1. Introduction to Information Retrieval
2. Information Retrieval Models

Mounia Lalmas
Yahoo! Research Barcelona

Information Retrieval Summer School 2011
Bangalore, India

About myself

- Positions
  - 1999-2008 Lecturer to Professor, Queen Mary University of London
  - 2008-2010 Microsoft Research/RAEng Research Professor, University of Glasgow
  - 2011- Visiting Principal Scientist, Yahoo! Research Barcelona

---

- Research topics
  - XML retrieval and evaluation (INEX)
  - Quantum theory to model interactive information retrieval
  - Aggregated search
  - Bridging the digital divide (Eastern Cape, South Africa)
  - **Models and measures of user engagement**

Outline

Outline

- Terminology
- Retrieval Tasks
- A Conceptual Model for IR
- Document and Document Representation
- Queries
- Best-match retrieval
- History
- Topics in IR
- Information Retrieval vs Information Extraction vs Web Search
- Important forums (Conferences and Journals)

Terminology

General: Information Retrieval, Information Need, Query, Retrieval Model, Retrieval Engine, Search Engine, Relevance, Relevance Feedback, Evaluation, Information Seeking, Human-Computer-Interaction, Browsing, Interfaces, Ad-hoc Retrieval, Filtering

---

Related: Document Management, Knowledge Engineering

Expert: term frequency, document frequency, inverse document frequency, vector-space model, probabilistic model, BM25, DFR, page rank, stemming, precision, recall

Document/Information/Knowledge

Retrieval/Management/Engineering

	document	information	knowledge
retrieval	indexing	ranking	reasoning
management	scanning	filtering	learning
engineering	structuring	modelling	annotating

Information Need

- Example of an information need in the context of the world wide web:

  Find all documents *(information!)* about universities in India that (1) offer master degrees in Information Retrieval and (2) are registered with ACM SIGIR. The information *(the document!)* should include full curriculum, fees, student campus, e-mail and other contact details.

- Formal representation of an information need = Query

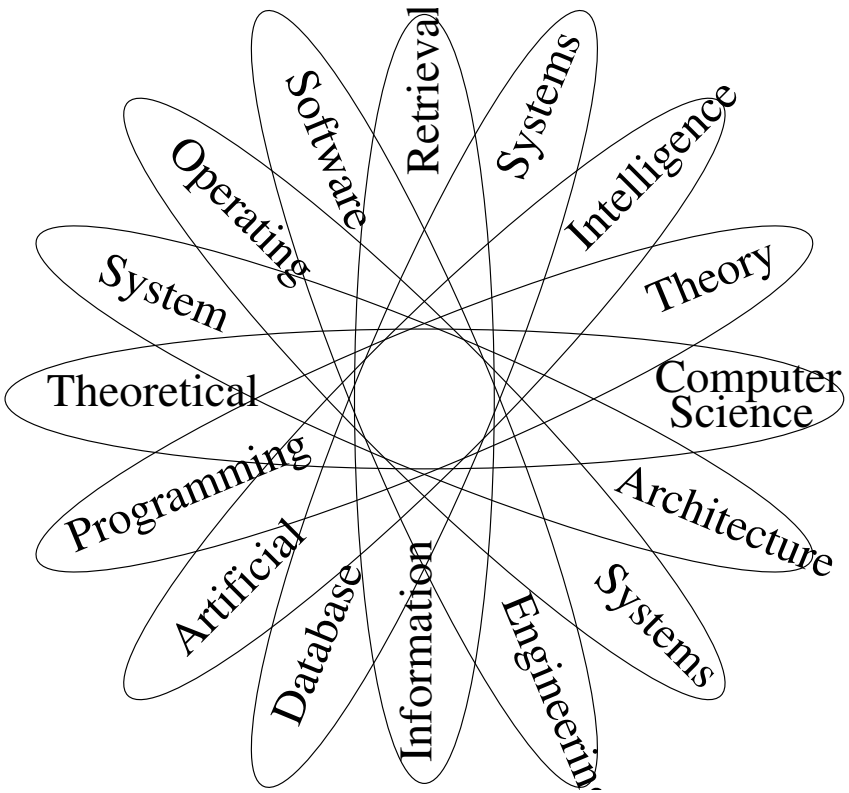
Information Retrieval: Informal Definition

Representation, storage, organisation and access of information (information items, information objects, documents).

Find relevant (useful) information

- **Goal of an IR system** - RECALL
  Retrieve all relevant documents (e.g. legal)
- **Goal of an IR system** - PRECISION
  Retrieve the most relevant documents (e.g. web).
- **Goal of an IR system:**
  - Retrieve as few non-relevant documents as possible.
  - Retrieve relevant documents before non-relevant documents.

Information Retrieval in Computer Science



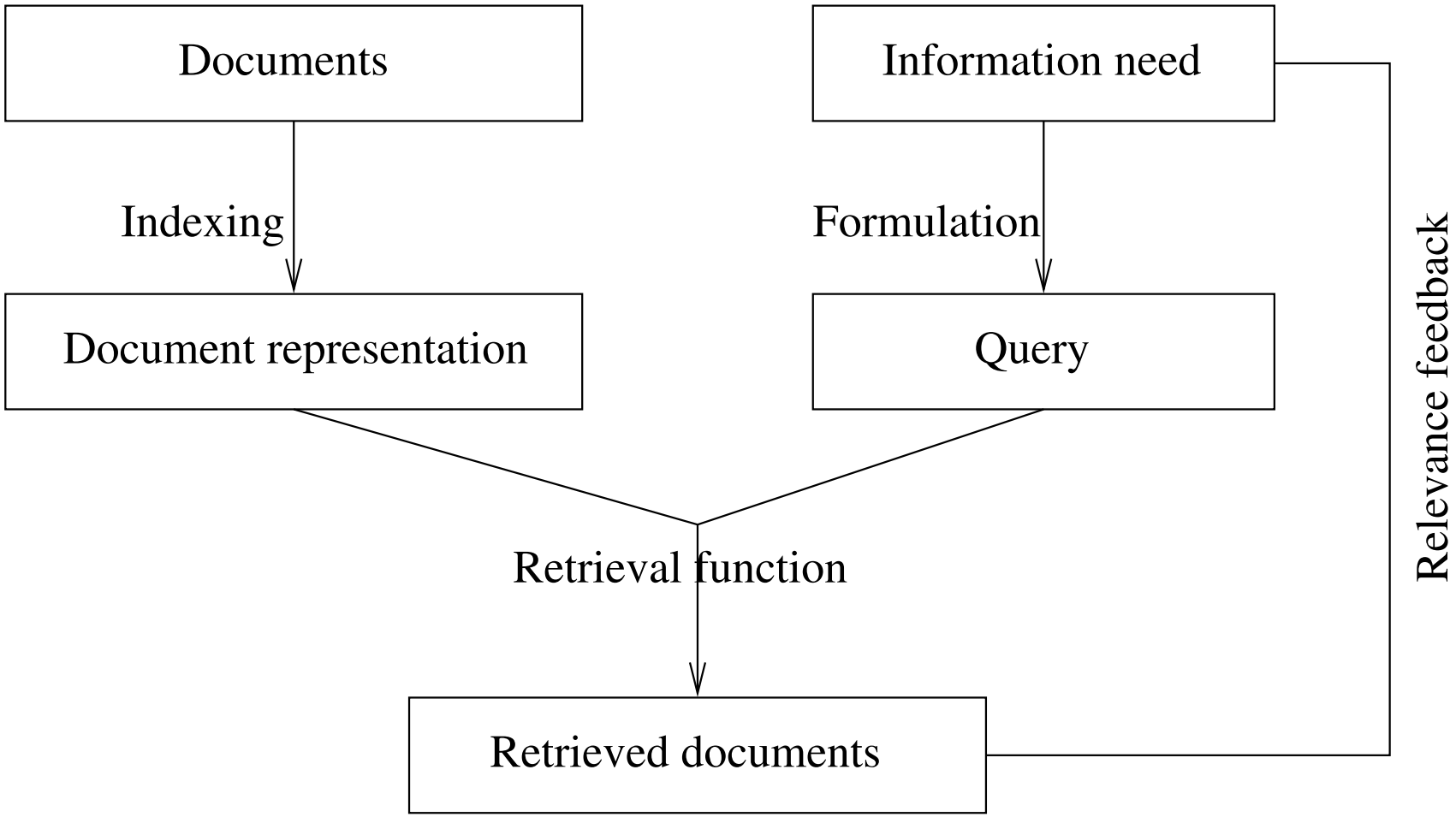
Information Retrieval / Data Retrieval

	Information Retrieval	Data Retrieval
Matching	vague	exact
Model	probabilistic	deterministic
Query language	natural	artificial
Query specification	incomplete	complete
Items wanted	relevant	all (matching)
Error handling	insensitive	sensitive

Retrieval Tasks

- Ad-hoc retrieval (querying) – pull technology
- Interactive query formulation
- Filtering – push technology
- Categorisation
- Clustering
- Search by browsing (hypertext)
- Search by browsing and querying (digital libraries and the web)

A Conceptual Model for IR



Documents

- Unit of retrieval
- A passage of free text
  - composed of text, strings of characters from an alphabet
  - composed of natural language:
    newspaper articles, journal paper, dictionary definition, e-mail messages
  - size of documents:
    arbitrary, newspaper article vs journal article vs e-mail

---

- Sub-document can also be a unit of retrieval (passage, XML element, answer to a question)

Document Representation

- Free-text representation: extracted directly from text, good performance in broad domains.
- Controlled vocabulary representation: most concise representation, good performance in narrow domains with limited number of (expert) users.

- Full-text representation: most complete representation, optimal performance, huge resource requirements.
- Reduced (partial) content representation: stopwords, stemming, noun phrases, compression.

- Structure representation: chapter, section, paragraph.

Queries

- Information Need
- Simple queries
  - composed of two or three, perhaps of dozen of keywords
  - e.g. as in web retrieval
- Boolean queries
  - 'neural network AND speech recognition'
  - e.g. as in online catalog and patent search
- Context queries
  - proximity search, phrase queries
  - e.g. neural ftand network distance at most 5 words (XQuery Full Text)

Best-Match Retrieval

- Compare the terms in a document and query
- Compute "similarity" between each document in the collection and the query based on the terms they have in common
- Sorting the document in order of decreasing similarity with the query
- The outputs are a ranked list and displayed to the user – the top ones are more relevant as judged by the system

---

Document term descriptors to access text

$\longleftrightarrow$

User term descriptors characterising user needs

History

- Manual IR in libraries: manual indexing; manual categorisation
- 70ies and 80ies: Automatic IR in libraries
- 90ies: IR on the web and in digital libraries

Success factors: Response time, coverage, interactivity, low (no!) costs, precision-oriented (you do not "feel" the recall)

---

precision $\approx$ correctness, recall $\approx$ completeness

(Some) Topics in IR

1. Retrieval models (ranking function, learning to rank, machine learning)
2. Text processing ("Indexing"): NLP / understanding (language models)
3. Interactivity and users
4. Efficiency, compression, MapReduce, Scalability
5. Distributed IR (data fusion, aggregated search, federated search)
6. Multimedia: image, video, sound, speech
7. Evaluation including crowd-sourcing
8. Web retrieval and social media search
9. Cross-lingual IR (FIRE), Structured Data (XML),
10. Digital libraries, Enterprise Search, Legal IR, Patent Search, Genomics IR

(see very nice (old) article in http://www.dlib.org/dlib/november95/11croft.html)

Information Retrieval vs Information Extraction

- Information Retrieval
  - Given a set of terms and a set of document terms select only the most relevant document (precision), and preferably all the relevant ones (recall)
- Information Extraction
  - Extract from the text what the document means
- IR can FIND documents but needs not "understand" them

Information Retrieval vs Web Search

- Most people equate information retrieval with web search
- Information retrieval is concerned with **the finding of** (any kind of) relevant information

