

# Indexing and Retrieval: Basics

Mandar Mitra

---

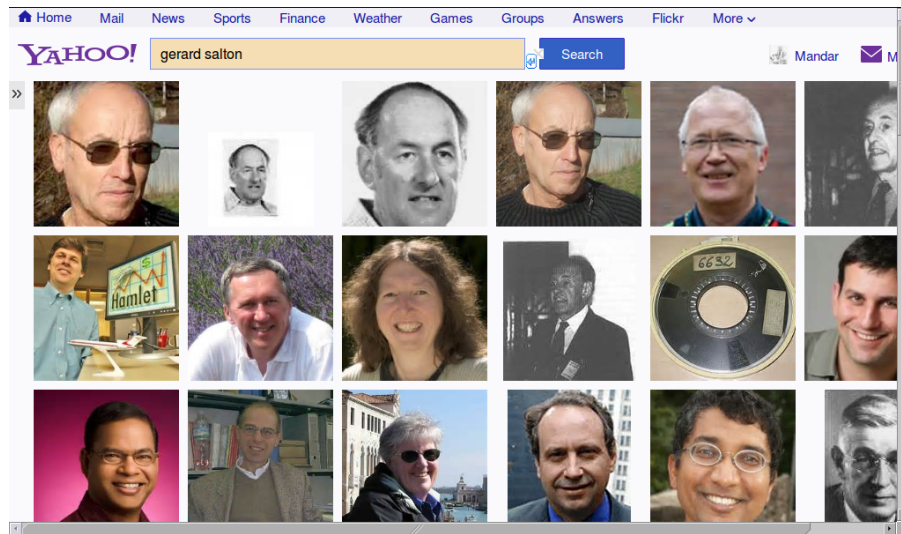
## Information retrieval

The screenshot shows a Google search interface with the query "information retrieval tutorial slides". The search results are displayed on the "Web" tab. The results include:

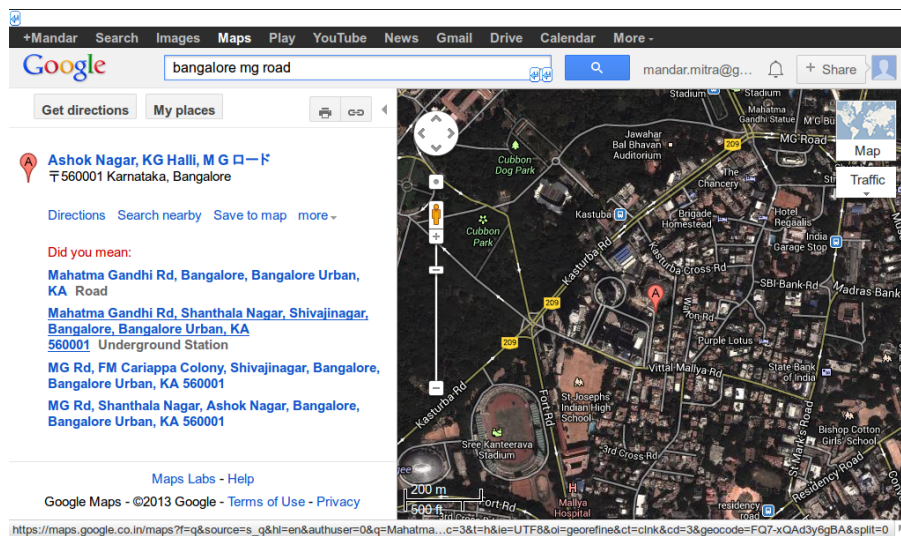
- About 3,810,000 results (0.32 seconds)
- Scholarly articles for information retrieval tutorial slides**
  - [DB&IR: both sides now](#) - Weikum - Cited by 47
  - [... Indexing of Interdisciplinary Collections of Slides and ...](#) - Simons - Cited by 26
  - [... by exploiting presentation slide information for ...](#) - Kawahara - Cited by 19
- [PPT] Introduction to Information Retrieval - Department of C...**
  - [www.cs.uiuc.edu/class/fa07/cs411/lectures/cs411-07-maryam.ppt](#)
  - Documents. Query. Formulation. Resource. query reformulation,. relevance feedback. Slide is from Jimmy Lin's tutorial. 10. Introduction to Information Retrieval.
- Slides from the ECIR'12 "Quantum Information Access and ...**
  - [www.bpiwovar.net/.../slides-from-the-ecir12-quantum-information-acce...](#)
  - Apr 2, 2012 - Hi Benjamin, I congratulate you for this amazing recopilation of concepts and ideas on the relation between the quantum formalism and IR.
- Introduction to Information Retrieval: Slides - The Stanford ...**
  - [nlp.stanford.edu/IR-book/news/slides.html](#)
  - 20+ items - Introduction to Information Retrieval: Slides. Powerpoint slides ...
  - 02 The term vocabulary & postings lists
  - 08 Evaluation in information retrieval
- Crowdsourcing for Information Retrieval: Principles, Method...**
  - [www.slideshare.net/.../crowdsourcing-for-information-retrieval-principle...](#)
  - Tutorial by Omar Alonso and Matthew Lease, presented July 24, 2011 at

---

## Information retrieval



## Information retrieval



## Information retrieval

- *What's the best smartphone under Rs.10,000?*
- *I'm free this evening. How do I entertain myself?*
- ...

