Introduction

Information Retrieval

Indian Statistical Institute

Course details

Books

- **[MRS]** *Introduction to Information Retrieval*, Manning, Raghavan, Schütze.
  `https://nlp.stanford.edu/IR-book/`
- **[BCC]** *Information Retrieval Implementing and Evaluating Search Engines*, Büttcher, Clarke, Cormack.
  `http://www.ir.uwaterloo.ca/book/`
- **[CMS]** *Search Engines: Information Retrieval in Practice*, Croft, Metzler, Strohman.
  `http://www.search-engines-book.com/`
- Foundations and Trends in Information Retrieval (FTIR)
  `https://www.nowpublishers.com/INR`

Weightage: Mid-sem 20% Project 30% End-sem 50%

Slides: Available from
`http://www.isical.ac.in/~mandar/courses.html` and
`http://www.isical.ac.in/~debapriyo`

Terminology

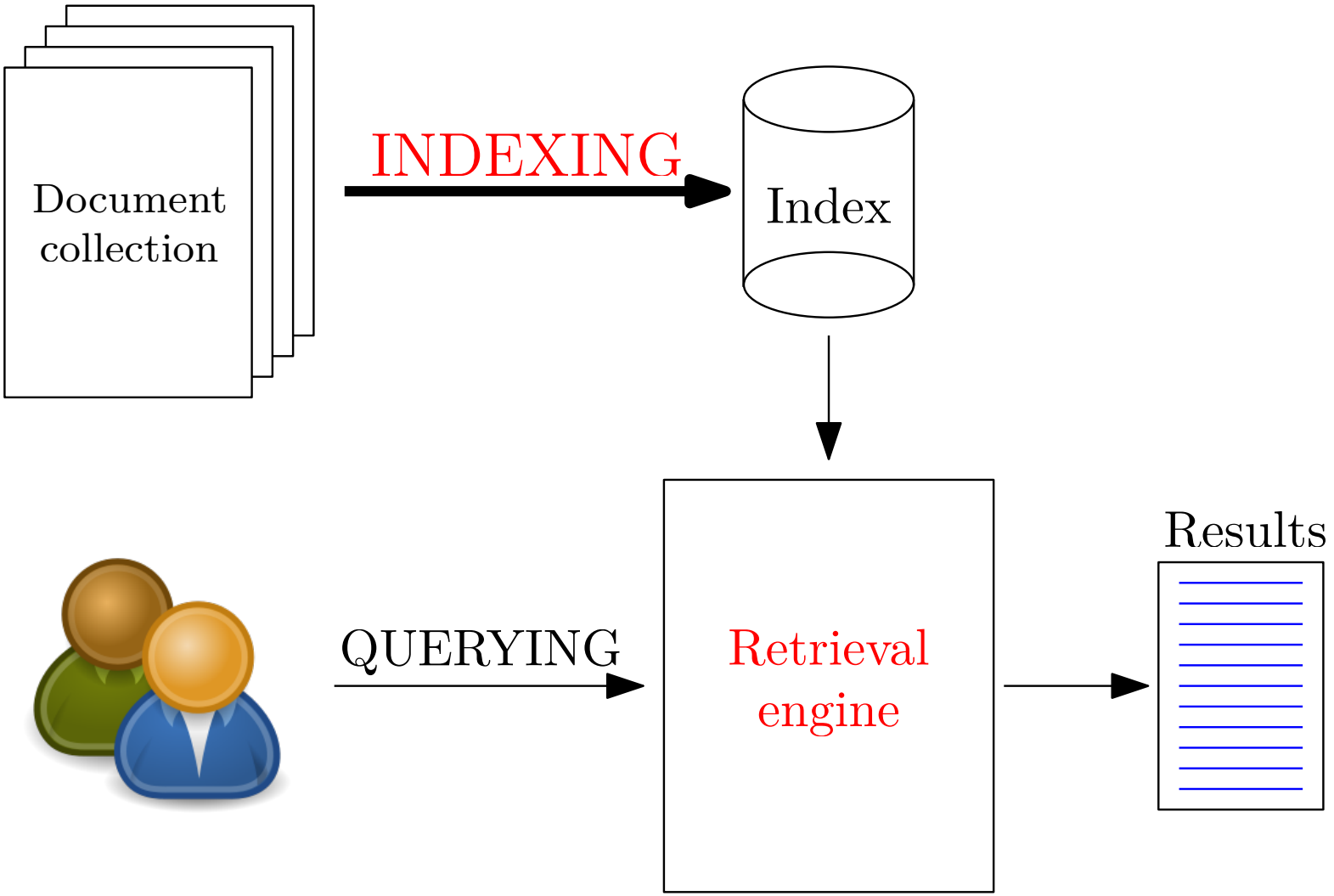
Problem definition:

Given a user's *information need*, find *documents* satisfying that need.

