Introduction to Information Retrieval

1. seminar

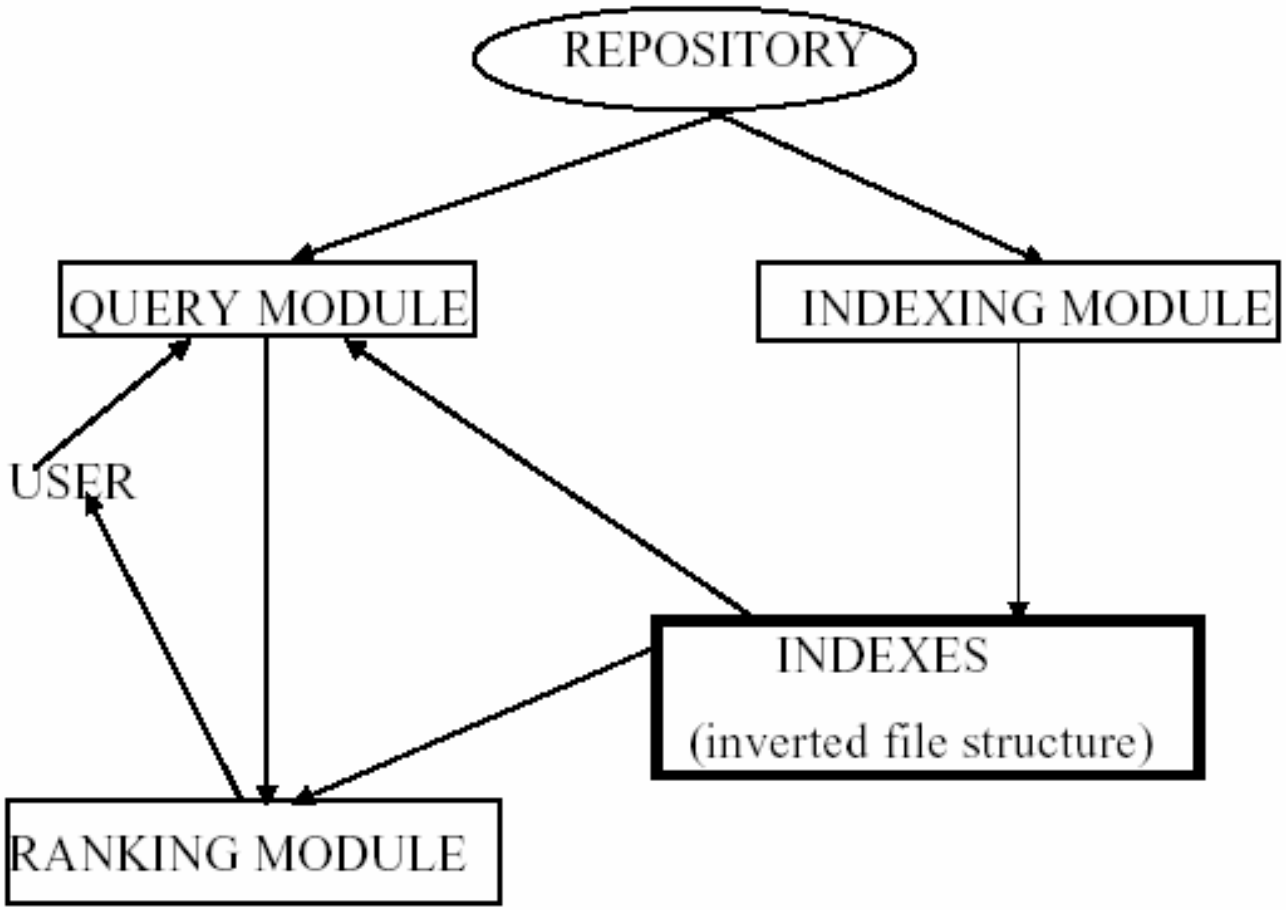
IR architecture, document processing, indexing, weighting

University of Pannonia

Tamás Kiezer, Miklós Erdélyi

Review (1)

- IR architecture overview



Review (2)

- Document processing workflow
  - Parsing
  - Tokenization
  - Stopword removal
  - Stemming
  - Inverted file building (indexing)