Information Retrieval Forums

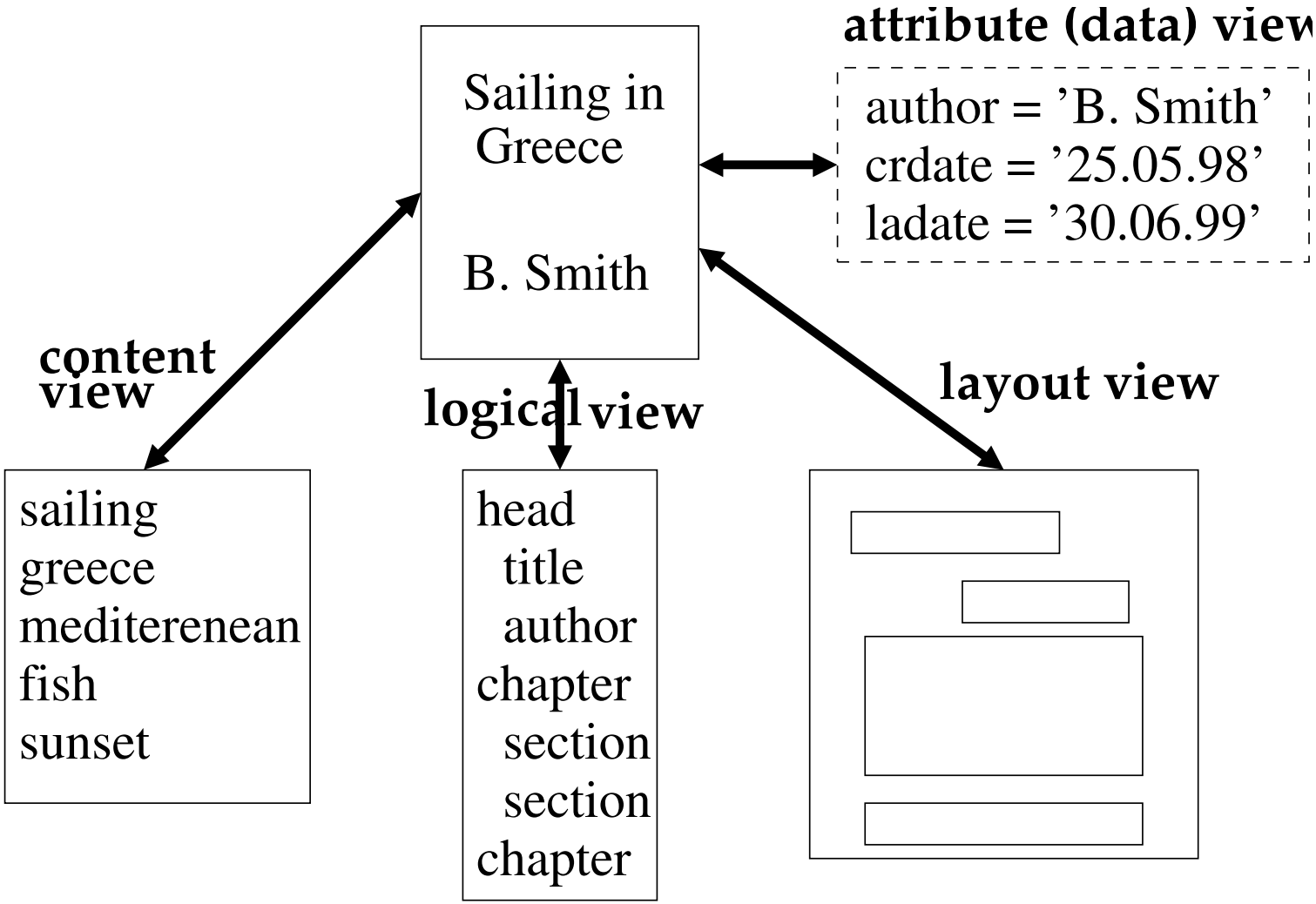
Conferences: SIGIR, CIKM, SPIRE, FQAS, BCS-IRSG (ECIR), RIAO, SAC-IAR, IIIX, EDCL, JCDL, IRF, ICTIR
http://www.sigir.org/events/events-upcoming.html

Journals: TOIS, IP&M, IR, JDOC, JASIST
http://www.sigir.org/resources.html

Outline

- Terminology
- Generating document representations
- Edit Distance and Soundex (e.g. used for spelling correction)
- Index term weighting
- Inverted file

Document views



"Media" types

- text
- image
- graphic
- audio (sound, speech, music)
- video
- animation

Document types

- **monomedia document**: text document, etc.
- **multimedia document**: document containing parts of different media
- **hypertext document:** document with links; referred to as non-linear document
- **hypermedia document:** multimedia + hypertext

- **user generated (content) document:** blogs, comments, tweets

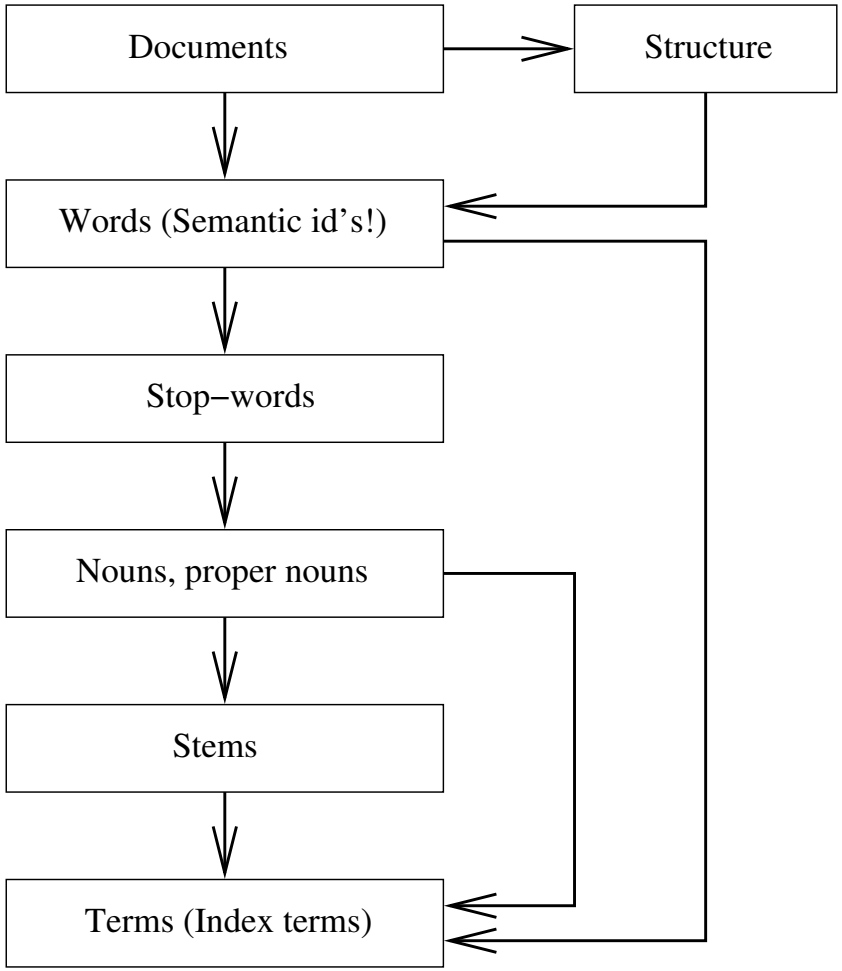
Outline

1. Indexing language
2. Overview
3. Identify words
4. Remove stop-words
5. Detect other specials (e. g. formulae analysis, date detection)
6. Stem words
7. Using a Thesaurus (before or after stemming!?)

Indexing Language

- Language used to describe documents and queries
  - Index terms selected subset of words
  - Derived from the text or arrived at independently
- Keyword searching
  - Statistical analysis of document based of word occurrence frequency
  - Automated, efficient and potentially inaccurate
- Searching using controlled vocabularies
  - More accurate results but time consuming if documents manually indexed

Overview


Documents
Structure
Words (Semantic id's!)
Stop−words
Nouns, proper nouns
Stems
Terms (Index terms)

Identify words

- define word separators: white spaces are word separators
- ignore punctuation: '.', ',', etc. is transposed to white space (word separator); exception: numbers such as 1.7 or 10,000
- deal with hyphen (dash) and underscore: '-', '_'; exception: formulae
- deal with apostrophe and quotes
- deal with numbers
- deal with strings and compounds (phrases)
- transpose to lower case (lower case preferred to upper case)

Remove stop-words

is, a, the, or, and, ...

not?

other?

Stop-word list often defined manually. How would you derive the stop-word list automatically?

Reduction: between 30 and 50 per cent

Detection of special expressions

Approach 1: Use heuristic rules (regular expressions) for recognising nouns, proper nouns, credit card numbers, email addresses, phone numbers, dates, web page locators, etc.

Approach 2: Learn rules

---

application-dependent!

Stemming approaches

1. dictionary-based: play+ing $\rightarrow$ play
2. rule-based:
   Prefix or suffix ("affix") removal, language-dependent.
   { sailing, sailor } $\rightarrow$ sail
   Widely used stemmer: Porter stemmer (Snowball)
3. n-gram-based: information $\rightarrow$ { inf, nfo, for }

- For other languages, e.g. Japanese, Chinese, etc very different

Stemming - Plural (in English)

1. If word ends in "ies" but not "eies", aies"
   "ies → "y
2. if word ends in "es" but not "aes, "ees, "oes"
   "es → "e"
3. if word ends in "s but not "us" or "ss"
   "s" → ""

First applicable rule is the one used

Stemming issues and terminology

under-stemming and *over-stemming*

over-stemming: { organisation, organ } $\rightarrow$ org

Polysemous: Several related (homologous) meanings.

Homonym: Several distinct meanings (e. g. bank, left).

Homograph: Same spelling but meaning depends on pronunciation (e.g. bear).

Homophone: Similar pronunciation but different spellings (night and knight, worth and worse).

Morpheme: The smallest part of a word with a meaning.

Example

"The destruction of the Amazon rain forests"

Case normalisation
Stop word removal (From fixed list)

"destruction amazon rain forests"

Suffix removal (stemming).

"destruct amazon rain forest"

Using a Thesaurus

bank: 1. finance institute; 2. river edge.

sailor: person sailing on boats.

Disambiguation: If bank occurs, then decide whether it has the financial or the river meaning.

Widely known thesaurus: WordNet
http://wordnet.princeton.edu/perl/webwn

Edit Distance

What is the value of edit-distance("Virginia", "Vermont")=?

Scan Virginia, and replace non-matching characters.

V*i*rginia
Ver*g*inia
Verm*i*nia
Vermon*i*a
Vernont*a*
Vermont

edit-distance(virginia, vermont) = 5

Soundex

Soundex translation table:

1	b,f,p,v
2	c,g,j,k,q,s,x,z
3	d, t
4	l
5	m, n
6	r

What about vowels?

Soundex code: One letter plus 3 digits.

Keep first letter, discard vowels, discard repetitions, sequential consonants, etc.

Soundex: Examples

Miller	M460
Peterson	P362
Petersen	P362
Auerbach	A612
Uhrbach	U612
Moskowitz	M232
Moskovitz	M213

Index Term Weighting

- Effectiveness of an indexing language:
- Exhaustivity
  - number of different topics indexed
  - high exhaustivity: high recall and low precision
- Specificity
  - ability of the indexing language to describe topics precisely
  - high specificity: high precision and low recall

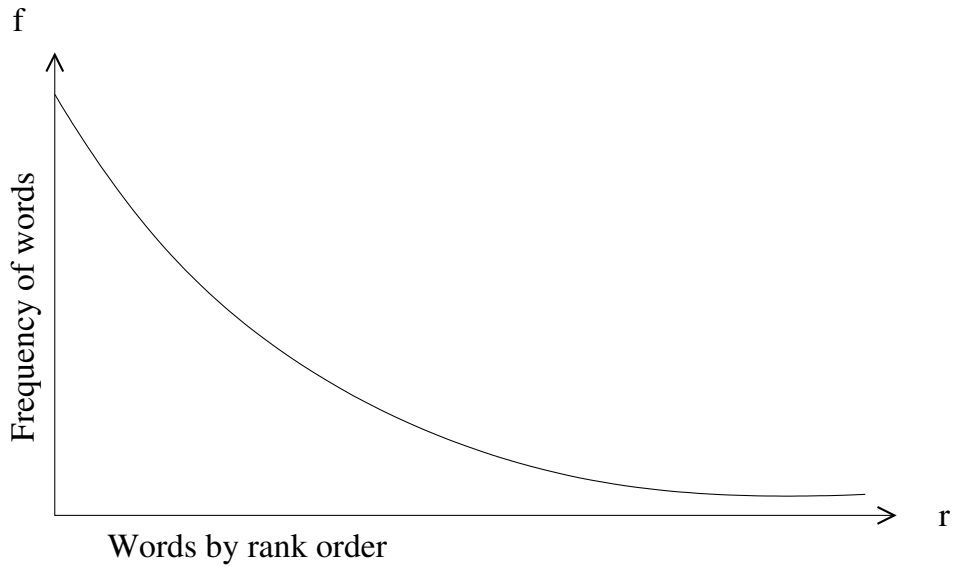
Index Term Weighting

- Exhaustivity
  - related to the number of index terms assigned to a given document
- Specificity
  - number of documents to which a term is assigned in a collection
  - related to the distribution of index terms in collection

- Index term weighting
  - index term frequency: occurrence frequency of a term in document
  - document frequency: number of documents in which a term occurs

Zipf's law [1949]

Distribution of word frequencies is similar for different texts (natural language) of significantly large size



Zipf's law holds even for different languages!

