



Dipasree Pal

Indian Statistical Institute, Kolkata

December 22, 2010

Overview

- ▶ Introduction
- ▶ Installing and running Terrier
- ▶ Terrier's main components
- ▶ Extending terrier
- ▶ Terrier's modules
- ▶ Important links

Introduction

- ▶ What is Terrier?
 - ▶ An Open Source Search Engine written in JAVA.
- ▶ Why Terrier?
 - ▶ Open Source Search Engine.
 - ▶ Well documented.
 - ▶ Most of the significant retrieval models are implemented
 - ▶ Many more
(<http://terrier.org/docs/v2.2.1/overview.html>).

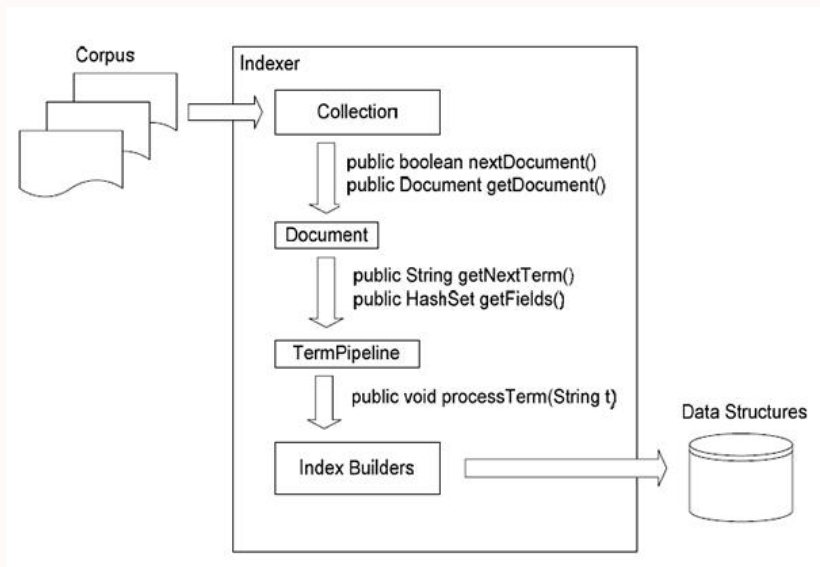
Installing and running Terrier

- ▶ Installation
- ▶ Indexing
- ▶ Retrieval
- ▶ Evaluation

Installation

- ▶ Download
- ▶ Unzip and Untar
- ▶ Correct JAVA version
- ▶ More details
(<http://terrier.org/docs/v2.2.1/quickstart.html>)

Indexing



Terrier's main components

Indexing

- ▶ Collection
Generates Document objects for each next document requested from the collection.
- ▶ Documents
Responsible for parsing and tokenising a document.
- ▶ Termpipeline
Can transform terms (stemmer) or removes terms (stopword) that should not be indexed.
- ▶ Indexer
Responsible for managing the indexing process.
- ▶ Builder
Builds data structures.

Indexing

- ▶ Necessary files
 - ▶ Document collection
 - ▶ List of properties
 - ▶ Stopword list
- ▶ Options
 - i
 - ▶ -l: ponte-croft language model

Terrier's main components

Data Structures

- ▶ Lexicon
(termid, term, df, cf)
-printlexicon

Terrier's main components

Data Structures

- ▶ Lexicon
(termid, term, df, cf)
-printlexicon
- ▶ Document Index
(Docid, Docname, Doclength)
-printdocid

Terrier's main components

Data Structures

- ▶ Lexicon
(termid, term, df, cf)
-printlexicon
- ▶ Document Index
(Docid, Docname, Doclength)
-printdocid
- ▶ Direct Index
(term, freq)
-printdirect

Terrier's main components

Data Structures

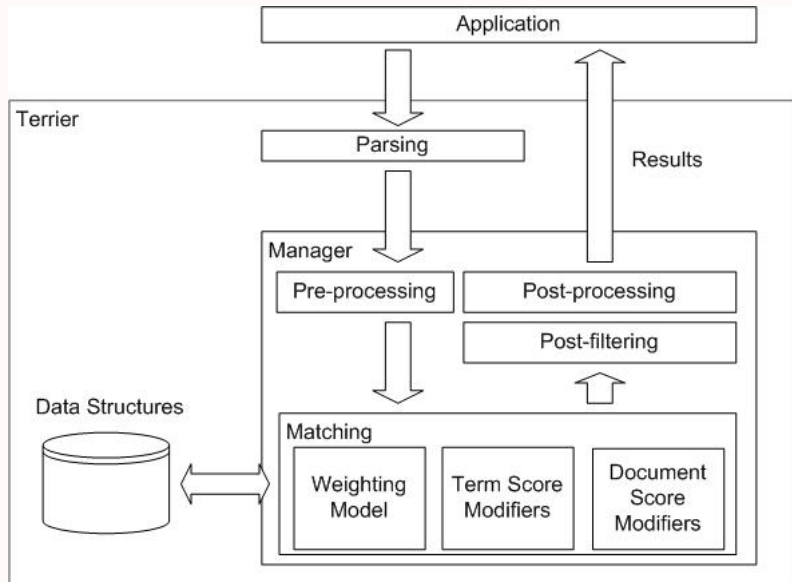
- ▶ Lexicon
(termid, term, df, cf)
-printlexicon
- ▶ Document Index
(Docid, Docname, Doclength)
-printdocid
- ▶ Direct Index
(term, freq)
-printdirect
- ▶ Inverted Index
tid(docid, freq)
-printinverted

Terrier's main components

Applications

- ▶ Trec Terrier
- ▶ Interactive Terrier
- ▶ Desktop Terrier

Retrieval



Terrier's main components

Retrieval

- ▶ Query
Models a query, that consists of subqueries and query terms.
- ▶ Manager
Handling/co-ordinating the main high-level operations of a query
- ▶ Matching
Responsible for determining which documents match and for scoring documents.
- ▶ Weighting Model
Represents the retrieval model that is used to weight the terms of a document.
- ▶ Document Score Modifier
Responsible for query dependent modification document scores.
- ▶ Term Score Modifiers
Modifies the scores of the documents for a given set of pointers, or postings.

Retrieval

- ▶ Necessary files
 - ▶ Properties file
 - ▶ Topics
 - ▶ Models
- ▶ Options
 - r
 - ▶ -c: parameter
 - ▶ -q: query expansion
 - ▶ -l: ponte-croft language model

Evaluation

- ▶ Necessary files
 - ▶ Qrels
- ▶ Options
 - e
 - ▶ -p: per query
 - ▶ filename.res: evaluate a particular result file

Extending Terrier

- ▶ Modification
 - ▶ Make
- ▶ Add new class
 - ▶ share/Manifest

Terrier code

Main packages

- ▶ Application: main application related modules
- ▶ Compression: modules that compress the index files
- ▶ Evaluation: evaluation related modules
- ▶ Indexing: modules that handle indexing issues
- ▶ Matching: modules that compute document score w.r.t a query
- ▶ Querying: query parser and retrieval related modules
- ▶ Sorting: sorting algorithms
- ▶ Structures: data structures managing modules
- ▶ Terms: termpipeline related modules
- ▶ Utility: some utility modules like `terrier_timer`

Important links

- ▶ homepage: <http://terrier.org/>
- ▶ Properties:
<http://terrier.org/docs/v2.2.1/properties.html>
- ▶ Forum: <http://terrier.org/forum/>
- ▶ Wiki: <http://ir.dcs.gla.ac.uk/wiki/Terrier>

THANK YOU!