- *Information need:* what user is looking for
- *Query:* actual representation of above
- *Document:* any unit / item that can be retrieved

For this course, we will only consider textual information
(no images/graphics, maps, speech, video, etc.).

Overview


Document collection
INDEXING
Index
QUERYING
Retrieval engine
Results

Steps

1. **Document acquisition:** how is the document collection obtained / constructed? (LATER)
2. **Indexing:** representing documents so that retrieval is easy
3. **Retrieval:** matching the user query against documents in the collection
4. **Evaluation:** how to determine whether the system did well? (NEXT WEEK)

Bag of words approach

- **Indexing:**
  - document $\rightarrow$ list of keywords / content-descriptors / *terms*
  - user's information need $\rightarrow$ (natural-language) query $\rightarrow$ list of keywords
- **Retrieval:** measure overlap between query and documents.

Indexing

1. Tokenisation
2. Stopword removal
3. Stemming
4. Phrase identification
5. Named entity extraction

Indexing – I

Tokenisation: identify individual words.

Information retrieval (IR) is the activity of obtaining information resources relevant to an information need from a collection of information resources. Searches can be based on full-text or other content-based indexing.

$$\Downarrow$$

Information retrieval IR is the activity of obtaining ...

Indexing – II

Stopword removal: eliminate common words

Information retrieval IR is the activity of obtaining ...

Indexing – III

- Stemming: reduce words to a common root.
  - e.g. resignation, resigned, resigns $\rightarrow$ resign
  - for common languages, use standard algorithms (Porter).

Indexing – IV

Phrases: multi-word terms e.g. computer science, data mining.

- Syntactic/linguistic methods
  - use a part of speech tagger
  - look for particular POS sequences, e.g., NN NN, JJ NN
    Example: computer/NN science/NN

Indexing – IV

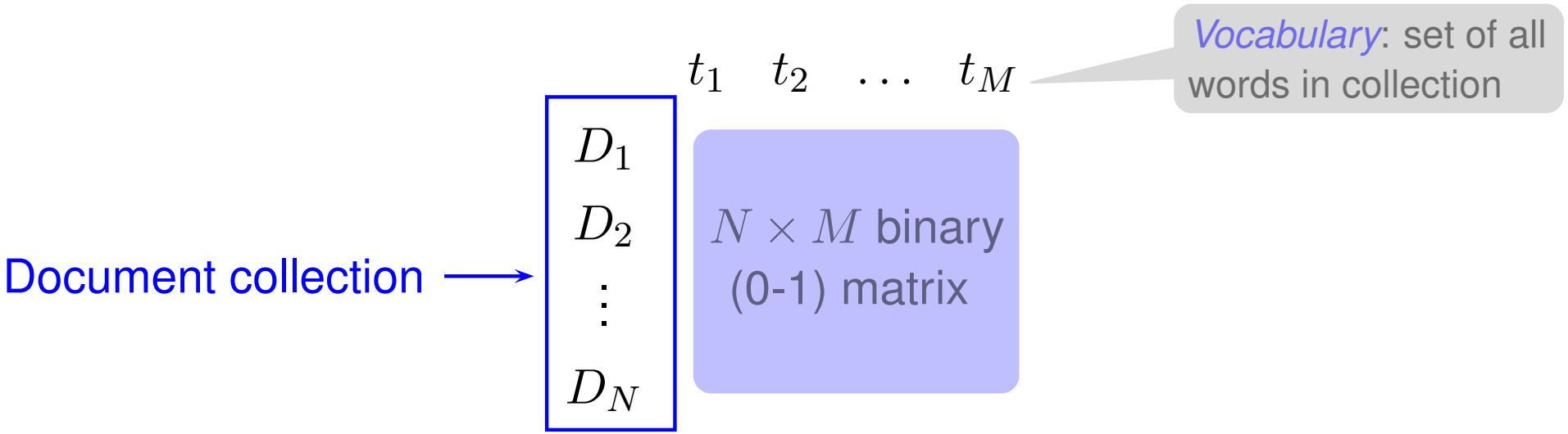
- Statistical methods: $f_{(a,b)} > \theta$ (threshold)
  - Raw frequency: $f_{raw}(a,b) = n_{(a,b)}$
  - Dice coefficient:

$$f_{dice}(a,b) = 2 \times n_{(a,b)}/(n_a + n_b)$$

    n_a, n_b number of bi-grams whose first (second) word is a (b)
  - ...

