    <number>302</number>
    <text>poliomyelitis and post-polio</text>
  </query>
  ...
</parameters>
```

## Indri Query Language

- Indri supports structured queries.
- Examples:
  - #1(white house) – match the phrase “white house”
  - #2(white house) – match the phrase “white \* house”, where \* is any word (or no word).
  - #uw2(white house) – match “white \* house” or “house \* white”
  - #syn(#1(white house) #1(united states)) – treat “white house” and “united states” as synonymous.
  - #or(#1(white house) #3(george bush)) – match either “white house” or “george \* \* bush”.

## Running an Experiment Using Indri

### Steps:

1. Obtain a corpus of documents.
2. Download and compile Indri.
3. Create an indexing parameter file.
4. Run the indexer.
5. Create a querying parameter file.
6. Query the index.
7. Evaluate results.

## Query the Index

- `runquery queryParamFile >resultsFile`
- Results are in 6-column format:
  - Query number, “Q0”, document ID, rank, score, “indri”
- E.g.:

```
301 Q0 CR93E-9750 1 -5.03454 indri
301 Q0 FBIS4-44962 2 -5.67745 indri
...
302 Q0 FBIS4-67701 1 -3.71833 indri
302 Q0 FBIS3-61373 2 -3.86585 indri
...
```

## Running an Experiment Using Indri

### Steps:

1. Obtain a corpus of documents.
2. Download and compile Indri.
3. Create an indexing parameter file.
4. Run the indexer.
5. Create a querying parameter file.
6. Query the index.
7. Evaluate results.

## Evaluate Results

- trec\_eval
  - [http://trec.nist.gov/trec\\_eval/trec\\_eval.7.3.tar.gz](http://trec.nist.gov/trec_eval/trec_eval.7.3.tar.gz)
  - trec\_eval -q qrels resultsFile
  - qrels for TREC data available from <http://trec.nist.gov/> or talk to me.
- trec\_eval calculates many common evaluation measures:
  - Average precision, precision@N, 11-point interpolated precision, R-precision, etc.
  - Assumes that unjudged documents are not relevant.

## More Info

- Trevor Strohman's Indri tips page:
  - <http://ciir.cs.umass.edu/~strohman/indri>
- Indri query language details:
  - <http://www.lemurproject.org/lemur/IndriQueryLanguage.php>
- Don Metzler's overview of the Indri retrieval model:
  - <http://ciir.cs.umass.edu/~metzler/indriretmodel.html>
- Indri reference:
  - Metzler, D. and Croft, W. B., "Combining the Language Model and Inference Network Approaches to Retrieval," *IP&M* 40(5), 735-750, 2004.

## Advanced Indexing and Querying

- Index fields:
  - `<field><name>byline</name></field>`
  - If `<byline>` tags are present in the documents, adding this line to the index parameter file will index them separately.
- Query fields:
  - `#1[byline](computer scientist)` will look for articles written by someone credited as a computer scientist in the byline.

## Additional Utilities

- `rmodel -query=query -index=index -documents=N -maxGrams=n`
  - Outputs a relevance model (an expanded query with weights for each term or phrase)
- `dumpindex index command [arguments]`
  - Dumps data from the index.