Introduction to Information Retrieval

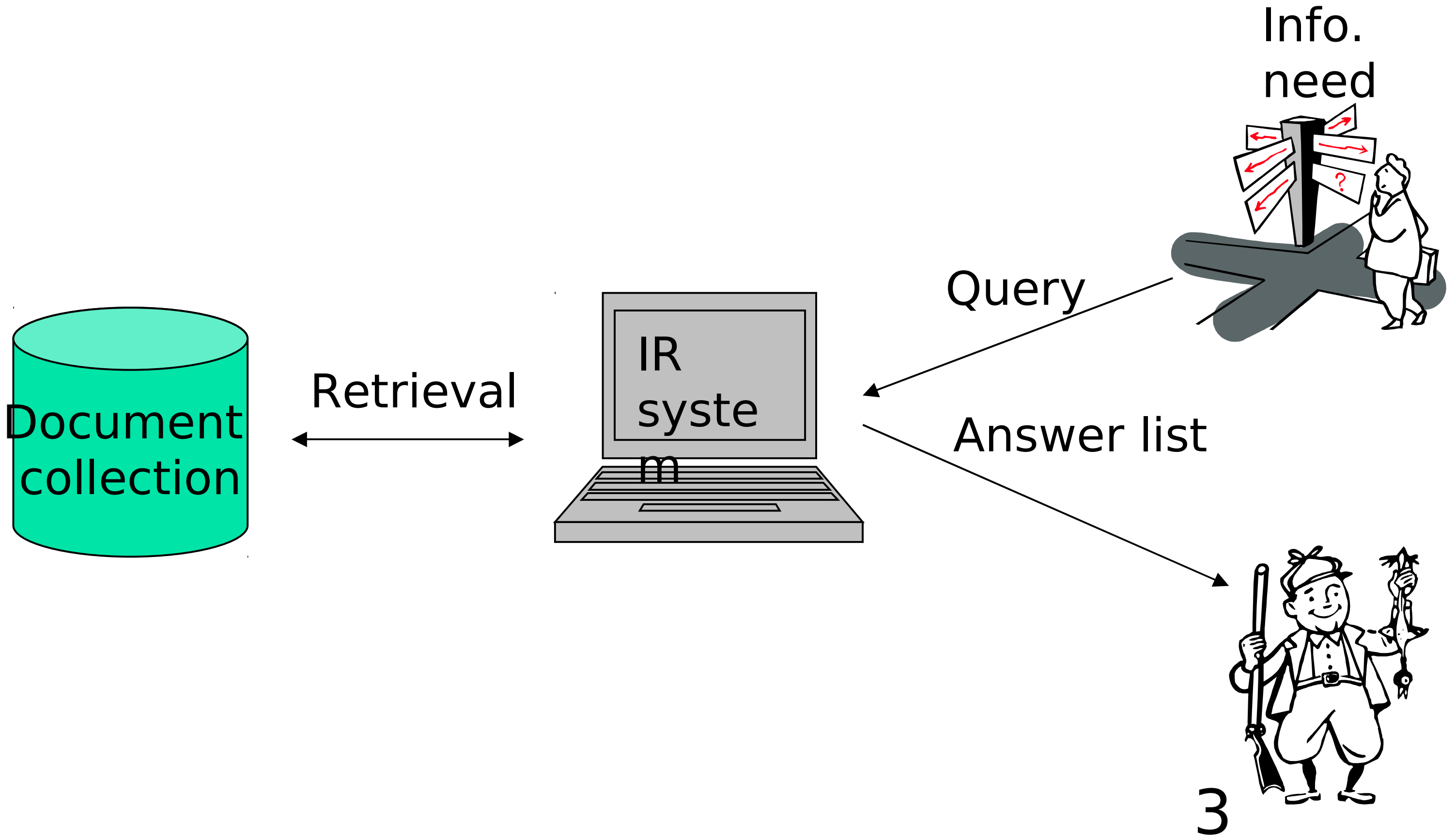
Jian-Yun Nie
University of Montreal
Canada

Outline

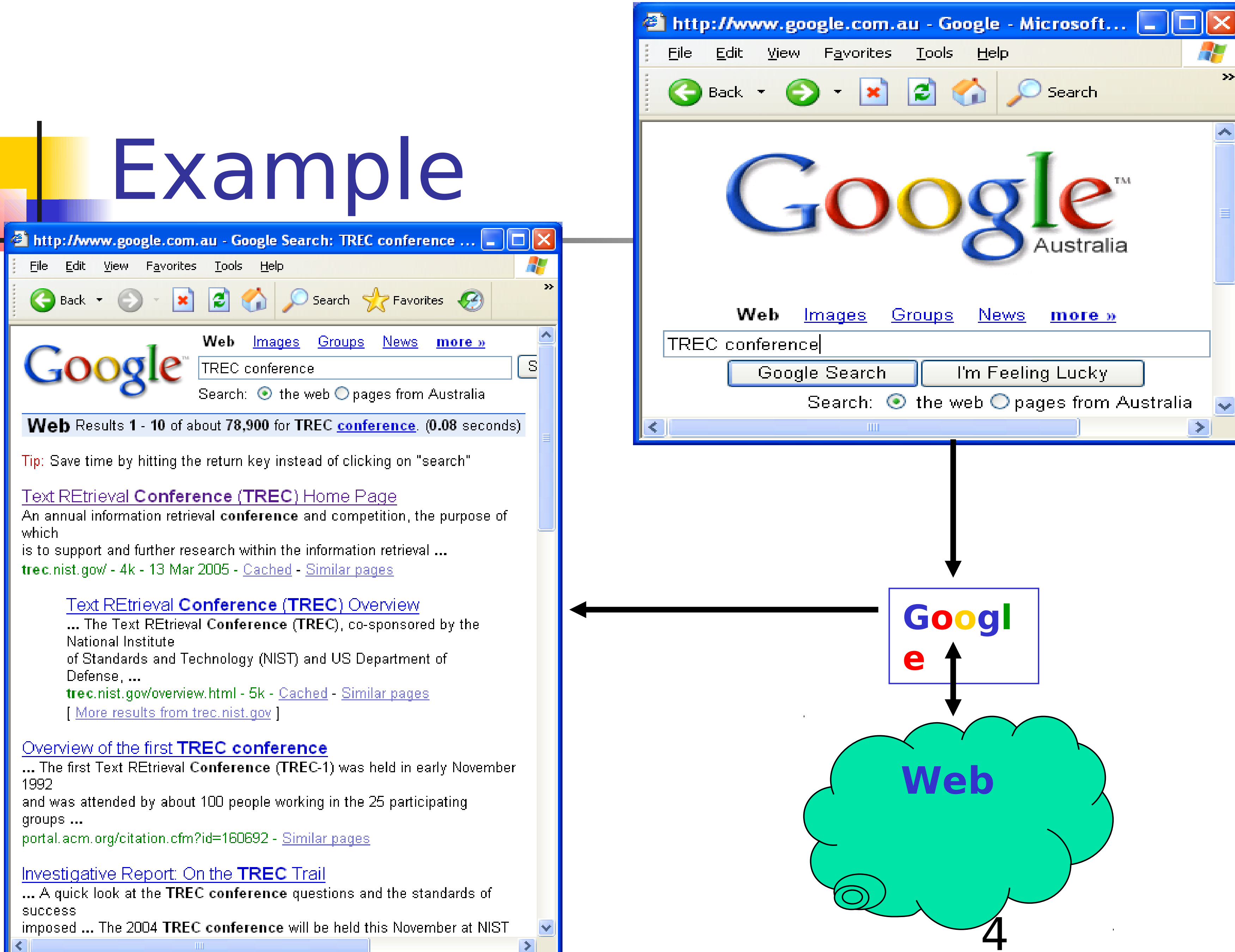
- What is the IR problem?
- How to organize an IR system? (Or the main processes in IR)
- Indexing
- Retrieval
- System evaluation
- Some current research topics