## Information retrieval

**Problem definition:**

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

- Types of information: text, images/graphics, speech, video, etc.
  - Text is still the most commonly used.
- 

**IR: bag of words approach**

- Document → list of keywords / content-descriptors / *terms*
  - User's information need → (natural-language) query → list of keywords
  - Measure overlap between query and documents.
- 

**Indexing****Tokenization: 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.



Information retrieval IR is the activity of obtaining ...

---

**Indexing: tokenization with NLTK****Getting started**

```
1 import nltk
2 from nltk.book import * # for existing corpora
```

**Tokenization I**

```
1 from nltk import word_tokenize
2 with open('filename.txt') as fp:
3     text = fp.read()
4     tokenlist = word_tokenize(text)
```

---

## Indexing: tokenization with NLTK

### Tokenization II

```
1 from nltk.corpus import PlaintextCorpusReader
2 corpus_root = './data'
3 filelist = PlaintextCorpusReader(corpus_root, '.*\.txt')
4 # filelist.fileids() gives ['file1.txt', 'file2.txt']
5 # filelist.words('file1.txt') gives [u'Reason', u'for', ...]
```

---

## Indexing: stopword removal

### Eliminate common words

Information retrieval IR is the activity of obtaining ...

### Stopword removal in NLTK

```
1 from nltk.corpus import stopwords
2 stoplist = stopwords.words('english') # [u'i', u'me', u'my', ...]
3 filtered = [ w.lower() for w in filelist.words('file1.txt')
4              if w.isalnum()
5              and w.lower() not in stoplist ]
```

---

## Indexing: stemming

- Stemming: reduce words to a common root.
  - e.g. resignation, resigned, resigns → resign
  - use standard algorithms (Porter).

### Stemming in NLTK

```
1 porter = nltk.PorterStemmer()
2 stemmed = [ porter.stem(w) for w in filtered ]
3 index_terms = sorted(set(stemmed))
```

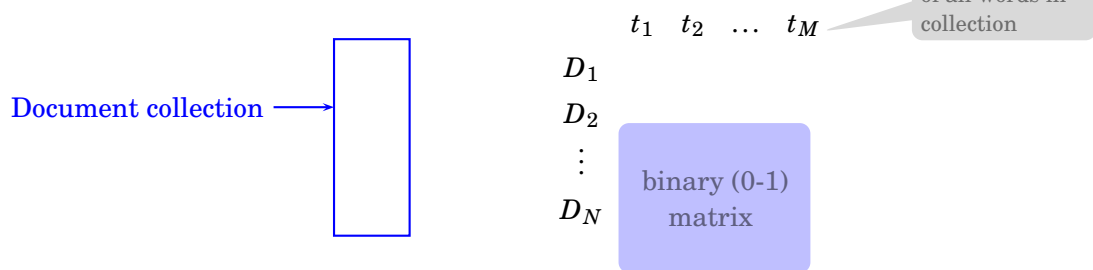
---

## Indexing

- Thesaurus: find synonyms for words in the document.
  - Phrases: find multi-word terms e.g. computer science, data mining.
    - use syntax/linguistic methods or “statistical” methods.
  - Named entities: identify names of people, organizations and places; dates; monetary or other amounts, etc.
- 

## IR: basic principle

- Document → list of keywords / content-descriptors / *terms*
- Document collection → *Term-Document Matrix*



- User's information need → (natural-language) query → list of keywords
- Measure overlap between query and documents.

## 1 Models

### 1.1 Boolean model

---

#### 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  $\rightarrow D_1, D_4, D_5, D_{10}, \dots$  hypertension  $\rightarrow D_2, D_4, D_8, D_{10}, \dots$

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

## 1.2 Vector space model

---

### Vector space model

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

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

- 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
  - Inverse document frequency (idf): uncommon term is more important  
Example: medicine vs. antibiotic
  - Normalization by document length
    - long docs. contain many distinct words.
    - long docs. contain same word many times.
    - term-weights for long documents should be reduced.
    - use # bytes, # distinct words, Euclidean length, etc.
  - Weight = tf x idf / normalization
-

## Term weights: commonly used weighting schemes

- Pivoted normalization [Singhal et al., SIGIR 96]

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

- BM25 (probabilistic model) [Robertson and Zaragoza, FTIR 2009]

$$\frac{tf \times \log(\frac{N-df+0.5}{df+0.5})}{k_1((1-b) + b \frac{dl}{avdl}) + tf}$$

---

## Retrieval

- Measure vocabulary overlap between user query and documents.

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

- Use inverted list (index).

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