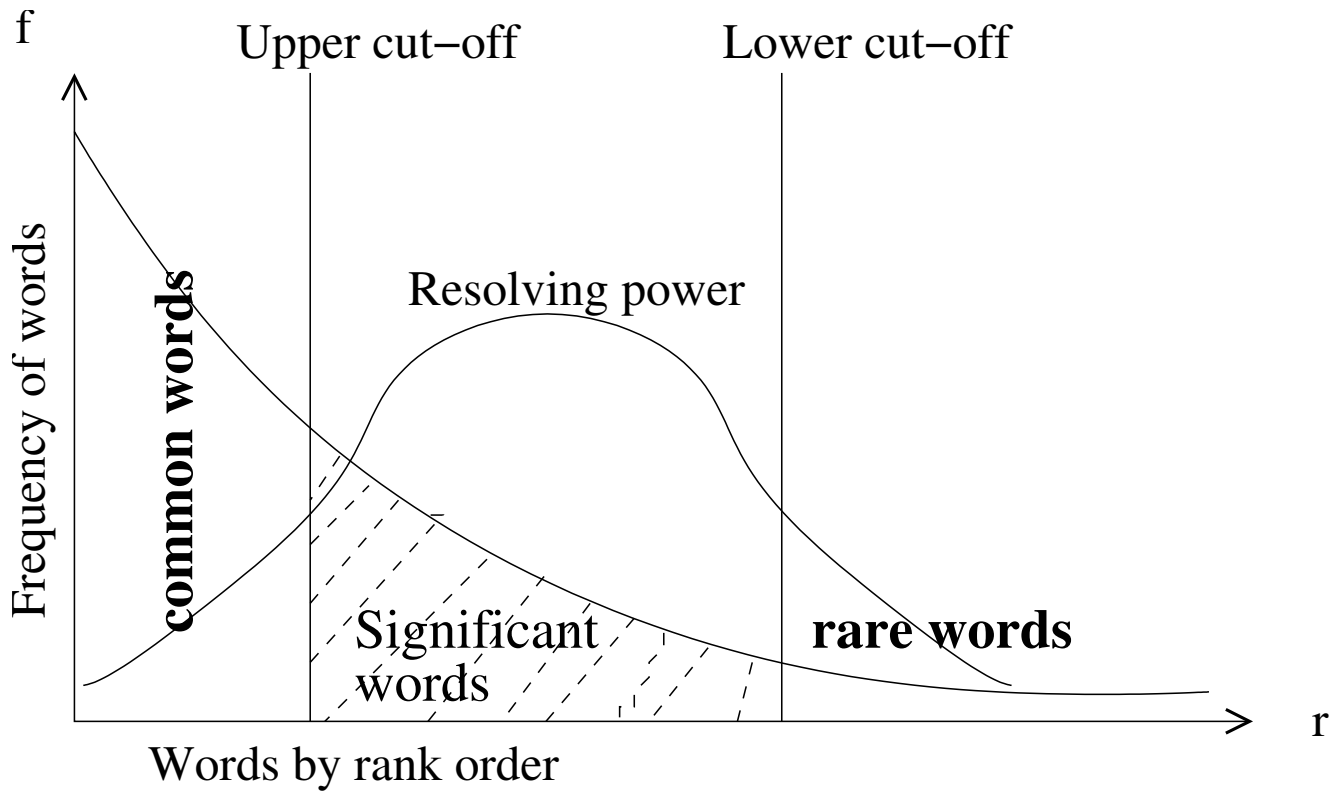
Luhn's analysis — Hypothesis

[1957]

Hypothesis: Frequency of words is a measurement of word significance.

... a measurement of the power of a word to discriminate documents by their content ...

Luhn's analysis — Observation



Luhn's analysis — Explanation

Resolving/Discriminating power of words

Optimal power half way between the cut-offs

tf-idf

$weight(t,d) = tf(t,d) \times idf(t)$	
N	number of documents in collection
$n(t)$	number of documents in which term t occurs
$idf(t)$	inverse document frequency of term t
$occ(t,d)$	occurrence of term t in document d
t_{max}	term in document d with highest occurrence
$tf(t,d)$	term frequency of t in document d

SMART retrieval system, Salton [1971]

$$tf(t,d) := \frac{occ(t,d)}{occ(t_{max},d)}$$

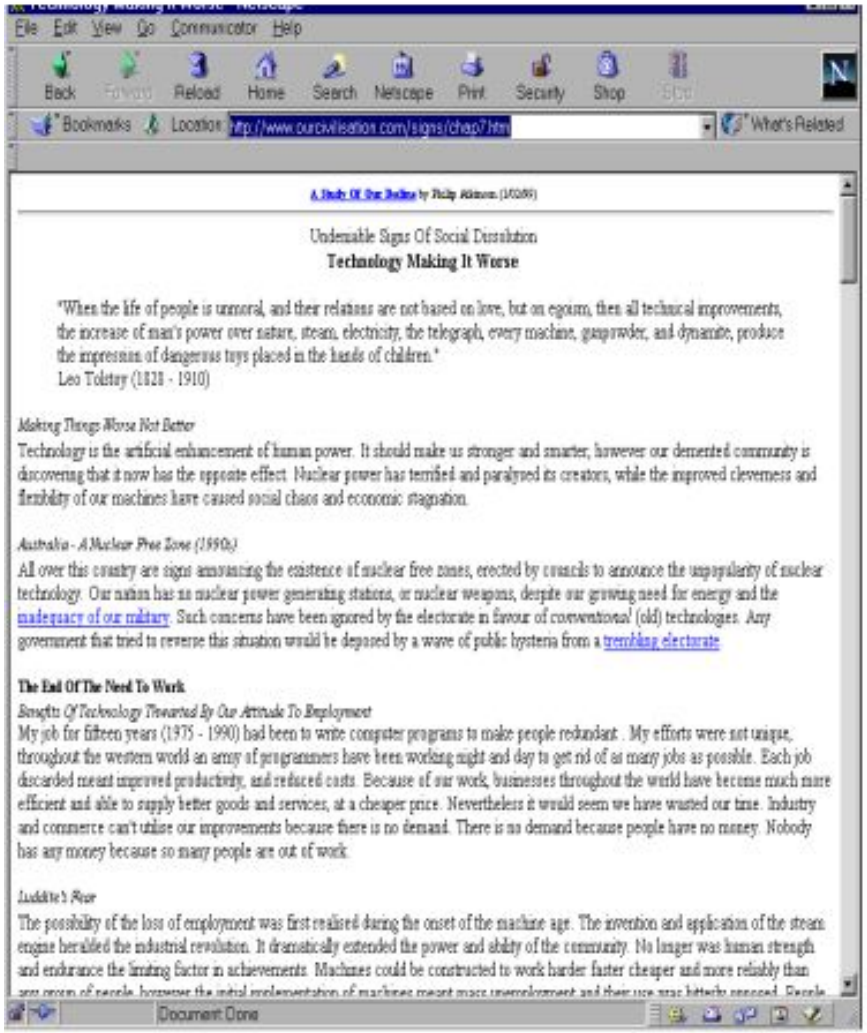
With lifting factor:

$$tf(t,d) := l + (1-l)\frac{occ(t,d)}{occ(t_{max},d)}$$

$$tf(t,d) := 0.5 + 0.5\frac{occ(t,d)}{occ(t_{max},d)}$$

$$idf(t) := \log \frac{N}{n(t)}$$

$$idf(t) = -\log \frac{n(t)}{N}$$



Example

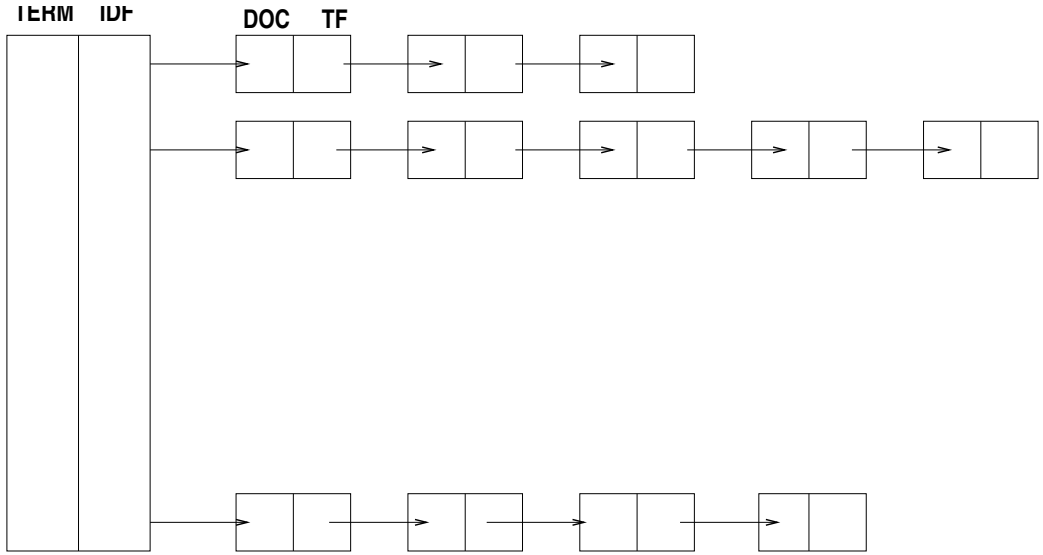
Nuclear 7	Computer 9
Poverty 5	Unemployment 1
Luddites 3	Machines 19
People 25	And 49

Weight(machine) =
19/25 x log(100/50)
= 0.76 x 0.3013 = 0.228988

Weight(luddite) =
3/25 x log(100/2)
= 0.12 x 1.69897 = 0.2038764

Weight(poverty) = 5/25 x log(100/2) = 0.2 x 1.69897 = 0.339794

Inverted file



- Word-oriented mechanism for indexing collections to speed up searching
- Searching:
  - vocabulary search (query terms)
  - retrieval of occurrence
  - manipulation of occurrence

Document vs Inverted Views

	Cosmonaut	astronaut	moon	car	truck
D1	1	0	1	1	1
D2	0	1	1	0	0
D3	0	0	0	1	1

	D1	D2	D3
Cosmonaut	1	0	0
astronaut	0	1	0
moon	1	1	0
Car	1	0	1
truck	1	0	1

What goes in the inverted file

- Boolean retrieval
  - Just the document number
- Ranked Retrieval
  - Document number and term weight (TF, IDF, TF*IDF, ...)
- Proximity operators
  - Word offsets for each occurrence of the term
  - Example: t17 (doc1,49) (doc1,70) (doc2,3)

How big is the inverted file

- Very compact for Boolean retrieval
  - About 10% of the size of the documents
  - If an aggressive stopword list is used
- Not much larger for ranked retrieval
  - Perhaps 20%
- Enormous for proximity operators
  - Sometimes larger than the documents
  - But access is fast - you know where to look

Outline

- What to evaluate
- Test collections
- Precision and recall

What can we evaluate in IR

- **coverage** of the collection: extent to which the system includes relevant material
  - this is (was) important in web retrieval (since it was the case that individual search - Altavista, Lycos, etc) engine covers maybe up to 16% of the web space.
- **efficiency** in terms of speed, memory usage, etc.
- **time lag (efficiency):** average interval between the time a request is made and the answer is given
- **presentation** of the output, has to do with interface and visualisation issues.
- **effort** involved by user in obtaining answers to a request
- **recall** of the system: proportion of relevant documents retrieved
- **precision** of the system: proportion of the retrieved documents that are actually relevant

System-oriented evaluation

- Test collection methodology
  - Benchmark (data set) upon which effectiveness is measured and compared
  - Data that tell us for a given query what are the relevant documents.
- Measuring effectiveness has been the most predominant in IR evaluation:
  - **recall** of the system: proportion of **relevant** documents retrieved
  - **precision** of the system: proportion of the retrieved documents that are actually **relevant**
- Looking at these two aspects is part of what is called **system-oriented evaluation**.