The problem of IR

- **Goal** = find documents *relevant* to an information need from a large document set



Example



http://www.google.com.au - Google - Microsoft...
File Edit View Favorites Tools Help
Back Search
Google™
Australia
Web Images Groups News more »
TREC conference
Google Search I'm Feeling Lucky
Search: the web pages from Australia
http://www.google.com.au - Google Search: TREC conference ...
File Edit View Favorites Tools Help
Back Search Favorites
Google
Web Images Groups News more »
TREC conference
Search: the web pages from Australia
Web Results 1 - 10 of about 78,900 for TREC conference. (0.08 seconds)
Tip: Save time by hitting the return key instead of clicking on "search"
Text REtrieval Conference (TREC) Home Page
An annual information retrieval conference and competition, the purpose of which
is to support and further research within the information retrieval ...
trec.nist.gov/ - 4k - 13 Mar 2005 - Cached - Similar pages
Text REtrieval Conference (TREC) Overview
... The Text REtrieval Conference (TREC), co-sponsored by the National Institute
of Standards and Technology (NIST) and US Department of Defense, ...
trec.nist.gov/overview.html - 5k - Cached - Similar pages
[More results from trec.nist.gov]
Overview of the first TREC conference
... The first Text REtrieval Conference (TREC-1) was held in early November 1992
and was attended by about 100 people working in the 25 participating groups ...
portal.acm.org/citation.cfm?id=160692 - Similar pages
Investigative Report: On the TREC Trail
... A quick look at the TREC conference questions and the standards of success
imposed ... The 2004 TREC conference will be held this November at NIST
Google
Web

IR problem

- First applications: in libraries (1950s)

  ISBN: 0-201-12227-8

  Author: Salton, Gerard

  Title: Automatic text processing: the transformation, analysis, and retrieval of information by computer

  Editor: Addison-Wesley

  Date: 1989

  Content: <Text>

- external attributes and internal attribute (content)
- Search by external attributes = Search in DB
- IR: search by content

Possible approaches

1. String matching (linear search in documents)

   - Slow

   - Difficult to improve

2. Indexing (*)

   - Fast

   - Flexible to further improvement

Indexing-based IR

Document Query

indexing **indexing**

(Query analysis)

Representation Representation

(keywords) **Query** (keywords)

evaluation

Main problems in IR

- Document and query indexing
  - How to best represent their contents?
- Query evaluation (or retrieval process)
  - To what extent does a document correspond to a query?
- System evaluation
  - How good is a system?
  - Are the retrieved documents relevant? (precision)
  - Are all the relevant documents retrieved? (recall)

Document indexing

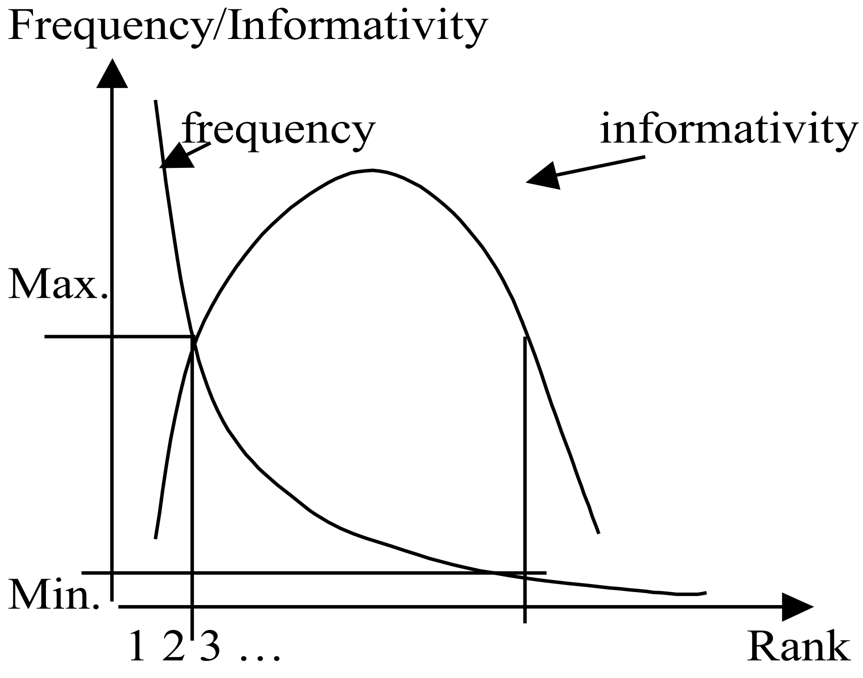
- Goal = Find the important meanings and create an internal representation
- Factors to consider:
  - Accuracy to represent meanings (semantics)
  - Exhaustiveness (cover all the contents)
  - Facility for computer to manipulate
- What is the best representation of contents?
  - Char. string (char trigrams): not precise enough
  - Word: good coverage, not precise
  - Phrase: poor coverage, more precise
  - Concept: poor coverage, precise