Indexing

Document collection $\rightarrow$ *Term-Document Matrix*



Retrieval models

Boolean model

- Keywords combined using AND, OR, (AND) NOT
  e.g. (medicine OR treatment) AND (hypertension OR “high blood pressure”)

Boolean model

- Keywords combined using AND, OR, (AND) NOT
  e.g. (medicine OR treatment) AND (hypertension OR "high blood pressure")

- Efficient and easy to implement (list merging)
  - AND $\equiv$ intersection
    OR $\equiv$ union
  - Example:
    medicine $\rightarrow D_1, D_4, D_5, D_{10}, \ldots$
    hypertension $\rightarrow D_2, D_4, D_8, D_{10}, \ldots$

Boolean model

- Keywords combined using AND, OR, (AND) NOT
  e.g. (medicine OR treatment) AND (hypertension OR "high blood pressure")

- Efficient and easy to implement (list merging)
  - AND $\equiv$ intersection
    OR $\equiv$ union
  - Example:
    medicine $\rightarrow D_1, D_4, D_5, D_{10}, \ldots$
    hypertension $\rightarrow D_2, D_4, D_8, D_{10}, \ldots$

- Drawbacks
  - OR — one match as good as many
    AND — one miss as bad as all
  - no ranking
  - queries may be difficult to formulate

Vector space model (VSM)

- Any text item (“document”) is represented as list of terms and associated weights.

	t_1	t_2	$\ldots$	t_M
D_1	w_{11}	w_{12}		w_{1M}
D_2	w_{21}	w_{22}		w_{2M}
$\vdots$				
D_N	w_{N1}	w_{N2}		w_{NM}

- Term = keywords or content-descriptors
- Weight = measure of the importance of a term in representing the information contained in the document

Term weights

- Term frequency (tf)
  - repeated words are strongly related to content
  - importance does not grow linearly with frequency
    $\Rightarrow$ use sub-linear function
  - examples:

$$1 + \log(tf), \quad 1 + \log\big(1 + \log(tf)\big), \quad \frac{tf}{k + tf}$$

- Inverse document frequency (idf): uncommon term is more important
  Example: medicine vs. antibiotic
  - commonly used functions

$$\log \frac{N}{1 + df}, \quad \log \frac{N - df + 0.5}{df + 0.5}$$

Term weights

- Normalisation by document length: term-weights for long documents should be reduced
  - long docs. contain many distinct words.
  - long docs. contain same word many times.
  - Intuition: each term covers a smaller portion of the overall information content of a long document
  - use # bytes, # distinct words, Euclidean length, etc.
- Weight = tf x idf / normalisation

Term weights: “traditional” weighting schemes

- Cosine normalisation

$$\frac{(1+\log(tf)) \times \log \frac{N}{1+df}}{\sqrt{\sum w_i^2}}$$

- Pivoted normalisation

$$\frac{\frac{1+\log(tf)}{1+\log(average\ tf)} \times \log(\frac{N}{df})}{(1.0-slope) \times pivot + slope \times \#\ unique\ terms}$$

VSM: retrieval

- Measure vocabulary overlap between user query and documents.

$$
\begin{array}{rcccc}
 & & t_1 & \ldots & t_M \\
Q & = & q_1 & \ldots & q_M \\
D & = & d_1 & \ldots & d_M \\
Sim(Q,D) & = & \vec{Q}.\vec{D} & & \\
 & = & \sum_i q_i \times d_i & &
\end{array}
$$

  - more matches between $Q, D \Rightarrow Sim(Q,D) \uparrow$
  - matches on *important* terms between $Q, D \Rightarrow Sim(Q,D) \uparrow$

VSM: retrieval

- Measure vocabulary overlap between user query and documents.

$$\begin{aligned} & & t_1 & \ldots & t_M \\ Q & = & q_1 & \ldots & q_M \\ D & = & d_1 & \ldots & d_M \\ Sim(Q,D) & = & \vec{Q}.\vec{D} & & \\ & = & \sum_i q_i \times d_i & & \end{aligned}$$

  - more matches between $Q, D \Rightarrow Sim(Q,D) \uparrow$
  - matches on *important* terms between $Q, D \Rightarrow Sim(Q,D) \uparrow$
- Use inverted list (index).

$$t_i \rightarrow (D_{i_1}, w_{i_1}), \ldots, (D_{i_k}, w_{i_k})$$