Parsing

- Stored information available in diverse formats (HTML, PDF, DOC, etc.)
- Must convert them to a „canonical” format (ie. plain text)
- Many open source tools are available to do parsing in practice
  - NekoHTML, pdftohtml, PDFBox, wvWare, etc.
- Metadata (DCMI)
- Examples

Tokenization (segmentation)

- Chopping the document unit up into pieces called *tokens*
- Language-specific (needs language identification)
- How do we recognize word boundaries?
  - -, /, ., ?, !, …
  - eg. by non-alphanumeric characters
- How do we handle numbers? (index size!)
- Non-trivial for eastern languages like Japanese, Chinese, etc.
- Examples

Stoplisting (1)

- Idea: too frequent or too rare words do not convey useful information
  - Throw away these words during preprocessing using a stoplist
- Example English stoplist:

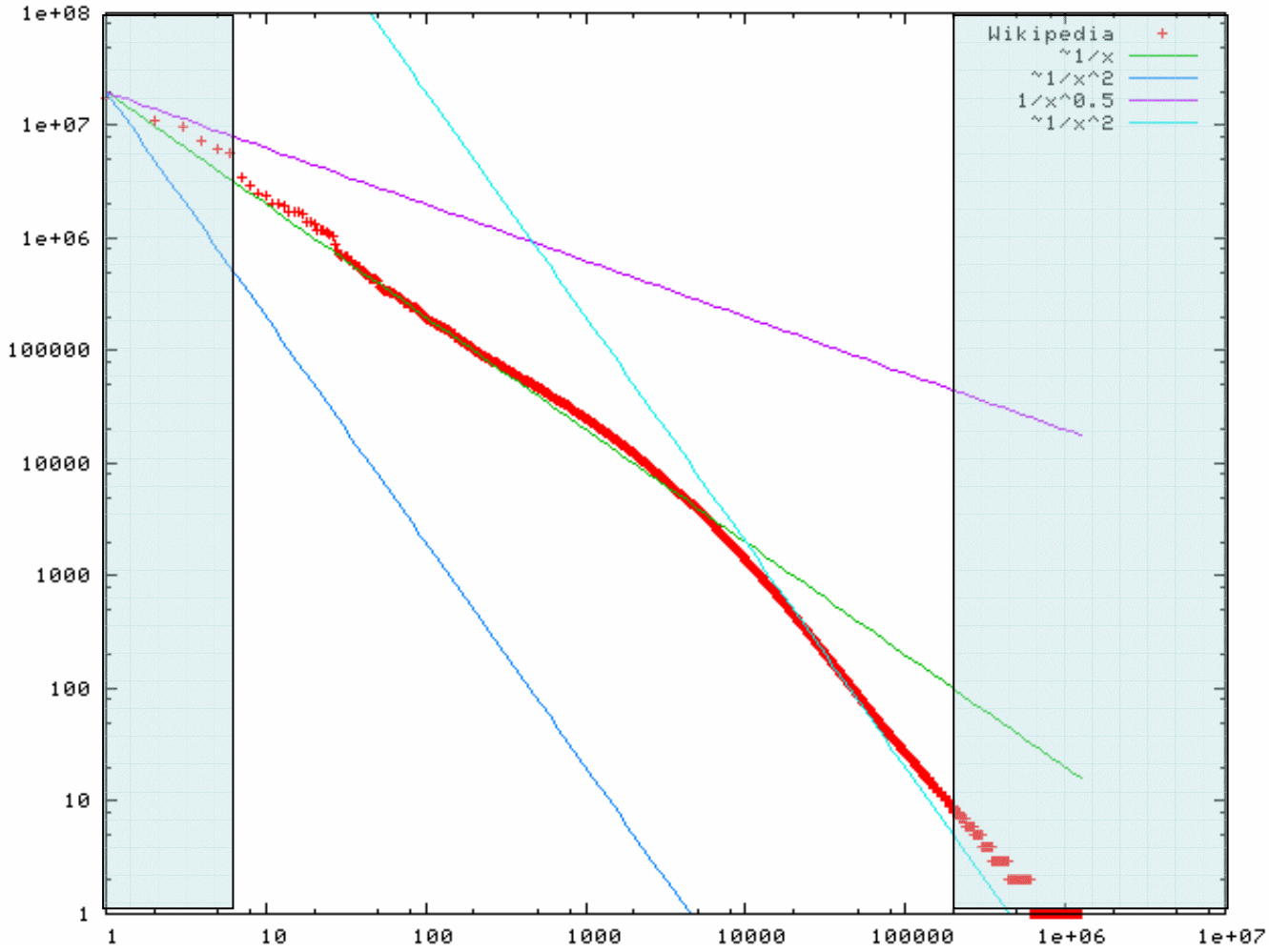
a ab about above ac according across ads ae af after afterwards against albeit all almost alone along already also although always among amongst an and another any anybody anyhow anyone