Coverage (Recall) ⟷ **Accuracy (Precision)**

String **Word** **Phrase** **Concept**

Keyword selection and weighting

- How to select *important* keywords?
  - Simple method: using middle-frequency words



tf*idf weighting schema

- tf = term frequency
  - frequency of a term/keyword in a document

  The higher the tf, the higher the importance (weight) for the doc.
- df = document frequency
  - no. of documents containing the term
  - distribution of the term
- idf = inverse document frequency
  - the unevenness of term distribution in the corpus
  - the specificity of term to a document

  The more the term is distributed evenly, the less it is specific to a document

$$\text{weight}(t,D) = \text{tf}(t,D) * \text{idf}(t)$$

Some common *tf*idf* schemes

- tf(t, D)=freq(t,D)
- tf(t, D)=log[freq(t,D)]
- tf(t, D)=log[freq(t,D)]+1
- tf(t, D)=freq(t,d)/Max[f(t,d)]

idf(t) = log(N/n)

n = #docs containing t

N = #docs in corpus

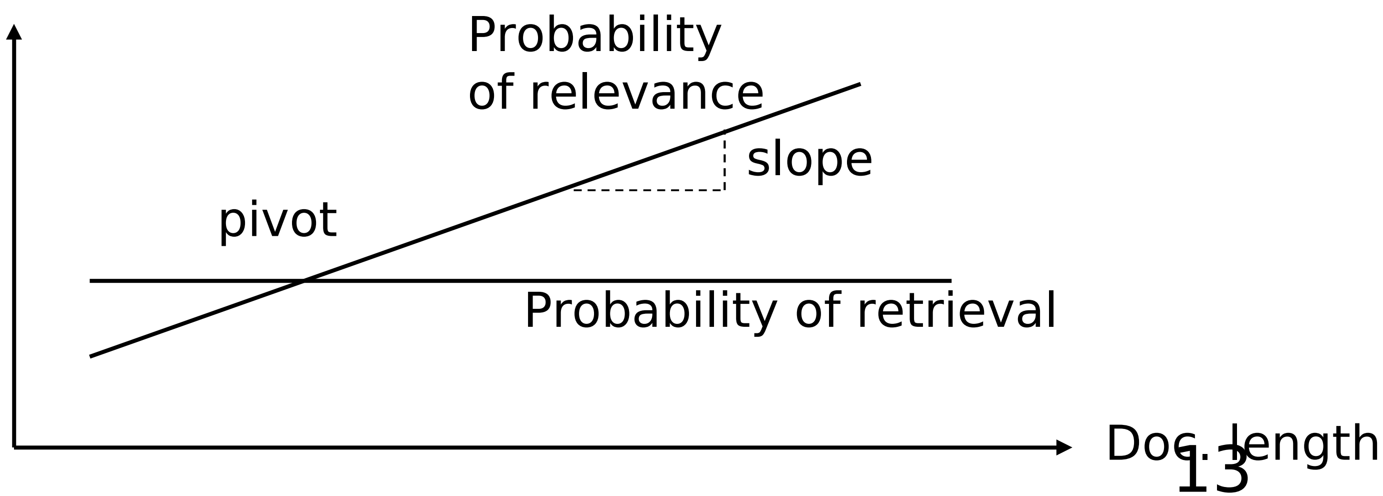
weight(t,D) = tf(t,D) * idf(t)

- Normalization: Cosine normalization, /max, …

Document Length Normalization

- Sometimes, additional normalizations e.g. length:

$$pivoted(t,D) = \frac{weight(t,D)}{1 + \frac{slope}{(1 - slope) \times povot} normalized\_weight(t,D)}$$



Stopwords / Stoplist

- function words do not bear useful information for IR
  of, in, about, with, I, although, …
- Stoplist: contain stopwords, not to be used as index
  - Prepositions
  - Articles
  - Pronouns
  - Some adverbs and adjectives
  - Some frequent words (e.g. document)

- The removal of stopwords usually improves IR effectiveness
- A few “standard” stoplists are commonly used.

Stemming

- Reason:
  - Different word forms may bear similar meaning (e.g. search, searching): create a “standard” representation for them
- Stemming:
  - Removing some endings of word

    computer
    compute
    computes
    computing
    computed
    computation

    comput

Porter algorithm

(Porter, M.F., 1980, An algorithm for suffix stripping, *Program*, **14**(3) :130-137)

- Step 1: plurals and past participles
  - SSES -> SS caresses -> caress
  - (*v*) ING -> motoring -> motor
- Step 2: adj->n, n->v, n->adj, ...
  - (m>0) OUSNESS -> OUS callousness -> callous
  - (m>0) ATIONAL -> ATE relational -> relate
- Step 3:
  - (m>0) ICATE -> IC triplicate -> triplic
- Step 4:
  - (m>1) AL -> revival -> reviv
  - (m>1) ANCE -> allowance -> allow
- Step 5:
  - (m>1) E -> probate -> probat
  - (m > 1 and *d and *L) -> single letter controll -> control

Lemmatization

- transform to standard form according to syntactic category.