Test Collections

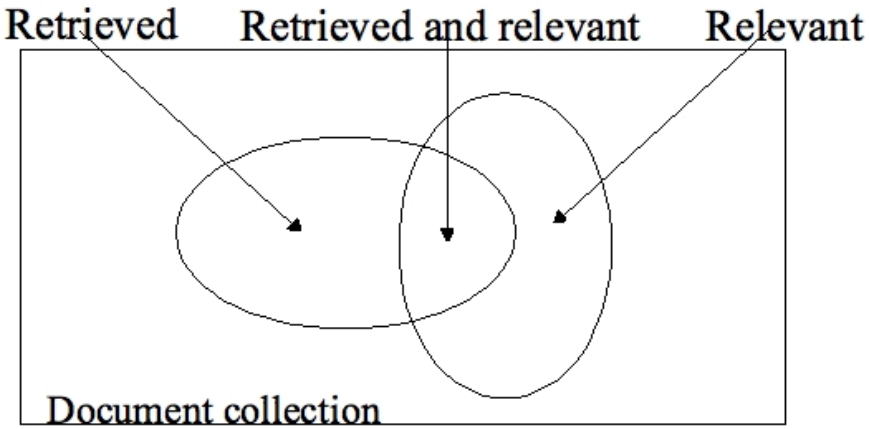
- Compare retrieval performance using a test collection
  - Document collection, that is the document themselves. The document collection depends on the task, e.g. evaluating web retrieval requires a collection of HTML documents.
  - Queries / requests, which simulate real user information needs.
  - Relevance judgements, stating for a query the relevant documents.
- To compare the performance of two techniques:
  - each technique used to evaluate test queries
  - results (set or ranked list) compared using some performance measure
  - most common measures -precision and recall
- Usually use multiple measures to get different views of performance
- Usually test with multiple collections as performance is collection dependent

Effectiveness

- We recall that the goal of an IR system is to retrieve as many relevant documents as possible and as few non-relevant documents as possible.
- Evaluating the above consists of a comparative evaluation of technical performance of IR system(s):
  - In traditional IR, technical performance means the effectiveness of the IR system: the ability of the IR system to retrieve relevant documents and suppress non-relevant documents
  - Effectiveness is measured by the combination of **recall** and **precision**.

Recall / Precision

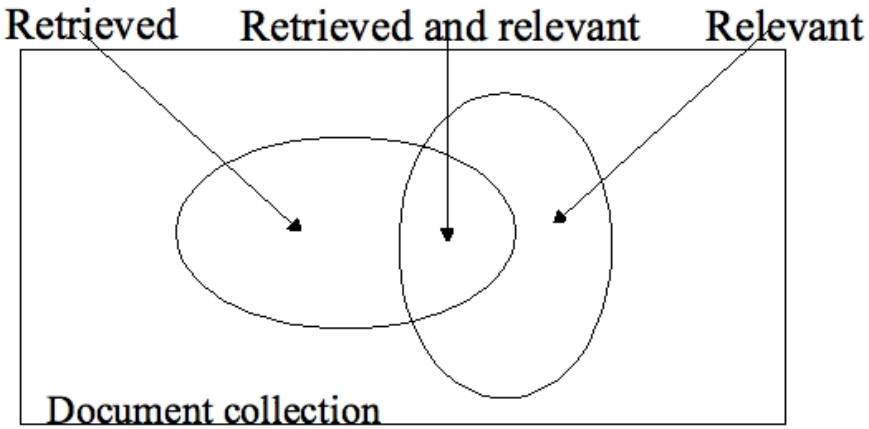
For a given query, the document collection can be divided into three sets: the set of retrieved document, the set of relevant documents, and the rest of the documents.



Note: knowing which documents are relevant comes from the test collection

Recall / Precision

In the ideal case, the set of retrieved documents is equal to the set of relevant documents. However, in most cases, the two sets will be different. This difference is formally measured with precision and recall.



$$Precision = \frac{\text{number of relevant documents retrieved}}{\text{number of documents retrieved}}$$

$$Recall = \frac{\text{number of relevant documents retrieved}}{\text{number of relevant documents}}$$

Recall / Precision

$$Precision = \frac{\text{number of relevant documents retrieved}}{\text{number of documents retrieved}}$$

$$Recall = \frac{\text{number of relevant documents retrieved}}{\text{number of relevant documents}}$$

The above two measures do not take into account where the relevant documents are retrieved, this is, at which rank (crucial since the output of most IR systems is a ranked list of documents).

This is very important because an effective IR system should not only retrieve as many relevant documents as possible and as few non-relevant documents as possible, but also it should retrieve relevant documents **before** the non-relevant ones.

Recall / Precision

- Let us assume that for a given query, the following documents are relevant (10 relevant documents)

  {d3, d5, d9, d25, d39, d44, d56, d71, d89, d123}

- Now suppose that the following documents are retrieved for that query:

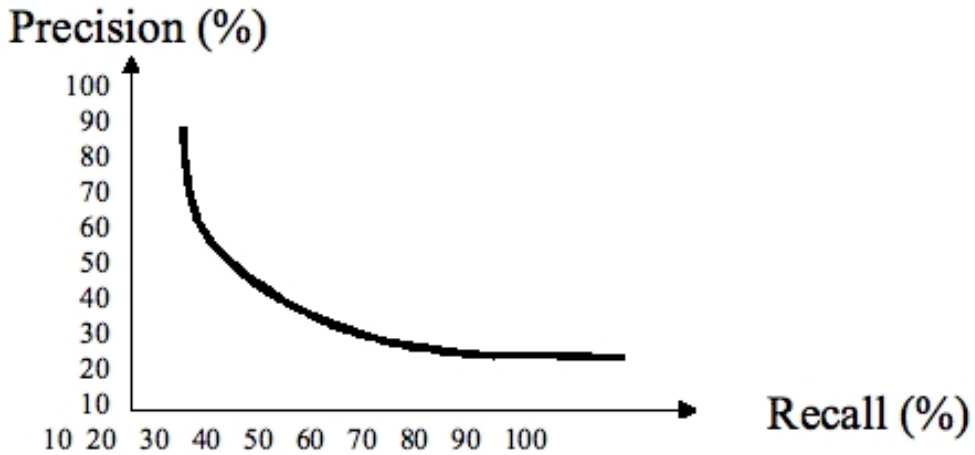
rank	doc	precision	recall	rank	doc	precision	recall
1	**d123**	1/1	1/10	8	d129		
2	d84			9	d187		
3	**d56**	2/3	2/10	10	**d25**	4/10	4/10
4	d6			11	d48		
5	d8			12	d250		
6	**d9**	3/6	3/10	13	d113		
7	d511			14	**d3**	5/14	5/10

- For each relevant document (in red bold), we calculate the precision value and the recall value. For example, for d56, we have 3 retrieved documents, and 2 among them are relevant, so the precision is 2/3. We have 2 of the relevant documents so far retrieved (the total number of relevant documents being 10), so recall is 2/10.

Recall / Precision

- For each query, we obtain pairs of recall and precision values
  - In our example, we would obtain (1/10, 1/1) (2/10, 2/3) (3/10, 3/6) (4/10, 4/10) (5/10, 5/14) . . . which are usually expressed in % (10%, 100%) (20%, 66.66%) (30%, 50%) (40%, 40%) (50%, 35.71%) . . .
  - This can be read for instance: at 20% recall, we have 66.66% precision; at 50% recall, we have 35.71% precision

The pairs of values are plotted into a graph, which has the following curve



The complete methodology

For each IR system / IR system version

- For each query in the test collection
  - We first run the query against the system to obtain a ranked list of retrieved documents
  - We use the ranking and relevance judgements to calculate recall/precision pairs
- Then we average recall / precision values across all queries, to obtain an overall measure of the effectiveness.

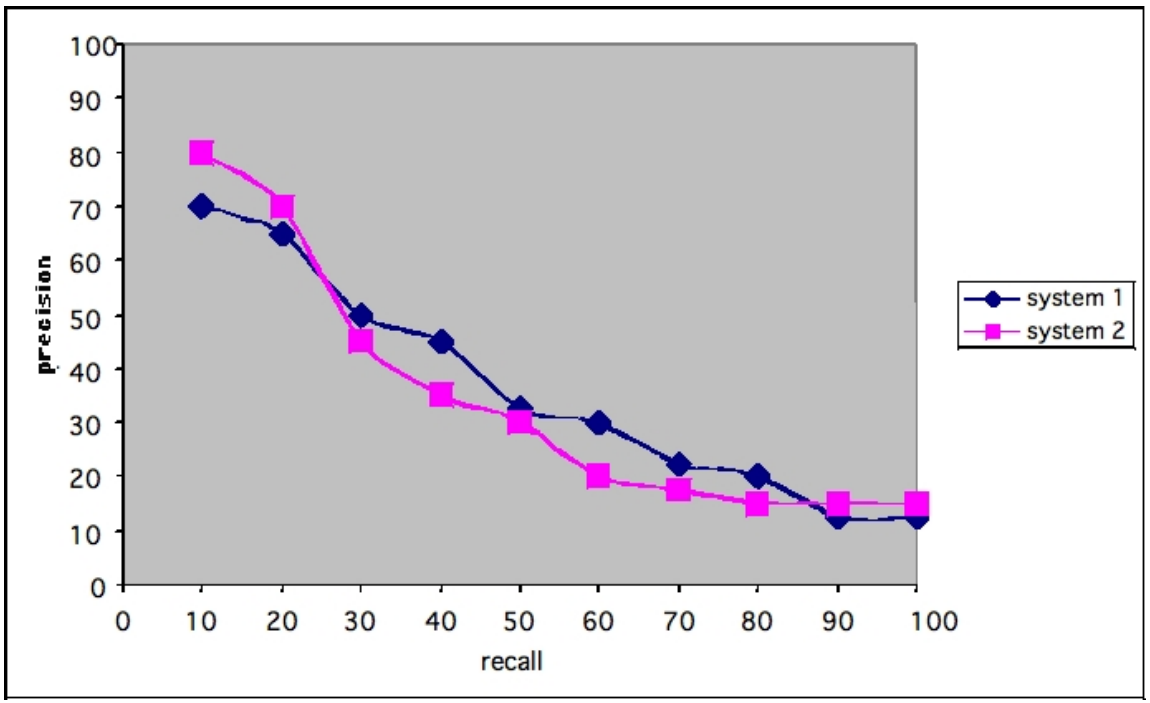
Averaging

Recall in %	Precision in %		
	Query 1	Query 2	Average
10	80	60	**70**
20	80	50	**65**
30	60	40	**50**
40	60	30	**45**
50	40	25	**32.5**
60	40	20	**30**
70	30	15	**22.5**
80	30	10	**20**
90	20	5	**11.5**
100	20	5	**11.5**

The same information can be displayed in a plot.

Comparison of systems

We can compare IR systems / system versions. For example, here we see that at low recall, system 2 is better than system 1, but this changes from recall value 30%, etc. It is common to calculate an average precision value across all recall levels, so that to have a single value to compare, so called Mean average precision (MAP).



Outline

- Notations - Revision
- Components of a retrieval model
- Retrieval Models I: Boolean, VSM, BIRM and BM25
- Retrieval Models II: Probabilities, Language Models, and DFR
- Retrieval Models III: Relevance feedback
- Retrieval Models IV: PageRank, inference networks, others

(Mathematical) Notations – Revision

- $d \rightarrow q$: d implies q as in classical logic
- $d \cap q$: the intersection of the set d and the set q
- $|d|$: the cardinal of the set d, ie the number of elements in the set d
- $d \cup q$: the union of the set d and the set q
- $\sum_{i=1,n} a_i = a_1 + a_2 + \ldots + a_n$
- $\prod_{i=1,n} a_i = a_1 \cdot a_2 \cdot \ldots \cdot a_n$

Components of a retrieval model

- D is the set of document representations (called call from now on documents for simplicity)
- Q is the set of information need representations (called from now on queries)
- $R(d, q)$ is a ranking function that
  - associates a real number, usually between 0 and 1, for a document $d \in D$ and a query $q \in Q$
  - can be used to define an ordering for the documents in D with respect to the query q,
  - where the ordering is suppose to reflect relevance (hopefully).

Components of a retrieval model

- For each retrieval model, we will make explicit the three components:
  - Document representation d
  - Query q
  - Ranking function $R(d, q)$

Boolean model

- Retrieve documents that make the query true.

$$R(d,q) = \begin{cases} 1 \text{ if } d \rightarrow q \\ 0 \text{ otherwise} \end{cases}$$

- Query (and document): logical combination of index terms
  q = (sailing $\wedge$ boats) $\vee$ (bowskill $\wedge\neg$ south_coast)
- "Query evaluation" based on inverted file:

  sailing = { d1, d2, d3, d4 }
  boats = { d1, d2 }
  bowskill= { d1, d2, d3 }
  south_coast= { d1 }
- Negation "felt" to be useful — proved to be not effective! Leads to empty results!
- No ranking: either a document is retrieved or not: $\{d_1, d_2, d_3\}$

Set-based models — co-ordination level match

- Query q: set of index terms and Document d: set of index terms
- Ranking based on the cardinality of $d \cap q$, ie number of terms common to the document and the query

$$R(d, q) = |d \cap q|$$

- For $q = \{$ sailing, boats, bowskill $\}$, the outcome is a (partially ranked) list of documents

$\lvert d \cap q \rvert$	documents	rank
3	d1, d2	1
2	d3	3
1	d4	4

Note: counter-intuitive results may occur due to document and query length.