…

with within without worse worst would wow www x y ye year yet yippee you your yours yourself yourselves

Stoplisting (2)

- Automatized generation of a stoplist: from the word frequency distribution



Stemming

- Idea: reduce lexicon size, improve retrieval efficiency
- Language-specific methods
  - Properly handling agglutinative languages such as Hungarian is difficult
- Stemming methods
  - Brute force, lemmatization, suffix stripping, affix stripping
- Over-stemming, under-stemming
- Normalization (equivalence classing of terms)

Stemming – Porter's method

- Suffix stripping method
- Well-tried for stemming English texts
- 4-step algorithm
  - Step 1 deals with plurals and past participles.
  - Step 2-3 removes adjective/noun formative syllables.
  - Step 4 removes noun formative syllables.
  - Step 5 tidies up.
- Example

Example: Porter's stemming rules (excerpt)

```
Step 2

    (m>0) ATIONAL ->  ATE          relational     ->  relate
    (m>0) TIONAL  ->  TION         conditional    ->  condition
                                   rational       ->  rational
    (m>0) ENCI    ->  ENCE         valenci        ->  valence
    (m>0) ANCI    ->  ANCE         hesitanci      ->  hesitance
    (m>0) IZER    ->  IZE          digitizer      ->  digitize
    (m>0) ABLI    ->  ABLE         conformabli    ->  conformable
    (m>0) ALLI    ->  AL           radicalli      ->  radical
    (m>0) ENTLI   ->  ENT          differentli    ->  different
    (m>0) ELI     ->  E            vileli       - >  vile
    (m>0) OUSLI   ->  OUS          analogousli    ->  analogous
    (m>0) IZATION ->  IZE          vietnamization ->  vietnamize
```

Example: Hunspell for stemming Hungarian text (too)

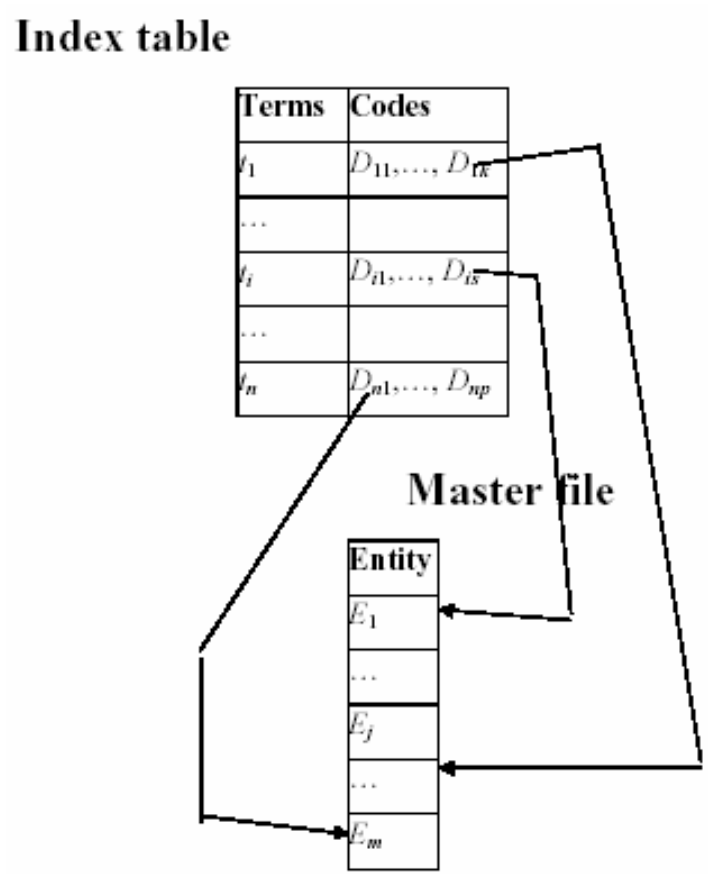
- Hunspell: general library for morphological analysis and stemming
- Affix stripper (does prefix and suffix stripping) with a dictionary of base words
- Example rules:

Rule:	Example:
Z -> Z	ráz -> rázz
Z -> ZÁL	ráz -> rázzál
E D Z -> ENEK	edz -> edzenek, pedz -> pedzenek
[ÓUÚ] -> VAL	manó -> manóval, hamu -> hamuval, bú -> búval
[^AE] -> NAK	okos -> okosnak

Inverted file structure – review

- Stores the postings list for each term
- Eases answering queries - how?

Terms in alphabetical order	Codes of documents in which the term occurs
t_1	$D_{11},\ldots,D_{1k}$
…	
t_i	$D_{i1},\ldots,D_{is}$
…	
t_n	$D_{n1},\ldots,D_{np}$



Inverted index construction

- Example:

Doc 1
I did enact Julius Caesar: I was killed i' the Capitol; Brutus killed me.

Doc 2
So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious:

term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

$\Longrightarrow$

term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

$\Longrightarrow$

term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
I	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Weighting methods – review

- Binary weighting: $w_{ij} = \begin{cases} 1 & if \quad t_i \quad occurs\ in \quad D_j \\ 0 & otherwise \end{cases}$
- Frequency weighting: $w_{ij} = f_{ij}$.
- Max-normalized (*max-tf*): $w_{ij} = \dfrac{f_{ij}}{\max\limits_{1 \le k \le n} f_{kj}}$
- Length-normalized (*norm-tf*): $w_{ij} = \dfrac{f_{ij}}{\sqrt{\sum_{k=1}^{n} f_{kj}^2}}$
- Term frequency inverse document frequency

$$w_{ij} = f_{ij} \times \left(\log \frac{m}{F_i} \right)$$

- Length normalized term frequency inverse document frequency (*norm-tf-idf*):

$$w_{ij} = \frac{f_{ij} \times \left(\log \dfrac{m}{F_i} \right)}{\sqrt{\sum_{k=1}^{n} \left(f_{kj} \times \left(\log \dfrac{m}{F_k} \right) \right)^2}}$$

Exercise: building a TD matrix

- Let us consider the following simple document collection:

Doc 1	breakthrough drug for schizophrenia
Doc 2	new schizophrenia drug
Doc 3	new approach for treatment of schizophrenia
Doc 4	new hopes for schizophrenia patients

- Build a frequency weighted TD matrix
- Build a norm-tf weighted TD matrix
- Build a norm-tf-idf weighted TD matrix

Solution: tf weighted TD matrix

Terms/Documents	Doc1	Doc2	Doc3	Doc4
approach	0	0	1	0
breakthrough	1	0	0	0
drug	1	1	0	0
hope	0	0	0	1
new	0	1	1	1
patient	0	0	0	1
schizophrenia	1	1	1	1
treatment	0	0	1	0