  E.g. verb + *ing* $\rightarrow$ verb

  noun + *s* $\rightarrow$ noun

  - Need POS tagging
  - More accurate than stemming, but needs more resources

- crucial to choose stemming/lemmatization rules

  noise v.s. recognition rate
- compromise between precision and recall

light/no stemming severe stemming

-recall +precision +recall -precision

Result of indexing

- Each document is represented by a set of weighted keywords (terms):

  $D_1 \rightarrow \{(t_1, w_1), (t_2, w_2), \ldots\}$

  e.g. $D_1 \rightarrow \{(\text{comput}, 0.2), (\text{architect}, 0.3), \ldots\}$

  $D_2 \rightarrow \{(\text{comput}, 0.1), (\text{network}, 0.5), \ldots\}$

- Inverted file:

  $\text{comput} \rightarrow \{(D_1, 0.2), (D_2, 0.1), \ldots\}$

  Inverted file is used during retrieval for higher efficiency.

Retrieval

- The problems underlying retrieval
  - Retrieval model
    - How is a document represented with the selected keywords?
    - How are document and query representations compared to calculate a score?
  - Implementation

Cases

- 1-word query:

  The documents to be retrieved are those that include the word

  - Retrieve the inverted list for the word
  - Sort in decreasing order of the weight of the word
- Multi-word query?
  - Combining several lists
  - How to interpret the weight?

  (IR model)

IR models

- Matching score model
  - Document D = a set of weighted keywords
  - Query Q = a set of non-weighted keywords
  - $R(D, Q) = \Sigma_i \, w(t_i , D)$

    where t_i is in Q.

Boolean model

- Document = Logical conjunction of keywords
- Query = Boolean expression of keywords
- $R(D, Q) = D \rightarrow Q$

e.g. $D = t_1 \wedge t_2 \wedge \ldots \wedge t_n$

$Q = (t_1 \wedge t_2) \vee (t_3 \wedge \neg t_4)$

$D \rightarrow Q$, thus $R(D, Q) = 1$.

Problems:

- R is either 1 or 0 (unordered set of documents)
- many documents or few documents
- End-users cannot manipulate Boolean operators correctly

  E.g. documents about *kangaroos* **and** *koalas*

Extensions to Boolean model (for document ordering)

- $D = \{\ldots, (t_i, w_i), \ldots\}$: weighted keywords
- **Interpretation:**
  - D is a member of class t_i to degree w_i.
  - In terms of fuzzy sets: $\mu_{ti}(D) = w_i$

A possible **Evaluation:**

$R(D, t_i) = \mu_{ti}(D);$

$R(D, Q_1 \wedge Q_2) = \min(R(D, Q_1), R(D, Q_2));$

$R(D, Q_1 \vee Q_2) = \max(R(D, Q_1), R(D, Q_2));$

$R(D, \neg Q_1) = 1 - R(D, Q_1).$

Vector space model

- Vector space = all the keywords encountered

  $<t_1, \quad t_2, \quad t_3, \ldots, t_n>$
- Document

  $D = \qquad < a_1, a_2, a_3, \ldots, a_n>$

  a_i = weight of t_i in D
- Query

  $Q = \qquad < b_1, b_2, b_3, \ldots, b_n>$

  b_i = weight of t_i in Q
- R(D,Q) = Sim(D,Q)

Matrix representation

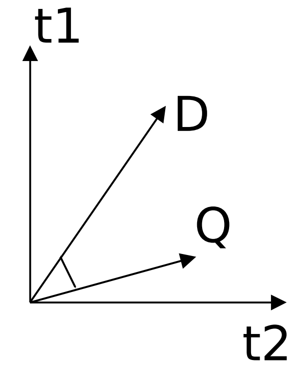
Document space

Term vector space

$$
\begin{array}{c|ccccc|}
 & t_1 & t_2 & t_3 & \dots & t_n \\
\hline
D_1 & a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\
D_2 & a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\
D_3 & a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\
\dots & & & & & \\
D_m & a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \\
\hline
\multicolumn{1}{c}{Q} & \multicolumn{1}{c}{b_1} & \multicolumn{1}{c}{b_2} & \multicolumn{1}{c}{b_3} & \multicolumn{1}{c}{\dots} & \multicolumn{1}{c}{b_n}
\end{array}
$$

Some formulas for Sim

Dot product $$Sim(D,Q) = \sum (a_i * b_i)$$



Cosine $$Sim(D,Q) = \frac{\sum_i (a_i * b_i)}{\sqrt{\sum_i a_i^2 * \sum_i b_i^2}}$$

Dice $$Sim(D,Q) = \frac{2\sum_i (a_i * b_i)}{\sum_i a_i^2 + \sum_i b_i^2}$$

Jaccard $$Sim(D,Q) = \frac{\sum_i (a_i * b_i)}{\sum_i a_i^2 + \sum_i b_i^2 - \sum_i (a_i * b_i)}$$

Implementation (space)

- Matrix is very sparse: a few 100s terms for a document, and a few terms for a query, while the term space is large (~100k)

- Stored as:

  $D1 \rightarrow \{(t1, a1), (t2,a2), \ldots\}$

  $t1 \rightarrow \{(D1,a1), \ldots\}$

Implementation (time)

- The implementation of VSM with dot product:
  - Naïve implementation: O(m*n)
  - Implementation using inverted file:

  Given a query = {(t1,b1), (t2,b2)}:

  1. find the sets of related documents through inverted file for t1 and t2
  2. calculate the score of the documents to each weighted term
     (t1,b1) $\rightarrow$ {(D1,a1 *b1), …}
  3. combine the sets and sum the weights ($\sum$)

  - O(|Q|*n)

Other similarities

Cosine:

$$Sim(D,Q) = \frac{\sum_i (a_i * b_i)}{\sqrt{\sum_j a_i^2 * \sum_j b_i^2}} = \sum_i \frac{a_i}{\sqrt{\sum_j a_j^2}} \frac{b_i}{\sqrt{\sum_j b_j^2}}$$

- use $\sqrt{\sum_j a_j^2}$ and $\sqrt{\sum_j b_j^2}$ to normalize the weights after indexing
- Dot product

(Similar operations do not apply to Dice and Jaccard)

Probabilistic model

- Given D, estimate P(R|D) and P(NR|D)
- P(R|D)=P(D|R)*P(R)/P(D)   (P(D), P(R) constant)

  $\propto$ P(D|R)