Set-based models — other coefficients

$R(d, q)$ is based on $|d \cap q|$ but with some normalisation so that to not obtain counter-intuitive results

Dice: $$R(d, q) = \frac{2 \cdot |d \cap q|}{|d| + |q|}$$

Jaccard: $$R(d, q) = \frac{|d \cap q|}{|d \cup q|}$$

Cosine: $$R(d, q) = \frac{|d \cap q|}{\sqrt{|d| \cdot |q|}}$$

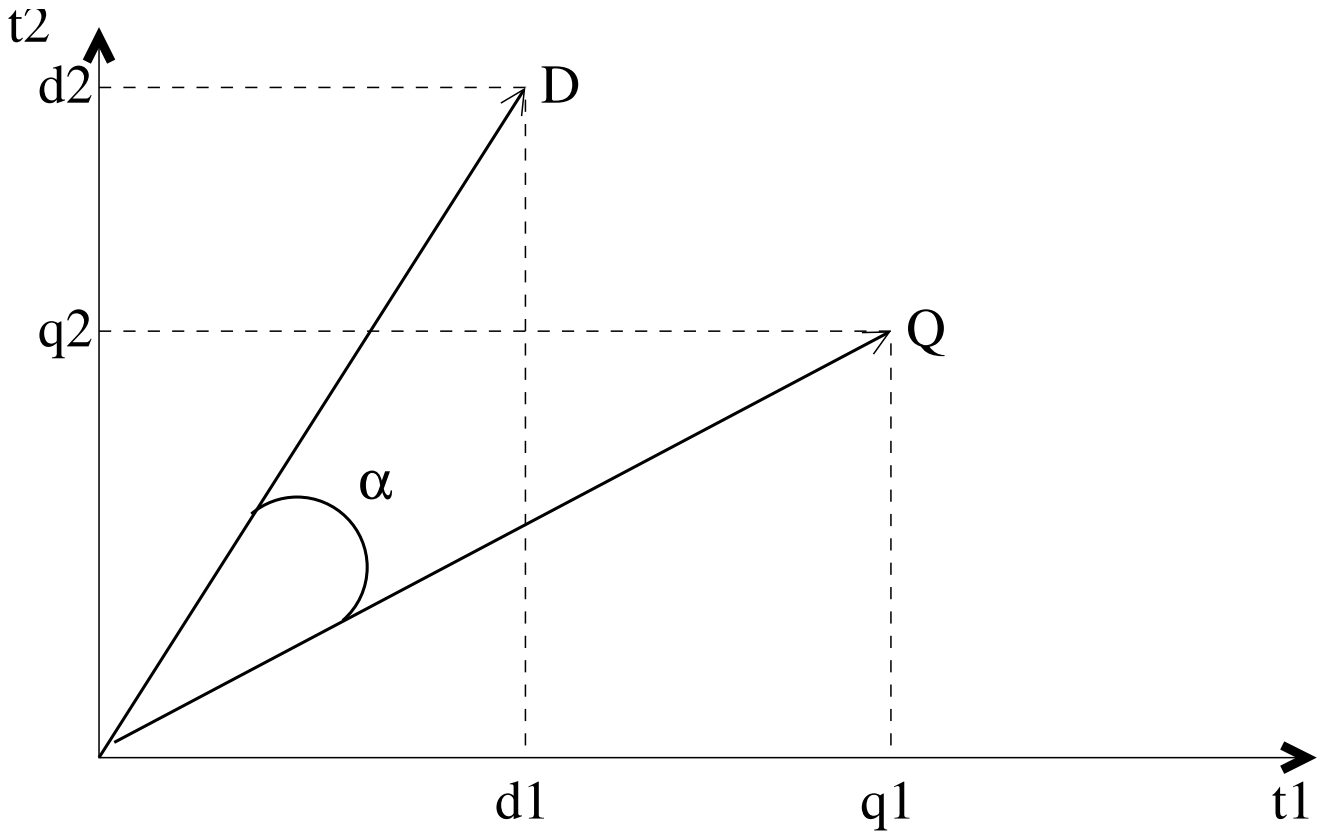
Vector space model — Introduction

- Set of n terms $\{t_1, t_2, \ldots, t_n\}$ (order is important)
- Document represented as a vector: $d =< d_1, d_2, \ldots, d_n >$
  Query represented as a vector: $q =< q_1, q_2, \ldots, q_n >$
  - d_i = weight of term t_i in document d (eg, based on $tf \times idf$)
  - q_i = weight of term t_i in query q (eg, 1 if $t_i \in q$; 0 otherwise)
- Ranking function, called retrieval status value (often written RSV):

$$R(d,q) = RSV(d,q) = \frac{\sum_{i=1,n} d_i q_i}{(\sum_{i=1,n} d_i^2)^{1/2} (\sum_{i=1,n} q_i^2)^{1/2}} = \cos\alpha$$

- Dimension of the vector is n
- Special case: binary vectors (cosine function as given in previous slide)

Vector space model — Graphical interpretation



Here $n = 2$, meaning two terms in the collection.

Vector space model — Vector Notation

- Document represented as a vector: $\vec{d} =< d_1, d_2, \ldots, d_n >$
- Query represented as a vector: $\vec{q} =< q_1, q_2, \ldots, q_n >$
- Ranking function (retrieval status value):

$$R(d,q) = \frac{\sum_{i=1,n} d_i q_i}{(\sum_{i=1,n} d_i^2)^{1/2} (\sum_{i=1,n} q_i^2)^{1/2}} = \cos\alpha$$

$$R(d,q) = sim(\vec{d},\vec{q}) = \cos\alpha = \frac{\vec{d}\cdot\vec{q}}{\sqrt{\vec{d}^2}\cdot\sqrt{\vec{q}^2}}$$

Generalised vector space model

$$R(d,q) := \vec{d}^T \cdot G \cdot \vec{q}$$

$$G = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} d_1 & d_2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} d_1 & (d_1 + d_2) \end{pmatrix} \cdot \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}$$

$$= d_1 \cdot q_1 + (d_1 + d_2) \cdot q_2 = d_1 \cdot q_1 + d_2 \cdot q_2 + d_1 \cdot q_2$$

Generalised vector-space model

$$G = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad R(d,q) = d_1 \cdot q_1 + d_2 \cdot q_2 + d_1 \cdot q_2$$

- Relationships between terms are considered, as modelled in G:
  - Upper triangle: $G_{1,2} = 1$ produces $(d_1 \cdot q_2)$. Lower triangle: ???
  - Here if term t_1 in document, and term t_2 in query, then consider a match, but not the other way round.
  - Example: t_1=dog and t_2=animal

Probabilistic Model

"Given a user query q and a document d, estimate the probability that the user will find d relevant."

- We only discuss the Binary Independence Retrieval Model (BIRM)
  - based on information related to presence and absence of terms in relevant and non-relevant documents
  - information acquired through relevance feedback process:
    - user stating which of the retrieved documents are relevant / non-relevant (covered later)

Binary independence retrieval model (BIRM)

- A document is described by presence/absence of terms: $d =< x_1, x_2, \ldots, x_n >$ with $n =$ number of terms.

$$x_i = \begin{cases} 1 \text{ if document } d \text{ indexed by } t_i \\ 0 \text{ otherwise} \end{cases}$$

1- compute for given query q:
  - $P(r|d,q)$, the probability of d being relevant (r)
  - $P(\neg r|d,q)$, the probability of d not being relevant ($\neg r$)

2- then decide whether document represented by d is relevant to query q.

- The decision is expressed by the Bayes' decision rule.

BIRM: The Bayes' decision rule

- For each query q defined as a set of terms, we have a set of relevant documents (binary vectors)
  - $P(r|d,q)$: probability of judgement being relevant (r) given document d and query q
  - $P(\neg r|d,q)$: probability of judgement being *not* relevant ($\neg r$) given document d and query q

---

Bayesian decision rule:

if $P(r|d,q) > P(\neg r|d,q)$
  retrieve d
else
  do not retrieve d

BIRM: Bayes' decision rule and retrieval function

Bayes' decision rule:
"if $P(r|d,q) > P(\neg r|d,q)$ then retrieve d; otherwise don't"

From above decision rule, a retrieval function $R(d,q) = g(d,q)$ is derived:

$$g(d,q) = \begin{cases} g(d,q) > C & \text{retrieve document represented by } d \\ g(d,q) \leq C & \text{do not retrieve document represented by } d \end{cases}$$

for some constant C

We show how $g(d,q)$ is obtained.

BIRM: The Bayes' decision rule

if $P(r|d,q) > P(\neg r|d,q)$
  retrieve d
else
  do not retrieve d

- The rule says: if $P(r|d,q) > P(\neg r|d,q)$ then d is relevant for query q; otherwise d is not relevant.
- To implement this rule, need to compute $P(r|d,q)$ and $P(\neg r|d,q)$
- Since these probabilities are with respect to same query q, simplify the above to $P(r|d)$ and $P(\neg r|d)$
  $\rightarrow$ We show how to obtain $g(d,q) = g(d)$

BIRM: Bayes' theorem

The rule is implemented through the use of Bayes' theorem

$$P(r|d) = \frac{P(d|r) \cdot P(r)}{P(d)} \qquad P(\neg r|d) = \frac{P(d|\neg r) \cdot P(\neg r)}{P(d)}$$

- $P(d)$: probability of observing d at random, ie probability of d irrespective of whether it is relevant or not.
- $P(d|r)$: probability of observing d given relevance
- $P(d|\neg r)$: probability of observing d given non relevance
- $P(r)$: prior probability of observing a relevant document
- $P(\neg r)$: prior probability of observing a non relevant document
- Note that from probability theory: $P(d) = P(d|r) \cdot P(r) + P(d|\neg r) \cdot P(\neg r)$

BIRM: Bayes' theorem and Bayes' decision rule

$$P(r|d) > P(\neg r|d)$$

can be rewritten as:

$$\frac{P(d|r)\cdot P(r)}{P(d)} > \frac{P(d|\neg r)\cdot P(\neg r)}{P(d)}$$

which is the same as:

$$P(d|r) \cdot P(r) > P(d|\neg r) \cdot P(\neg r)$$

The above can be rewritten as

$$\frac{P(d|r) \cdot P(r)}{P(d|\neg r) \cdot P(\neg r)} > 1$$

BIRM: Independence Assumption

We recall that $d =< x_1, x_2, \ldots, x_n >$ where $x_i = 1$ or 0.

BIRM assume independence with respect to relevance:

$$P(d|r) = P(< x_1, \ldots, x_n > |r) = \prod_{i=1,n} P(x_i|r)$$

BIRM assume independence with respect to non relevance:

$$P(d|\neg r) = P(< x_1, \ldots, x_n > |\neg r) = \prod_{i=1,n} P(x_i|\neg r)$$

BIRM: Notations

$a_i := P(x_i = 1\|r)$:	probability that term t_i occurs in a relevant document
$1 - a_i = P(x_i = 0\|r)$:	probability that term t_i does not occur in a relevant document
$b_i := P(x_i = 1\|\neg r)$:	probability that term t_i occurs in a non-relevant document
$1 - b_i = P(x_i = 0\|\neg r)$:	probability that term t_i does not occur in a non-relevant document

---

(In literature, you often find p_i and q_i. Leads to confusion with P and q!)

BIRM: Using the notations

$$P(d|r) = \prod_{i=1,n} P(x_i|r) = \prod_{i=1,n} a_i^{x_i}(1-a_i)^{1-x_i}$$

$$P(d|\neg r) = \prod_{i=1,n} P(x_i|\neg r) = \prod_{i=1,n} b_i^{x_i}(1-b_i)^{1-x_i}$$

Example: Document $d =< 0,1,1,0,0,1 >$ and $n = 6$ (6 terms):

$$P(< 0,1,1,0,0,1 > |r) = (1-a1) \cdot a_2 \cdot a_3 \cdot (1-a_4) \cdot (1-a_5) \cdot a_6$$
$$P(< 0,1,1,0,0, >)|\neg r) = (1-b_1) \cdot b_2 \cdot b_3 \cdot (1-b_4) \cdot (1-b_5) \cdot b_6$$

BIRM: The way to the retrieval function $g(d)$

We return now to:

$$\frac{P(d|r) \cdot P(r)}{P(d|\neg r) \cdot P(\neg r)} > 1$$

For a set of documents, $\frac{P(r)}{P(\neg r)}$ is constant, so we only have to deal with:

$$\frac{P(d|r)}{P(d|\neg r)} > 1$$

Using the indepenpendence assumptions, and notations:

$$\frac{\prod_{i=1,n} P(x_i|r)}{\prod_{i=1,n} P(x_i|\neg r)} = \frac{\prod_{i=1,n} a_i^{x_i}(1-a_i)^{1-x_i}}{\prod_{i=1,n} b_i^{x_i}(1-b_i)^{1-x_i}} > 1$$

BIRM: The way to the retrieval function $g(d)$

From the following:

$$\frac{\prod_{i=1,n} a_i^{x_i}(1-a_i)^{1-x_i}}{\prod_{i=1,n} b_i^{x_i}(1-b_i)^{1-x_i}} > 1$$

We take the log:

$$\log \frac{\prod_{i=1,n} a_i^{x_i}(1-a_i)^{1-x_i}}{\prod_{i=1,n} b_i^{x_i}(1-b_i)^{1-x_i}} > \log(1) = 0$$

This gives:

$$\sum_{i=1,n} x_i \log \frac{a_i(1-b_i)}{b_i(1-a_i)} + \sum_{i=1,n} \log \frac{1-a_i}{1-b_i} > 0$$

BIRM: The way to the retrieval function $g(d)$

From:

$$\sum_{i=1,n} x_i \log \frac{a_i(1-b_i)}{b_i(1-a_i)} + \sum_{i=1,n} \log \frac{1-a_i}{1-b_i} > 0$$

we obtain:

$$\boxed{g(d) = \sum_{i=1,n} c_i \cdot x_i + C}$$

where

$$c_i = \log \frac{a_i(1-b_i)}{b_i(1-a_i)}$$

$$C = \sum_{i=1,n} \log \frac{1-a_i}{1-b_i}$$

BIRM: Why such a $g(d)$?

