End-To-End Test Collection Evaluation using Indri on Killdevil

All of the examples below assume that you are in the directory: **/webdex/expir2013/**

To navigate to this directory, type **cd /webdex/expir2013/**

Step 1: Indexing

```
COLLECTION          INDRI BUILD          INDEX
(XML)       ⟹       INDEX         ⟹
```
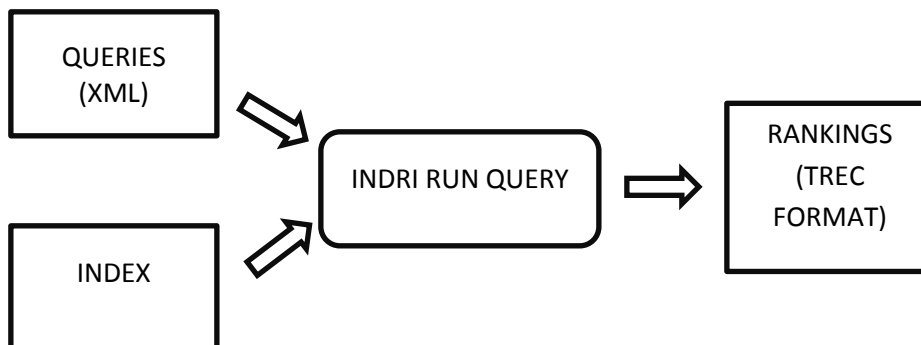
Command:

```
indri/indri-5.4/buildindex/IndriBuildIndex
    -corpus.path=corpus/twitter2011/tweets2011_en/
    -corpus.class=trectext
    -index=index/twitter2011_en/
```

Parameters:

(1) `corpus.path:` specifies the path to the root directory containing the corpus. Every file directly and indirectly under this root directory will be indexed.

(2) `corpus.class:` specifies the data format. For our purposes, this will always be `trectext`

(3) `index:` specifies the path to the directory where the index will be saved.

Step 2: Running a set of queries

```
QUERIES
(XML)                                      RANKINGS
            ⟹      INDRI RUN QUERY   ⟹     (TREC
INDEX       ⟹                              FORMAT)
```

Command:

```
indri/indri-5.4/runquery/IndriRunQuery
    corpus/twitter2011/queries.xml
```

```
-index=index/twitter2011_en/
-count=100
-trecFormat=true
```
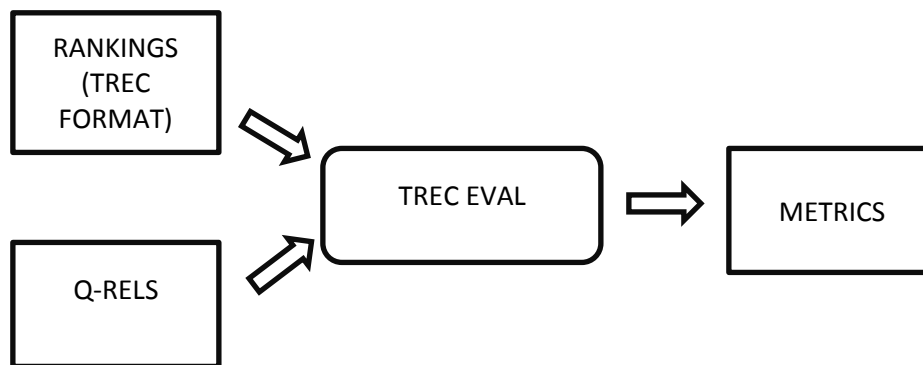
Parameters:

(1) `(QUERY FILE):` specifies the queries to be run in XML format.

(2) `index:` specifies the path to the directory where the index is stored.

(3) `count:` specifies the maximum number of top documents to retrieve for each query

(4) `trecFormat:` specifies the format used to produce the results. For our purposes, this will always be `true`.

Additional Details:

This program writes the output rankings to the command line. In UNIX, you can easily re-direct this output to a (previously non-existent) file by using the ">" operator. The full command would be:

```
indri/indri-5.4/runquery/IndriRunQuery
    corpus/twitter2011/queries.xml
    -index=index/twitter2011_en/
    -count=100 > /path/to/output/file.output
```

Step 2: Computing Evaluation Metrics

Command:

```
script/trec_eval.9.0/trec_eval
    corpus/twitter2011/qrels.txt
    /path/to/output/file.output
    -q
```

Parameters:

(1) `(QRELS FILE):` specifies the path to the QRELS file. This file (provided by TREC) contains the relevance judgements for every query-document pair that made it into the pool. In the case, of the microblog track, 0=non-relevant, 1=marginally relevant, and 2=highly relevant. Not every query-document pair appears in this file. That is because not every query-document pair made into the pool when the pooling process was done. Any query-document pair not in this file is considered non-relevant by TRECEVAL.

(2) `(RANKINGS FILE):` specifies the path to the file that contains the rankings in TRECFORMAT.

(3) `-q:` specifies that you want metrics computed for every query in addition to aggregate metrics average across queries.

Additional Details:

This program writes the output rankings to the command line. In UNIX, you can easily re-direct this output to a (previously non-existent) file by using the ">" operator. The full command would be:

```
script/trec_eval.9.0/trec_eval
     corpus/twitter2011/qrels.txt
     /path/to/output/file.output
     -q > /path/to/results/file.results
```