  D = {t_1=x_1, t_2=x_2, …} $x_i = \begin{cases} 1 & present \\ 0 & absent \end{cases}$

- $P(D \mid R) = \prod_{(t_i = x_i) \in D} P(t_i = x_i \mid R)$

$$= \prod_{t_i} P(t_i = 1 \mid R)^{x_i} P(t_i = 0 \mid R)^{(1-x_i)} = \prod_{t_i} p_i^{x_i} (1 - p_i)^{(1-x_i)}$$

$$P(D \mid NR) = \prod_{t_i} P(t_i = 1 \mid NR)^{x_i} P(t_i = 0 \mid NR)^{(1-x_i)} = \prod_{t_i} q_i^{x_i} (1 - q_i)^{(1-x_i)}$$

Prob. model (cont'd)

For document ranking

$$Odd(D) = \log \frac{P(D|R)}{P(D|NR)} = \log \frac{\prod_{t_i} p_i^{x_i} (1-p_i)^{(1-x_i)}}{\prod_{t_i} q_i^{x_i} (1-q_i)^{(1-x_i)}}$$

$$= \sum_{t_i} x_i \log \frac{p_i(1-q_i)}{q_i(1-p_i)} + \sum_{t_i} \log \frac{1-p_i}{1-q_i}$$

$$\propto \sum_{t_i} x_i \log \frac{p_i(1-q_i)}{q_i(1-p_i)}$$

Prob. model (cont'd)

- How to estimate p_i and q_i?
- A set of *N* relevant and irrelevant samples:

$$p_i = \frac{r_i}{R_i} \qquad q_i = \frac{n_i - r_i}{N - R_i}$$

r_i Rel. doc. with t_i	n_i-r_i Irrel.doc. with t_i	n_i Doc. with t_i
R_i-r_i Rel. doc. without t_i	N-R_i–n+r_i Irrel.doc. without t_i	N-n_i Doc. without ti
R_i Rel. doc	N-R_i Irrel.doc.	N Samples

Prob. model (cont'd)

$$Odd(D) = \sum_{t_i} x_i \log \frac{p_i(1-q_i)}{q_i(1-p_i)}$$

$$= \sum_{t_i} x_i \frac{r_i(N-R_i-n_i+r_i)}{(R_i-r_i)(n_i-r_i)}$$

- Smoothing (Robertson-Sparck-Jones formula)

$$Odd(D) = \sum_{t_i} x_i \frac{(r_i+0.5)(N-R_i-n_i+r_i+0.5)}{(R_i-r_i+0.5)(n_i-r_i+0.5)} = \sum_{t_i \in D} w_i$$

- When no sample is available:
  p_i=0.5,
  $q_i=(n_i+0.5)/(N+0.5)\approx n_i/N$
- May be implemented as VSM

BM25

$$Score(D,Q) = \sum_{t \in Q} w \frac{(k_1+1)tf}{K+tf} \frac{(k_3+1)qtf}{k_3+qtf} + k_2 \,|\, Q \,|\, \frac{avdl - dl}{avdl + dl}$$

$$K = k_1((1-b) + b\frac{dl}{avdl - dl})$$

- k1, k2, k3, d: parameters
- qtf: query term frequency
- dl: document length
- avdl: average document length

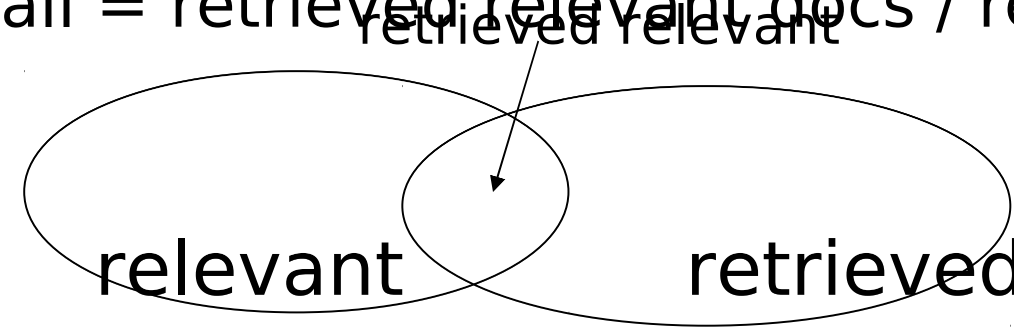
(Classic) Presentation of results

- Query evaluation result is a list of documents, sorted by their similarity to the query.
- E.g.

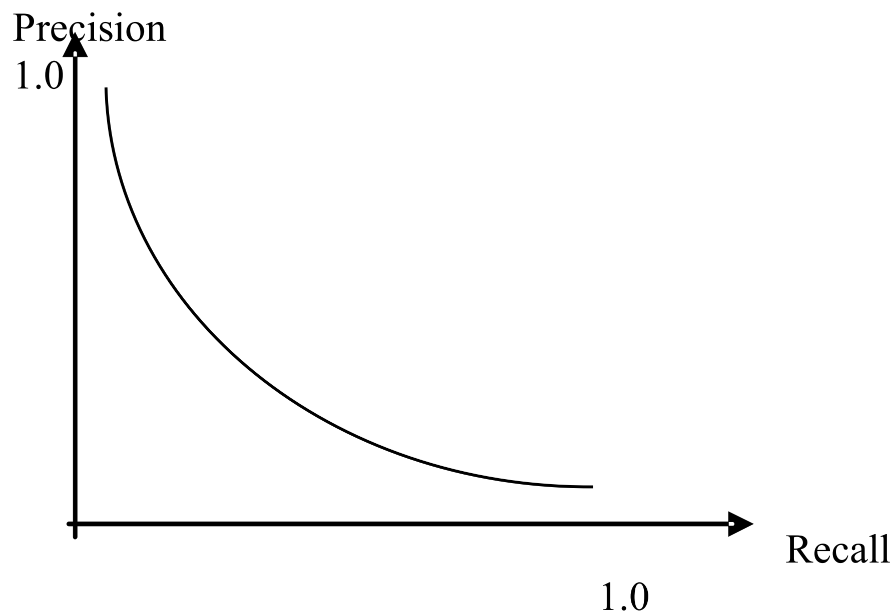
doc1	0.67
doc2	0.65
doc3	0.54
…	

System evaluation

- Efficiency: time, space
- Effectiveness:
  - How is a system capable of retrieving relevant documents?
  - Is a system better than another one?
- Metrics often used (together):
  - Precision = retrieved relevant docs / retrieved docs
  - Recall = retrieved relevant docs / relevant docs