- c_i are weights associated with terms t_i, eg discrimination power.
- Simple addition:
  - for $c_i > 0$, term t_i occurring in document is a good indication of relevance
  - for $c_i < 0$, term t_i occuring in document is a good indication of non-relevance
  - for $c_i = 0$, term t_i occuring in document means nothing
- C constant for all documents given the same query:

  cut-off value
- Retrieval strategy:

  if $g(d) \geq C$ then retrieve d; otherwise do not retrieve d

  or simply rank by $g(d)$ value (ignore C)

BIRM: Estimating c_i

For each term t_i :

	RELEVANT	NON-RELEVANT	
$x_i = 1$	r_i	$n_i - r_i$	n_i
$x_i = 0$	$R - r_i$	$N - n_i - R + r_i$	$N - n_i$
	R	$N - R$	N

n_i: number of documents with term t_i

r_i: number of relevant documents with term t_i

R: number of relevant documents

N: number of documents

These data can be extracted after a relevance feedback process: user points out the relevant documents from a list of retrieved documents.

BIRM: Estimating c_i

- We recall:
  - a_i $(1 - a_i)$: probability that a relevant document contains (does not contain) the term t_i
  - b_i $(1 - b_i)$: probability that a non relevant document contains (does not contain) the term t_i

$$a_i = \frac{r_i}{R} \qquad b_i = \frac{n_i - r_i}{N - R}$$

so

$$c_i = \log \frac{a_i(1 - b_i)}{b_i(1 - a_i)} = \log \frac{r_i/(R - r_i)}{(n_i - r_i)/(N - n_i - R + r_i)}$$

BIRM: Estimating c_i - RSJ weights

$$c_i = \log \frac{r_i/(R - r_i)}{(n_i - r_i)/(N - n_i - R + r_i)}$$

is usually re-written:

$$c_i = \log \frac{(r_i + 0.5)/(R - r_i + 0.5)}{(n_i - r_i + 0.5)/(N - n_i - R + r_i + 0.5)}$$

0.5 is added to keep the c_i value from being infinite when r_i and R are small.

c_i is also referred to as term weight in BIRM; also referred to as Robertson-Spark Jones (RSJ) weights and written $w^{(1)}$.

BIRM: How does it work in practice?

- When no sample is available, R is not known
  - set $a_i = 0.5$ and $b_i = n_i/N$
  - leads to $c_i = \log(N - n_i)/n_i$ which can be viewed as a probabilisitic *idf*
  - $g(d)$ thus with *idf* weights produces initial ranking

  Relevance feedback is then applied, and R, r_i can be defined, which has been shown to improve ranking.

BIRM: Example – Using the original c_i weights

2 terms t_1 and t_2; $d = (x_1, x_2)$; 20 documents $d_1, \ldots d_{20}$; the query is made of term t_1 and t_2

d	Rel	x_1	x_2	d	Rel	x_1	x_2	d	Rel	x_1	x_2
d_1	r	1	1	d_2	r	1	1	d_3	r	1	1
d_4	r	1	1	d_5	$\neg r$	1	1	d_6	r	1	0
d_7	r	1	0	d_8	r	1	0	d_9	r	1	0
d_{10}	$\neg r$	1	0	d_{11}	$\neg r$	1	0	d_{12}	r	0	1
d_{13}	r	0	1	d_{14}	r	0	1	d_{15}	$\neg r$	0	1
d_{16}	$\neg r$	0	1	d_{17}	$\neg r$	0	1	d_{18}	r	0	0
d_{19}	$\neg r$	0	0	d_{20}	$\neg r$	0	0				

$N = 20$; $R = 12$; $r_1 = 8$; $r_2 = 7$; $n_1 = 11$ and $n_2 = 11$

BIRM: Example

$$a_1 = r_1/R = 8/12;\ a_2 = 7/12;$$

$$b_1 = (n_1 - r_1)/(N - R) = (11 - 8)/(20 - 12) = 3/8;\ b_2 = 4/8$$

Thus: (use ln for the logs)

$$c_1 = \log \frac{a_1(1 - b_1)}{b_1(1 - a_1)} = \log 10/3 = 1.20$$

$$c_2 = \log 7/5 = 0.34$$

Retrieval function: $g(D) = 1.20x_1 + 0.34x_2 + C$

BIRM: Example: Result

Retrieval results (here we ignore C):

Rank	Document	$g(d)$
Rank 1	d_1, d_2, d_3, d_4, d_5	1.54
Rank 6	$d_6, d_7, d_8, d_9, d_{10}, d_{11}$	1.20
Rank 12	$d_{12}, d_{13}, d_{14}, d_{15}, d_{16}, d_{17}$	0.34

BIRM: Summary

- Probabilistic model uses probability theory to model the "uncertainty" in the retrieval process.
- Assumptions (here independence assumptions) are made explicit
- Term weight (c_i) without relevance information is inverse document frequency (this can be proven).
- Relevance feedback can improve the ranking by giving better probability estimates of term weights.
- No use of within-document term frequencies or document lengths.

Building on the probabilistic model: Okapi weighting

- Okapi system is based on the probabilistic model
- BIRM does not perform as well as the vector space model
  - does not use term frequency (tf) and document length (dl)
  - hurt performance on long documents
- What Okapi does:
  - add a tf component like in the vector space model
  - separate document and query length normalization
  - several tuning constants, which depend on the collection

BM25 (Best-match Okapi weight)

$$R(d,q) = BM25(d,q) =$$

$$\sum_{t \in q} \left(w_t \cdot \frac{(k_1+1)tf(t,d)}{K + tf(t,d)} \cdot \frac{(k_3+1)tf(t,q)}{k_3 + tf(t,q)} \right) + k_2 \cdot |q| \cdot \frac{avgdl - dl}{avgdl + dl}$$

$$K = k_1((1-b) + (b \cdot dl)/avdl)$$

w_t	term weight based on relevance feedback (RSJ - $w^{(1)}$) or IDF
$tf(t,d), tf(t,q)$	within term frequencies – document and query
k_1, k_2, k_3, b	tuning parameters
$dl, avgdl$	document length and average document length

BM25 – Parameters

$$\sum_{t \in q} \left(w_t \cdot \frac{(k_1+1)tf(t,d)}{K + tf(t,d)} \cdot \frac{(k_3+1)tf(t,q)}{k_3 + tf(t,q)} \right) + k_2 \cdot |q| \cdot \frac{avgdl - dl}{avgdl + dl}$$

$$K = k_1((1-b) + (b \cdot dl)/avdl)$$

- k_1 : governs the importance of within document frequency $tf(t,q)$
- k_2 : compensation factor for the high within document frequency values in large documents
- k_3 : governs the importance of within query frequency $tf(t,q)$
- b : relative importance of within document frequency and document length

The theoretical basis for the Okapi formula is the use of Poisson distributions to model within document frequency in relevant documents, and in non-relevant documents

(not discussed here).

BM25 (Best-match Okapi weight)

Experiments show:
k_2 =0; k_3 large; b closer to 1
Leading for instance to (with $k_1 = 1$ and $b = 0.75$):

$$BM25(d,q) = \sum_{t \in q} \left(w_t \cdot \frac{tf(t,d)}{K + tf(t,d)} \right)$$

$$K = 0.35 + (0.75 \cdot dl)/avdl$$

In experiments, Okapi weights give the best performance.
BM25 often used as baseline model in retrieval experiments.

Summary

- The vector space model is the most basic one.
- The BIRM is one of the important pieces of IR theory.
  - A ranking based on the probability of relevance is optimal with respect to a cost function where the costs for reading relevant documents are low and the costs for reading non-relevant documents are high (probability ranking principle).
- BM25 Okapi model is often the most "effective" model, the model to "beat" in retrieval experiments.
- BM25F (BM25 Field) – take document structure and anchor text into account

Outline

- A recap
- Language model (LM)
- Divergence from randomness model (DFR)

Boolean Model - Recap

No Rank: A document is judged to be relevant if the terms in the document satisfies the logical expression of the query

- A document is represented as a set of keywords (i.e. model of documents)
- Queries are Boolean expressions of keywords, connected by AND, OR, and NOT (i.e. model of queries)

Vector Space Model - Recap

- Rank according to the similarity metric (e.g. cosine) between the query and document.
- The smaller the angle between the document and query the more similar they are believed to be.
  - Documents are represented by a term vector
  - Queries are represented by a similar vector
- Ad-hoc weightings (term frequency x inverse document frequency) used
- No optimal ranking

Binary Independence Retrieval Model

- Rank by the probability of a document being relevant to the query:

$$P(r|d,q)$$

  - Documents are represented by a binary term vector

    Absence or presence of terms
  - We cannot estimate $P(r|d,q)$ directly, so we evoke Bayes' rule, to obtain $P(d|q,r)$, which itself leads to the function $g(d,q)$
- Based on the probability ranking principle, which "ensures" an optimal ranking

- Empirically Based
  - Success measured by experimental results
  - Ad hoc weighting schemes
- Few properties provable
  - Sometimes you want to analyze properties of methods
  - Is similar, relevant?

- Probability Ranking Principle
  - Minimises risk
  - Justifies decision
- Theoretical Framework
  - Nice theoretical properties, but performance benefits are unclear
  - Extensible

Generative Probabilistic Models

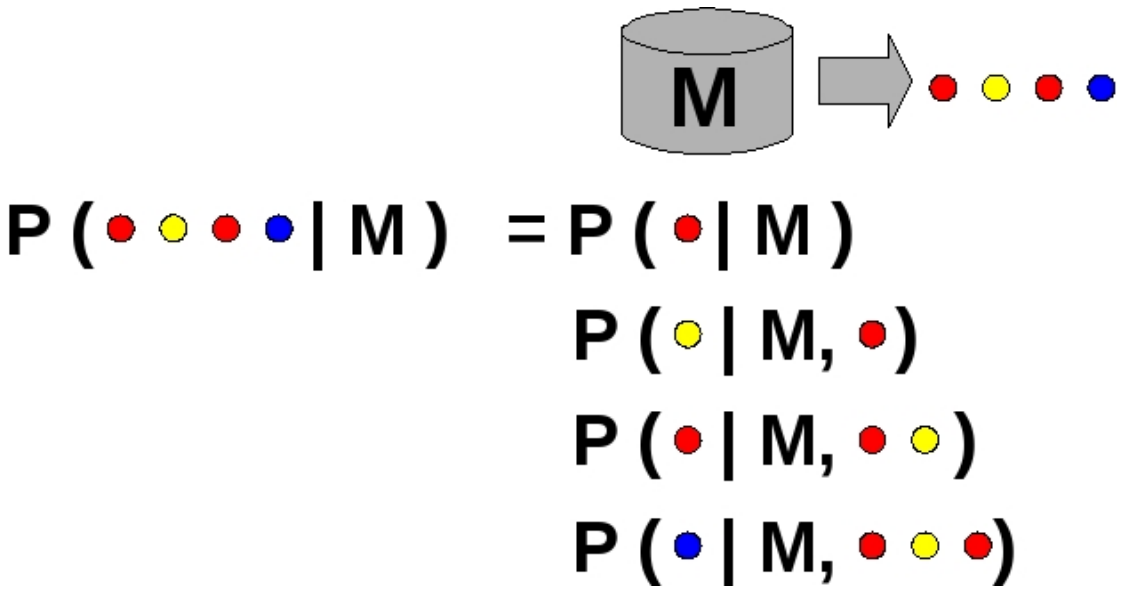
The **generative approach** A generator which produces events/tokens with some probability

- URN Metaphor a bucket of different colour balls (10 red, 5 blue, 3 yellow, 2 white)
  - What is the probability of drawing a yellow ball? $3/20$
  - What is the probability of drawing (with replacement) a red ball and a white ball? $10/20 \times 2/20 = 1/2 \times 1/10$
  - What is the probability of generating the sequence red ball and white ball (with replacement)?
- IR Metaphor: Documents are urns, full of tokens (balls) of (in) different terms (colours)

Generative Models - Language model

A statistical model for generating data

- Probability distribution over samples for a given **language**
- M $\rightarrow t_1 t_2 t_3 t_4$



Generative Probabilistic Models

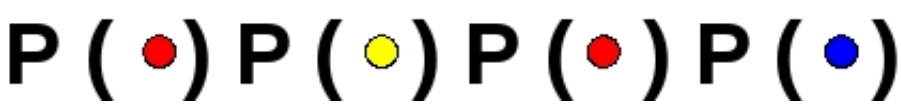
- What is the probability of producing the query from a document? $P(q|d)$
  - Referred to as the query-likelihood
- Assumptions:
  - The probability of a document being relevant is strongly correlated with the probability of a query given a document, i.e. $P(d|r)$ is correlated with $P(q|d)$
- System's task is to estimate for each of the documents in the collection, which one is the most likely to generate the query.