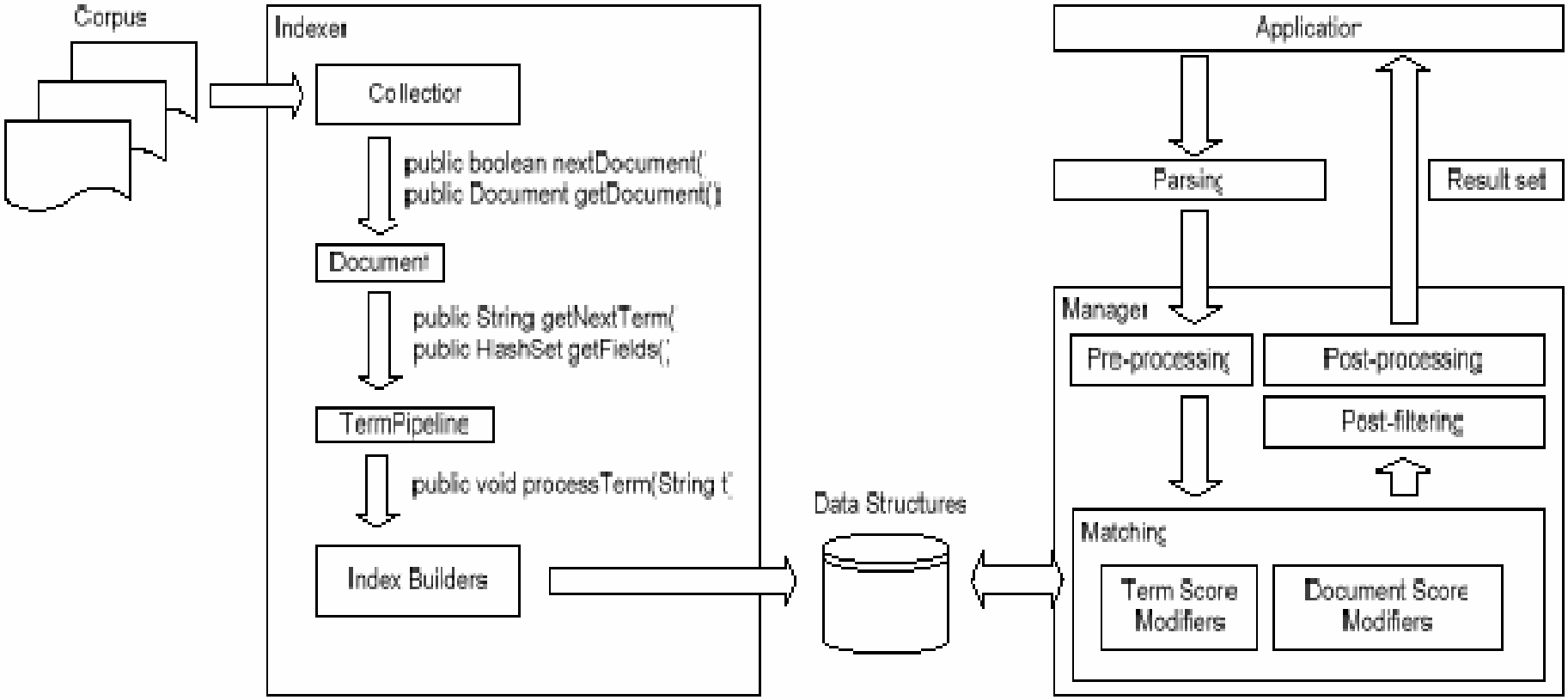
Solution: norm-tf weighted TD matrix

Terms/Documents	Doc1	Doc2	Doc3	Doc4
approach	0	0	0,5	0
breakthrough	0,57735	0	0	0
drug	0,57735	0,57735	0	0
hope	0	0	0	0,5
new	0	0,57735	0	0,5
patient	0	0	0	0,5
schizophrenia	0,57735	0,57735	0	0,5
treatment	0	0	0	0