General form of precision/recall



-Precision change w.r.t. Recall (not a fixed point)

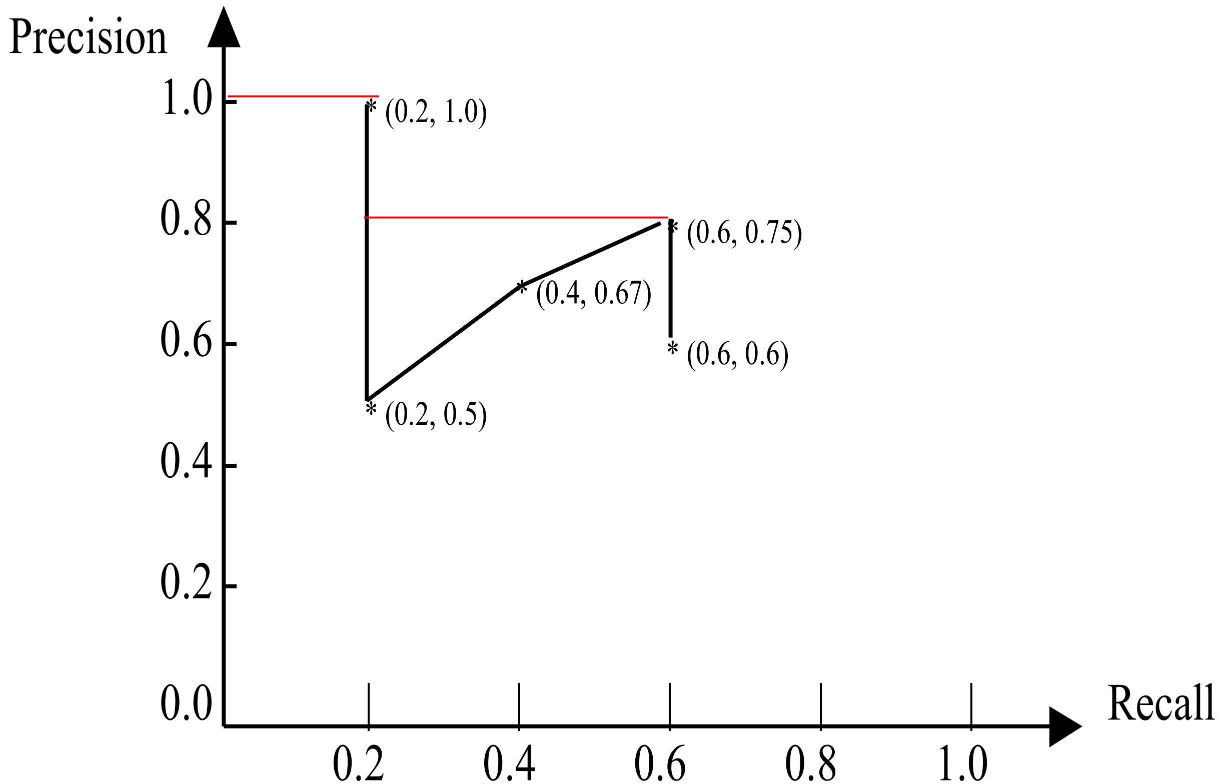
-Systems cannot compare at one Precision/Recall point

-Average precision (on 11 points of recall: 0.0, 0.1, …, 1.0)

An illustration of P/R calculation

List	Rel?
Doc 1	Y
Doc 2	
Doc 3	Y
Doc 4	Y
Doc	

Assume: 5 relevant docs.



MAP (Mean Average Precision)

$$MAP = \frac{1}{n} \sum_{Q_i} \frac{1}{|R_i|} \sum_{D_j \in R_i} \frac{j}{r_{ij}}$$

- r_{ij} = rank of the j-th relevant document for Q_i
- $|R_i|$ = #rel. doc. for Q_i
- n = # test queries
- E.g. Rank:

	1
5	8
10	

4 1st rel. doc.

2nd rel. doc.

3rd rel. doc.

$$MAP = \frac{1}{2}[\frac{1}{3}(\frac{1}{1} + \frac{2}{5} + \frac{3}{10}) + \frac{1}{2}(\frac{1}{4} + \frac{2}{8})]$$

Some other measures

- Noise = retrieved irrelevant docs / retrieved docs
- Silence = non-retrieved relevant docs / relevant docs
  - Noise = 1 – Precision; Silence = 1 – Recall
- Fallout = retrieved irrel. docs / irrel. docs
- Single value measures:
  - F-measure = 2 P * R / (P + R)
  - Average precision = average at 11 points of recall
  - Precision at n document (often used for Web IR)
  - Expected search length (no. irrelevant documents to read before obtaining n relevant doc.)

Test corpus

- Compare different IR systems on the same test corpus
- A test corpus contains:
  - A set of documents
  - A set of queries
  - Relevance judgment for every document-query pair (desired answers for each query)
- The results of a system is compared with the desired answers.