Language Models in IR (1998)

- Let us assume we point blindly, one at a time, at 3 words in a document
- What is the probability that I, by accident, pointed at the words "Master", "computer", and "Science"?
- Compute the probability, and use it to rank the documents.

Types of language models

- Unigram Models (Assume word independence)



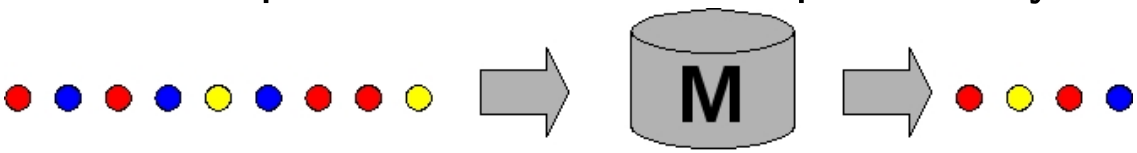
- Bigram Models

- There are others . . .

The fundamental problem

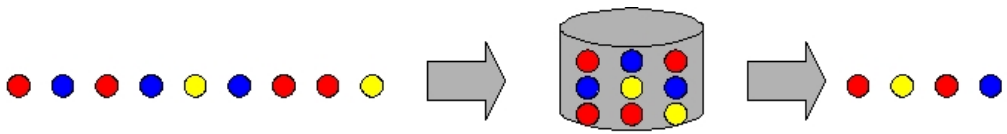
- Usually we do not know the model **M**, but have a sample representative of that model

- First estimate a model from a sample
- Then compute the observation probability



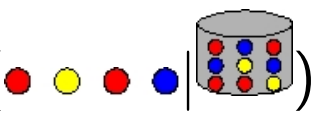
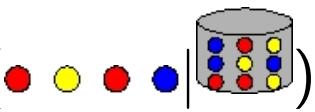
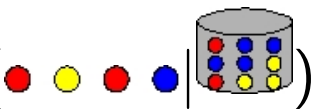
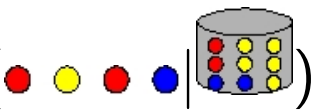
Example for unigram models

(Urn metaphor)



- P(● ● ● ●) ~ P (●) P (●) P (●) P (●)
  = 4/9 * 2/9 * 4/9 * 3/9

Example - Ranking documents with unigram models

- Rank models (documents) by probability of generating the query
- Q: 
- $P($  $) = 4/9 * 2/9 * 4/9 * 3/9 = 96/6561$
- $P($  $) = 3/9 * 3/9 * 3/9 * 3/9 = 81/6561$
- $P($  $) = 2/9 * 3/9 * 2/9 * 4/9 = 48/6561$
- $P($  $) = 2/9 * 5/9 * 2/9 * 2/9 = 40/6561$

Standard LM Approach

- Assume that query terms are drawn identically and independently from a document (unigram models)

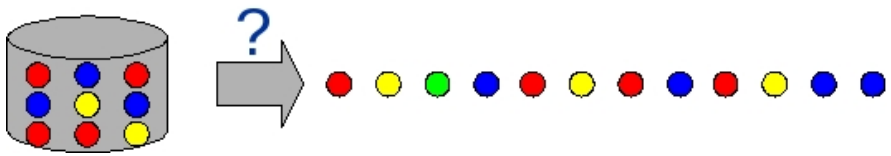
$$P(q|d) = \prod_{t \in q} P(t|d)^{n(t,q)}$$

(where $n(t, q)$ is the number of term t in query q)

- Maximum Likelihood Estimate of $P(t|d)$
  - Simply use the number of times the query term occurs in the document divided by the total number of term occurrences.
- Problem: **Zero Probability (frequency) Problem**

The Zero-frequency Problem

- Suppose some event not in our example
  - Model will assign zero probability to that event
  - And to any set of events involving the unseen event
- Happens frequently with language
- It is incorrect to infer zero probabilities
  - Especially when dealing with incomplete samples



Document Models

- Solution:
  - Infer a language model (θ_d) for each document, where $P(t|\theta_d) > 0$ for all t
  - Then we can estimate $P(q|\theta_d)$
- Standard approach is to use the probability of a term $p(t)$ to smooth the document model, thus

$$P(t|\theta_d) = \lambda P(t|d) + (1 - \lambda)P(t)$$

Estimating Document Models

- Basic Components
  - Probability of a term given a document (maximum likelihood estimate)

$$P(t|d) = \frac{n(t,d)}{\sum_{t'} n(t',d)}$$

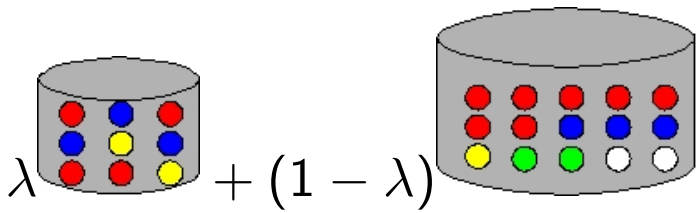
  - Probability of a term given the collection

$$P(t) = \frac{\sum_{d} n(t,d)}{\sum_{t'} \sum_{d'} n(t',d')}$$

  - $n(t,d)$ is the number of times term t occurs in document d

Smoothing

- Idea: shift part of probability mass to unseen events
- Interpolation with background (General English in our case)
  - Reflects expected frequency of events
  - Plays role of IDF in LM



Estimating Document Models

- Example of Smoothing methods
  - Laplace

$$P(t|\theta_d) = \frac{n(t,d) + \alpha}{\sum_{t'} n(t',d) + \alpha|T|}$$

    $|T|$ is the number of term in the vocabulary
  - Jelinek-Mercer

$$P(t|\theta_d) = \lambda \cdot P(t|d) + (1-\lambda) \cdot P(t)$$

  - Dirichlet

$$P(t|\theta_d) = \frac{|d|}{|d|+\mu} \cdot P(t|d) + \frac{\mu}{|d|+\mu} \cdot P(t)$$

Language Models - Implementation

We assume the following LM (**Jelinek-Mercer** smoothing):

$$P(q = t_1, t_2, \ldots, t_n|d) = \prod_{i=1}^{n}((1-\lambda) \cdot P(t_i) + \lambda \cdot P(t_i|d))$$

It can be shown that the above leads to:

$$P(q = t_1, t_2, \ldots, t_n|d) \approx \sum_{i=1}^{n} \log(1 + \frac{\lambda \cdot P(t_i|d)}{(1-\lambda) \cdot P(t_i)})$$

for ranking purpose (again use log to obtain summation)

Document Priors

- Remember $P(d|q) = P(q|d)P(d)/P(q) \approx P(q|d)P(d)$
- $P(d)$ is typically assumed to be uniform so is usually ignored leading to $P(d|q) \approx P(q|d)$
- $P(d)$ provides an interesting avenue for encoding a priori knowledge about the document
  - Document length (longer doc $\rightarrow$ more relevant)
  - Average Word Length (bigger words $\rightarrow$ more relevant)
  - Time of publication (newer doc $\rightarrow$ more relevant)
  - Number of web links (more in links $\rightarrow$ more relevant)
  - PageRank (more popular $\rightarrow$ more relevant)

"Language Modelling"

- Not just "English"
- But also, the *language* of
  - author
  - newspaper
  - text document
  - image
  - structure
  - ...

Summary LM

- Approach based on "probability" of relevance (like BIRM) but RSV is based on $P(q|d)$ (ranking can be done in terms of $P(d|q)$) and not $P(d|q,r)$
- Based on the probability that a term occurs in a sequence of terms.
- BIRM is based on the probability that term does or does not occur in a set of (retrieved) documents

Binomial Distribution

$$P(n) = \binom{N}{n} \cdot p^n \cdot (1-p)^{N-n}$$

Imagine you go on a sailing trip on the East Coast of England. Every second day, there is a beautiful sunset, i.e. $p = 1/2$. You go sailing for a week ($N = 7$). What is your chance to have exactly three ($n = 3$) beautiful sunset?

$$\binom{7}{3} \cdot p^3 \cdot (1-p)^{7-3} = 0.2734$$

Divergence from Randomness (DFR)

- See http://terrier.org/docs/v2.2.1/dfr_description.html
- Basic idea: "The more the divergence of the within-document term frequency from its frequency within the collection, the more divergent from randomness the term is, meaning the more the information carried by the term in the document."

$$weight(t|d) \propto -\log P_M(t \in d|collection)$$

M stands for the type of model of the divergence from randomness employed to compute the probability.
In the next slide, the binomial distribution (B) is used as the model of the divergence from randomness.

Binomial Distribution as Randomness Model

TF	Term frequency of term t (occurrence of t) in the collection
tf	Term frequency of term t in the document d
p	Probability to draw a document ($p = 1/N$, N is number of documents)

$$-\log P_B(t \in d|collection) = -\log \binom{TF}{tf} \cdot p^{tf} \cdot (1-p)^{TF-tf}$$

The probability that

the event (that occurs with probability p) occurs tf times in TF trials
a document occurs tf times in TF trials
a sunny day (which occurs with $1/N$) occurs on tf days in a TF days holiday

Binomial Distribution as Randomness Model

$$-\log P_B(t \in d|collection) = -\log \binom{TF}{tf} \cdot p^{tf} \cdot (1-p)^{TF-tf}$$

If N is the number of documents, then $TF/N = \lambda$ is the average occurrence of term t.

The above is minimum for $tf = \lambda$, meaning that term t has a random distribution – its distribution does not diverge from randomness – and as such is not informative.

Outline

- Query reformulation
- Relevance feedback
- Local analysis (also called pseudo-relevance feedback)
- Global analysis

Query reformulation

- No detailed knowledge of collection and retrieval environment
  - Difficult to formulate queries well designed for retrieval
  - Need many formulations of queries for good retrieval
- First formulation is usually a naive attempt to retrieve relevant information
- The idea is when documents are initially retrieved:
  - They should be examined for relevance information
  - Then we can improve the query for retrieving additional relevant documents
- Query reformulation:
  - Expanding original query with new terms
  - Reweighing the terms in the (expanded) query

Three approaches

- Relevance feedback
  - Approaches based on feedback from users
  - E.g. Rocchio, Ide, Binary Independence Model (covered in Retrieval Models I)
- Local analysis (also called pseudo-relevance feedback)
  - Approaches based on information derived from the set of initially retrieved documents (local set of documents)
- Global analysis
  - Approaches based on global information derived from the document collection

Relevance feedback

- Relevance feedback can be viewed as an iterative cycle:
  1. User presented with a list of retrieved documents
  2. User marks those which are relevant (or not relevant)
     - In practice: top 10-20 ranked documents are examined
     - Incremental: one document after the other
  3. The relevance feedback algorithm selects important terms from documents assessed relevant by users
  4. The relevance feedback algorithm emphasises the importance of these terms in a new query in the following ways:
     - Query expansion: add these terms to the query
     - Term reweighing: modify the term weights in the query
     - Query expansion + term reweighing
  5. The updated query is submitted to the system
  6. If the user is satisfied with the new set of retrieved documents, then the relevance feedback process stops, otherwise go to step 2

Relevance feedback - continued

- The approaches are (to recap):
  - Approach 1: Add/Remove/Change query terms
  - Approach 2: Re-weight query terms
  - Note: Approach 1 and 2 are not really different conceptually. Why?

Relevance Feedback — Vector-Space Model

Reformulation of a query moves the query vector closer to relevant documents, closer to the optimal query.

C: set of documents (the collection)
R: set of relevant documents in the collection

$$\vec{q}_{opt} = \frac{1}{|R|} \sum_{d_i \in R} \vec{d}_i - \frac{1}{|C| - |R|} \sum_{d_i \notin R} \vec{d}_i$$

Do you know R?

Relevance Feedback — Rochio

You need to use an estimate for R.