Example: Terrier IR Platform

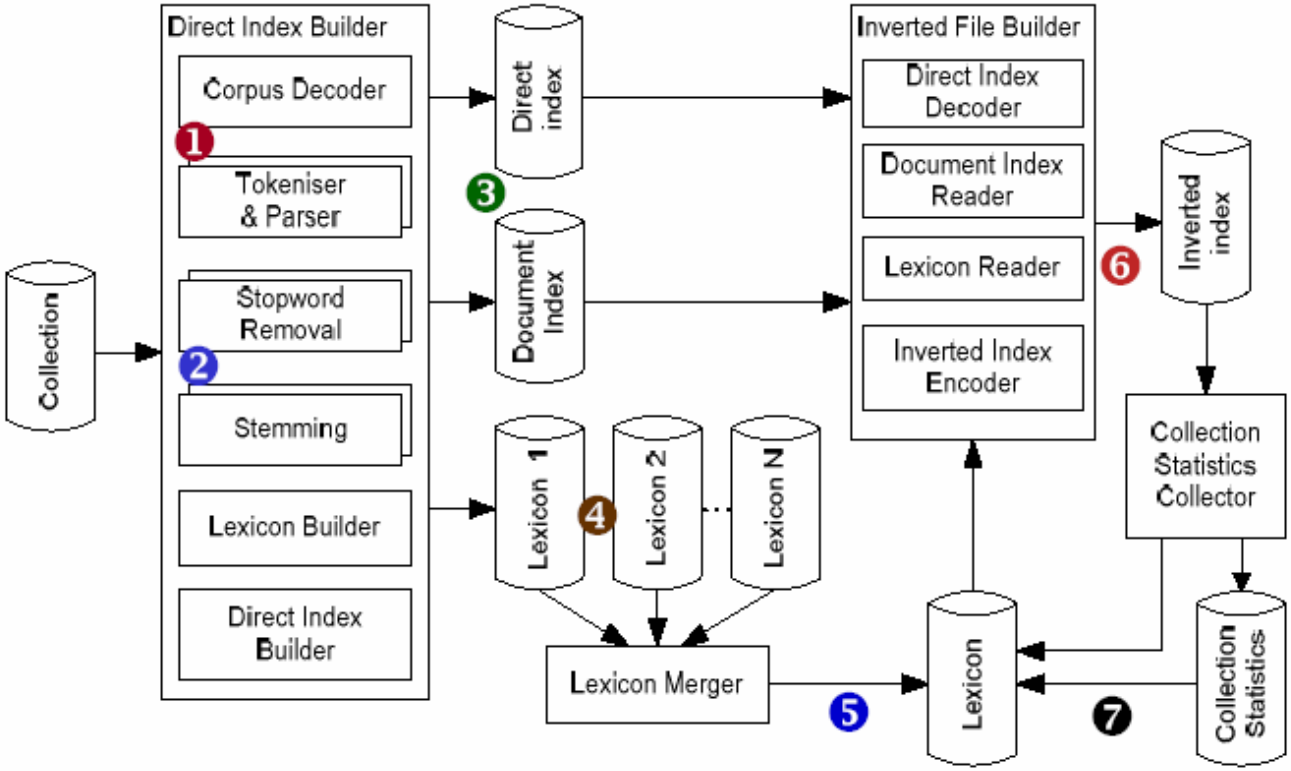
Indexing API

Querying API



Terrier: Indexing

1. Tokenise & parse documents
2. Application-dependent stopword removal and stemming
3. Build the direct and document indices
4. Use less memory by building lexicons for parts of the collection
5. Merge temporary lexicons to form one lexicon
6. Build the inverted index from the existing data structures
7. Update collection statistics and lexicon



Terrier: Search results

Search Results

Search results for **information retrieval** Page 1 of 20 (Showing 1 to 10 of 200 Results)

1. **Glasgow Information Retrieval Group**

http://ir.dcs.gla.ac.uk/

The **Information Retrieval** Group led by Professor Keith van Rijsbergen has a vigorous programme of research, based on both theory and experiment, aimed at developing novel, effective, and efficient

2. **Joemon Jose's web page**

http://www.dcs.gla.ac.uk/~jj/

Examples of recent work include the development of implicit feedback techniques for personalized **retrieval**, adaptive image **retrieval**, and architectures for peer to peer **information retrieval**

3. **Ali Azimi Bolourian Information Retrieval Dept. of Computing Science - University of Glasgow**

http://www.dcs.gla.ac.uk/~azimi/

I'm a PhD student at the **Information Retrieval** research group within the Department of Computing Science at the University of Glasgow, under supervision of Professor Keith van Rijsbergen .

4. **Hideo Joho - Information Retrieval (IR), Glasgow University**

http://www.dcs.gla.ac.uk/~hideo/

In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in **Information Retrieval** , to appear, Amsterdam, Netherlands: ACM.

Questions?