An evaluation example (SMART)

```
Run number:            1        2
Num_queries:          52       52
Total number of documents over all queries
    Retrieved:       780      780
    Relevant:        796      796
    Rel_ret:         246      229
Recall - Precision Averages:
    at 0.00        0.7695   0.7894
    at 0.10        0.6618   0.6449
    at 0.20        0.5019   0.5090
    at 0.30        0.3745   0.3702
    at 0.40        0.2249   0.3070
    at 0.50        0.1797   0.2104
    at 0.60        0.1143   0.1654
    at 0.70        0.0891   0.1144
    at 0.80        0.0891   0.1096
    at 0.90        0.0699   0.0904
    at 1.00        0.0699   0.0904
```

```
Average precision for all points
   11-pt Avg:       0.2859   0.3092
    % Change:                   8.2
Recall:
      Exact:        0.4139   0.4166
    at  5 docs:     0.2373   0.2726
    at 10 docs:     0.3254   0.3572
    at 15 docs:     0.4139   0.4166
    at 30 docs:     0.4139   0.4166
Precision:
      Exact:        0.3154
   0.2936
    At  5 docs:     0.4308   0.4192
    At 10 docs:     0.3538   0.3327
    At 15 docs:     0.3154   0.2936
    At 30 docs:     0.1577   0.1468
```

The TREC experiments

- Once per year
- A set of documents and queries are distributed to the participants (the standard answers are unknown) (April)
- Participants work (very hard) to construct, fine-tune their systems, and submit the answers (1000/query) at the deadline (July)
- NIST people manually evaluate the answers and provide correct answers (and classification of IR systems) (July – August)
- TREC conference (November)

TREC evaluation methodology

- Known document collection (>100K) and query set (50)
- Submission of 1000 documents for each query by each participant
- Merge 100 first documents of each participant -> global pool
- Human relevance judgment of the global pool
- The other documents are assumed to be irrelevant
- Evaluation of each system (with 1000 answers)

  - Partial relevance judgments
  - But stable for system ranking

Tracks (tasks)

- Ad Hoc track: given document collection, different topics
- Routing (filtering): stable interests (user profile), incoming document flow
- CLIR: Ad Hoc, but with queries in a different language
- Web: a large set of Web pages
- Question-Answering: When did Nixon visit China?
- Interactive: put users into action with system
- Spoken document retrieval
- Image and video retrieval
- Information tracking: new topic / follow up

CLEF and NTCIR

- CLEF = Cross-Language Experimental Forum
  - for European languages
  - organized by Europeans
  - Each per year (March – Oct.)
- NTCIR:
  - Organized by NII (Japan)
  - For Asian languages
  - cycle of 1.5 year

Impact of TREC

- Provide large collections for further experiments
- Compare different systems/techniques on realistic data
- Develop new methodology for system evaluation

- Similar experiments are organized in other areas (NLP, Machine translation, Summarization, ...)

Some techniques to improve IR effectiveness

- Interaction with user (relevance feedback)

  - Keywords only cover part of the contents

  - User can help by indicating relevant/irrelevant document

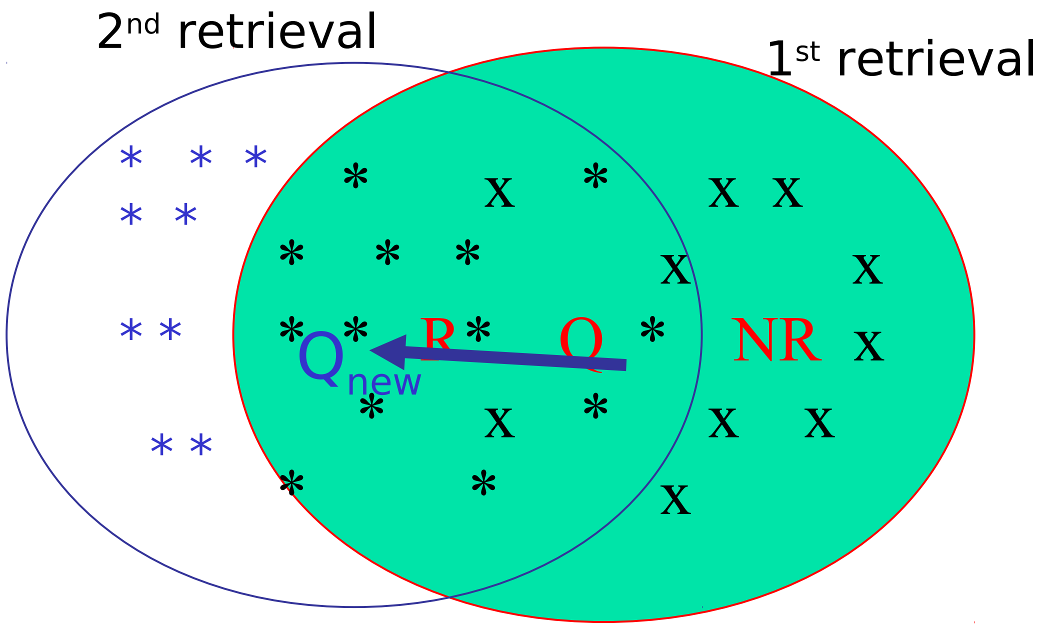
- The use of relevance feedback
  - To improve query expression:

    $$Q_{new} = \alpha * Q_{old} + \beta * Rel\_d - \gamma * Nrel\_d$$

    where Rel_d = centroid of relevant documents

    NRel_d = centroid of non-relevant documents

Effect of RF



Modified relevance feedback

- Users usually do not cooperate (e.g. AltaVista in early years)
- Pseudo-relevance feedback (Blind RF)
  - Using the top-ranked documents as if they are relevant:
    - Select m terms from n top-ranked documents
  - One can usually obtain about 10% improvement

Query expansion

- A query contains part of the important words
- Add new (related) terms into the query
  - Manually constructed knowledge base/thesaurus (e.g. Wordnet)
    - Q = information retrieval
    - Q' = (information + data + knowledge + …)
      (retrieval + search + seeking + …)
  - Corpus analysis:
    - two terms that often co-occur are related (Mutual information)
    - Two terms that co-occur with the same words are related (e.g. T-shirt and coat with wear, …)

Global vs. local context analysis

- Global analysis: use the whole document collection to calculate term relationships
- Local analysis: use the query to retrieve a subset of documents, then calculate term relationships
  - Combine pseudo-relevance feedback and term co-occurrences
  - More effective than global analysis

Some current research topics:

Go beyond keywords

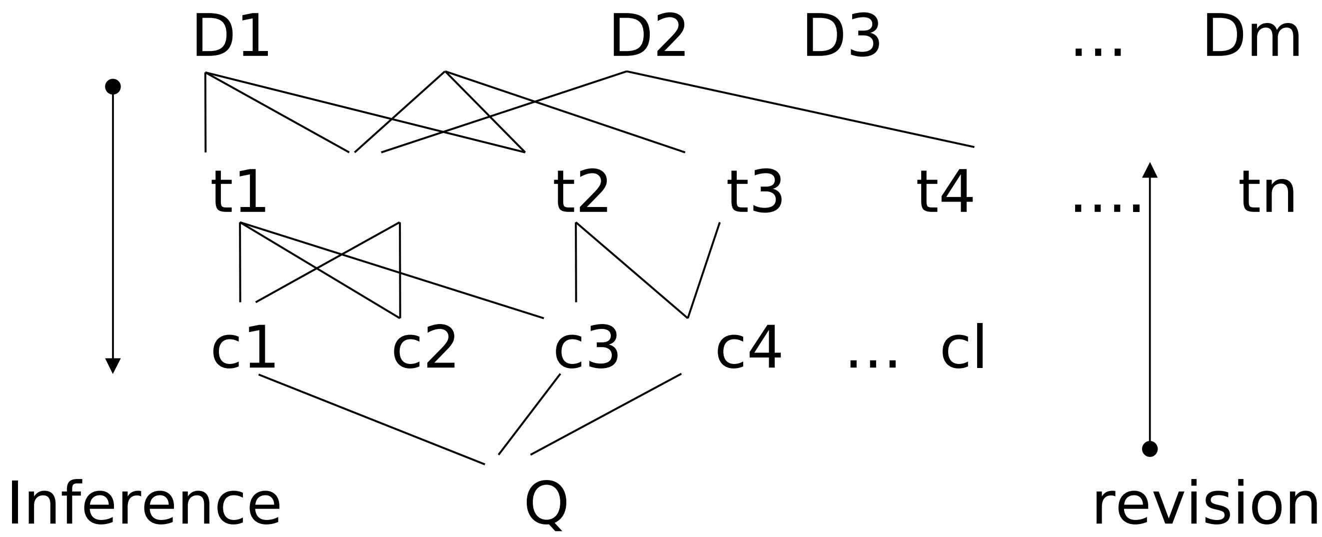
- Keywords are not perfect representatives of concepts
  - Ambiguity:

    table = data structure, furniture?
  - Lack of precision:

    “operating”, “system” less precise than “operating_system”
- Suggested solution
  - Sense disambiguation (difficult due to the lack of contextual information)
  - Using compound terms (no complete dictionary of compound terms, variation in form)
  - Using noun phrases (syntactic patterns + statistics)
- Still a long way to go

Theory ...

- Bayesian networks
  - P(Q|D)



- Language models

Logical models

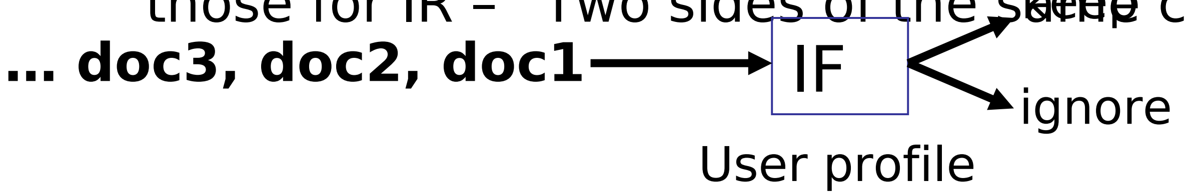
- How to describe the relevance relation as a logical relation?

  D => Q

- What are the properties of this relation?
- How to combine uncertainty with a logical framework?
- The problem: What is relevance?

Related applications: Information filtering

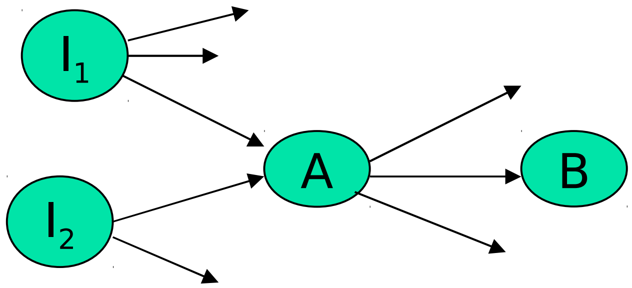
- IR: changing queries on stable document collection
- IF: incoming document flow with stable interests (queries)
  - yes/no decision (in stead of ordering documents)
  - Advantage: the description of user's interest may be improved using relevance feedback (the user is more willing to cooperate)
  - Difficulty: adjust threshold to keep/ignore document
  - The basic techniques used for IF are the same as those for IR – "Two sides of the same coin"



IR for (semi-)structured documents

- Using structural information to assign weights to keywords (Introduction, Conclusion, …)
  - Hierarchical indexing
- Querying within some structure (search in title, etc.)
  - INEX experiments
- Using hyperlinks in indexing and retrieval (e.g. Google)
- …

PageRank in Google



$$PR(A) = (1-d) + d\sum_i \frac{PR(I_i)}{C(I_i)}$$

- Assign a numeric value to each page
- The more a page is referred to by important pages, the more this page is important

- *d:* damping factor (0.85)

- Many other criteria: e.g. proximity of query words
  - "…information retrieval …" better than "… information … retrieval …"

IR on the Web

- No stable document collection (spider, crawler)
- Invalid document, duplication, etc.
- Huge number of documents (partial collection)
- Multimedia documents
- Great variation of document quality
- Multilingual problem
- …

Final remarks on IR

- IR is related to many areas:
  - NLP, AI, database, machine learning, user modeling…
  - library, Web, multimedia search, …
- Relatively week theories
- Very strong tradition of experiments
- Many remaining (and exciting) problems
- Difficult area: Intuitive methods do not necessarily improve effectiveness in practice

Why is IR difficult

- Vocabularies mismatching
  - Synonymy: e.g. car v.s. automobile
  - Polysemy: table
- Queries are ambiguous, they are partial specification of user's need
- Content representation may be inadequate and incomplete
- The user is the ultimate judge, but we don't know how the judge judges…
  - The notion of relevance is imprecise, context- and user-dependent
- But how much it is rewarding to gain 10% improvement!