D_r: set of relevant and retrieved documents

D_n: set of non-relevant and retrieved documents

$$\vec{q}_{next} = \alpha \cdot \vec{q}_{prev} + \beta \cdot \frac{1}{|D_r|} \sum_{d_i \in D_r} \vec{d}_i - \gamma \cdot \frac{1}{|D_n|} \sum_{d_i \in D_n} \vec{d}_i$$

- Known as the Rocchio formula
- The factors α, β, γ control the effect of previous query, relevant documents and non-relevant documents on the new query

Relevance Feedback — Rochio

$$\vec{q}_{next} = \alpha \cdot \vec{q}_{prev} + \beta \cdot \frac{1}{|D_r|} \sum_{d_i \in D_r} \vec{d}_i - \gamma \cdot \frac{1}{|D_n|} \sum_{d_i \in D_n} \vec{d}_i$$

- Usually information in relevant documents more important than in non-relevant documents ($\gamma << \beta$).
- Positive relevance feedback ($\gamma = 0$) is when we only extract information from documents assessed relevant.
- α emphasises the importance of the original query ($\vec{q}_{prev}$).

Relevance Feedback — Rochio in practice

$$\vec{q}_{next} = \vec{q}_{prev} + \beta \cdot \frac{1}{|D_r|} \sum_{d_i \in D_r} \vec{d}_i - \gamma \cdot \frac{1}{|D_n|} \sum_{d_i \in D_n} \vec{d}_i$$

- $\alpha = 1$
- Terms forming the reformulated query ($\vec{q}_{prev}$) are those:
  - in the original query,
  - that appear in more relevant documents that non-relevant documents
  - that appear in more than half of the relevant documents
- Negative weights ignored

Relevance Feedback — Ide

$$\vec{q}_{next} = \alpha \cdot \vec{q}_{prev} + \beta \cdot \sum_{d_i \in D_r} \vec{d}_i - \gamma \cdot \sum_{d_i \in D_n} \vec{d}_i$$

Start with $\alpha = \beta = \gamma = 1$.

The cardinalities of the sets of relevant and non-relevant documents are not considered.

Issues in relevance feedback

- Relevance feedback
  - Often users are not reliable in making relevance assessments, or do not make relevance assessments
  - Implicit relevance feedback by looking at what users access
    - clicks in web logs (see lecture on web search)
    - it works well because of the "wisdom of the crowd"
  - Positive, negative
  - Partial relevance assessments (e.g very relevant or partially relevant)?
  - Why is a document relevant?
- Interactive query expansion (as opposed to automatic)
  - Users choose the terms to be added

Local analysis

- Examine documents retrieved for query to determine query expansion with no user assistance
- Two strategies are used to add terms to the query:
  - Local clustering (terms that are synonyms, stemming variations)
  - Local context analysis (terms close to query terms in text - proximity of terms in text)
  - (BUT ALSO session analysis (queries used in same sessions as analyzed from logs) for query recommendation/suggestion)
- Two issues:
  - Query drift: if top documents are not that relevant, the reformulated query may not reflect the user information need
  - Computation cost high since must be done at retrieval time (on-line)

Global analysis

- Expand query using information from whole set of documents in collection
- No user assistance
- Make use of of a global thesaurus that is build based on the document collection.
- Two issues:
  - Approach to build thesaurus (e.g. term co-occurrence)
  - Approach to select terms for query expansion (e.g. the top 20 terms ranked according to IDF value)
- (BUT ALSO session analysis (queries used in same sessions as analyzed from logs) for query recommendation/suggestion)

Outline

- Other models
  - Inference network model
  - Extended boolean model
  - Fuzzy model
  - Page rank model
  - Uncertain inference (logical) model

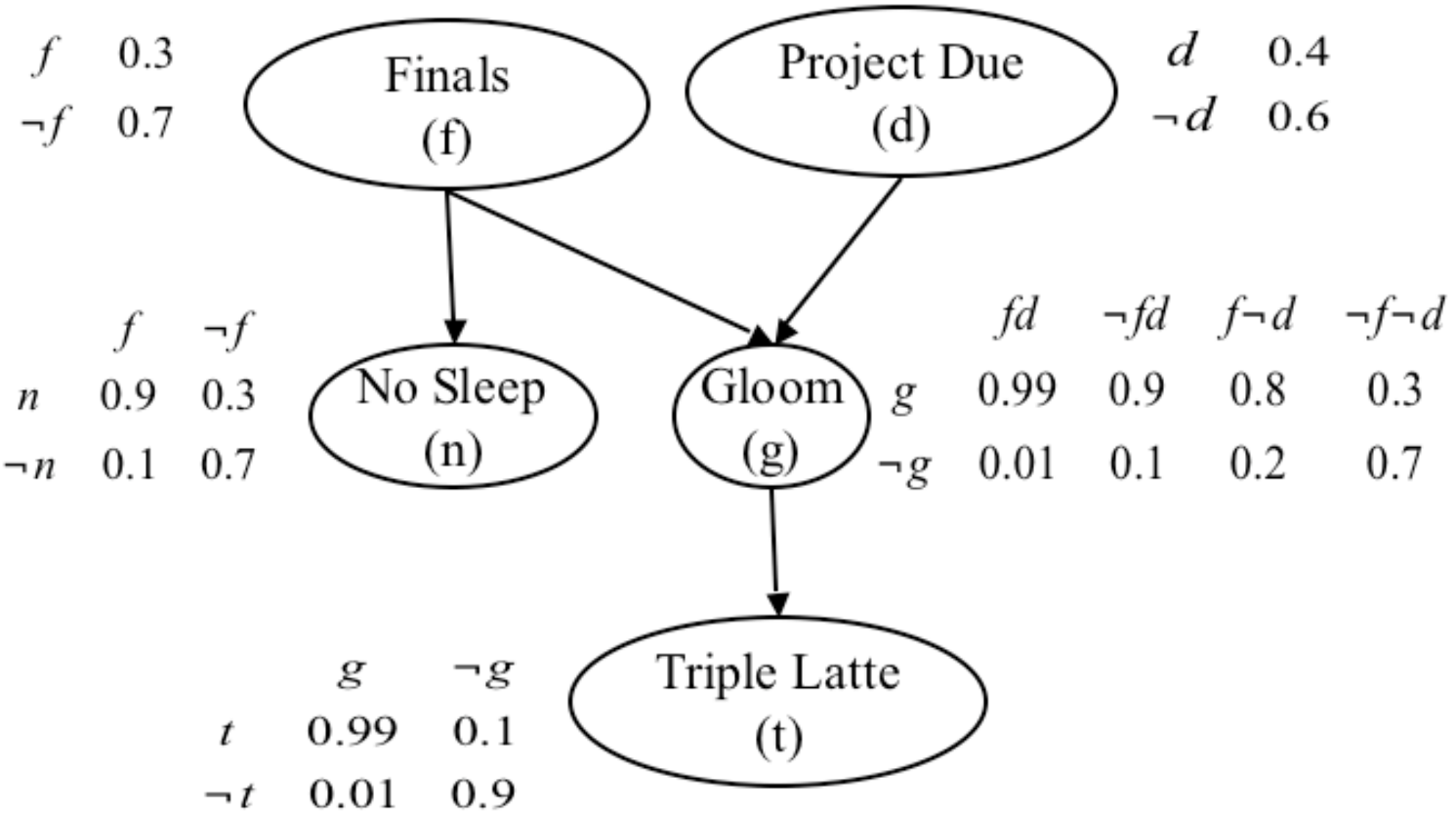
Inference Network Model

- Information Retrieval using inference networks
  Greatest strength for IR is in providing a framework permitting combination of multiple distinct evidence sources to support a relevance judgement (probability) on a given document.
  - Bayesian networks
    - Model causal relationship between events
    - Infer the belief that an event holds based on observations of other events
  - Turtle and Croft Inference Model
  - Belief model

We only discuss the inference model, which is a simplified version of a Bayesian network for IR.

Note that the "combination of evidence is now dealt mostly with machine learning approaches (features and parameters) = learning to rank.

Bayesian Network – A toy example



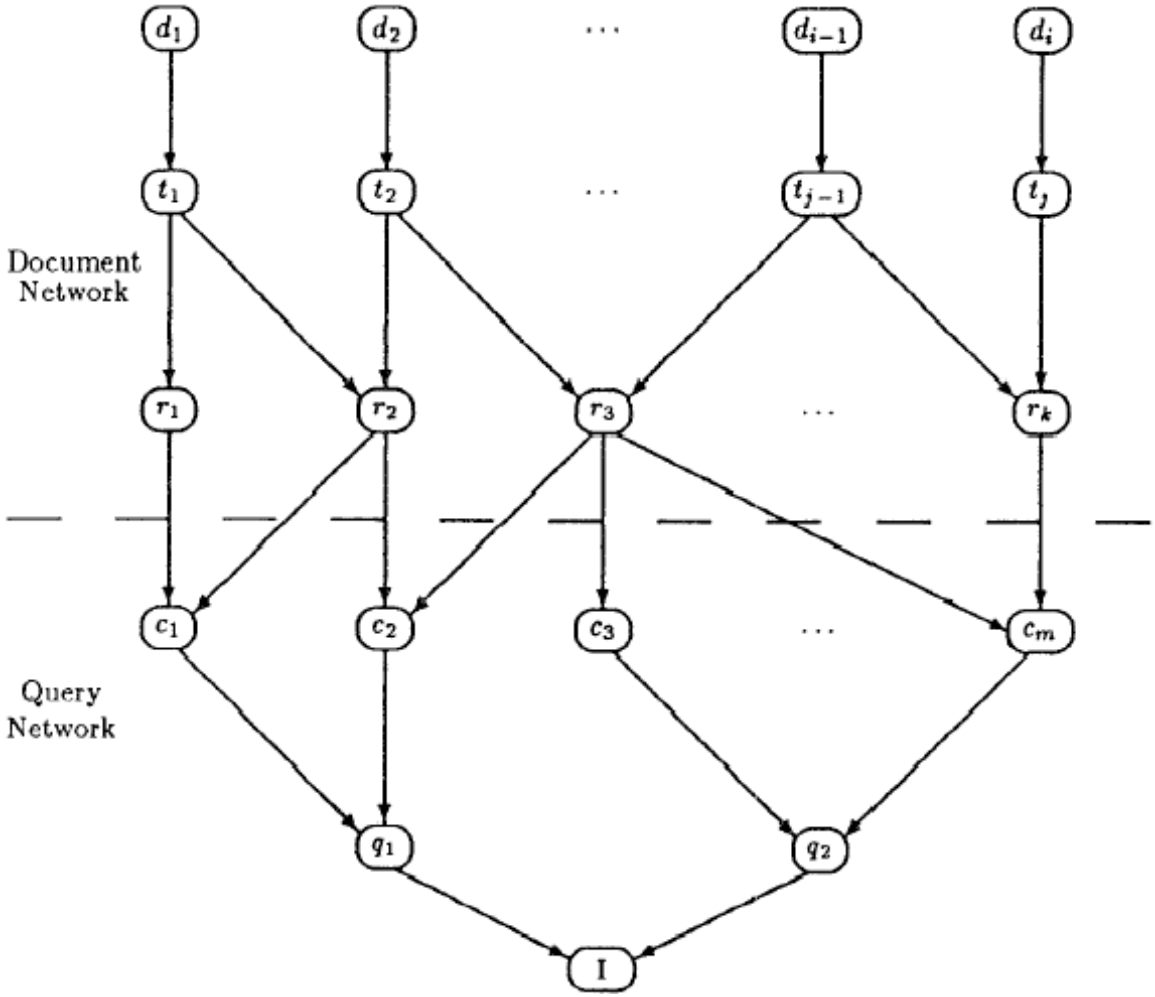
Bayesian Network – Links at dependency

- Link Matrix
  - Attached to each node
    - Give influences of parents on that node.
  - Nodes with no parent get a prior probability: e.g., f, d.
  - Interior node : conditional probability of all combinations of values of its parents: e.g., n, g, t.
- Make use of independence assumptions to make computation tractable
  - Nodes not connected by a link: no direct conditioning.
- Make use of link matrices (discussed later).

Inference Network

- Bayesian Network used to model documents, the document contents, and the query.
- Two sub-networks:
  - the Document Network (DN) produced during indexing and then static during retrieval;
  - the Query Network (QN) produced from the query text during retrieval.

Inference Network



Document network

- document nodes:
  - nodes for each document
  - retrievable units, i.e. those items we wish to see in the resultant ranking
- document concept nodes:
  - nodes for each concept with the collection
- causal link ($\rightarrow$) between document node and document concept node:
  - document content is represented by concept
  - a conditional probability, or weight, to indicate strength of link
- evaluation of a node on the basis of value of the parent nodes and the conditional probabilities.

Document network - Probabilities

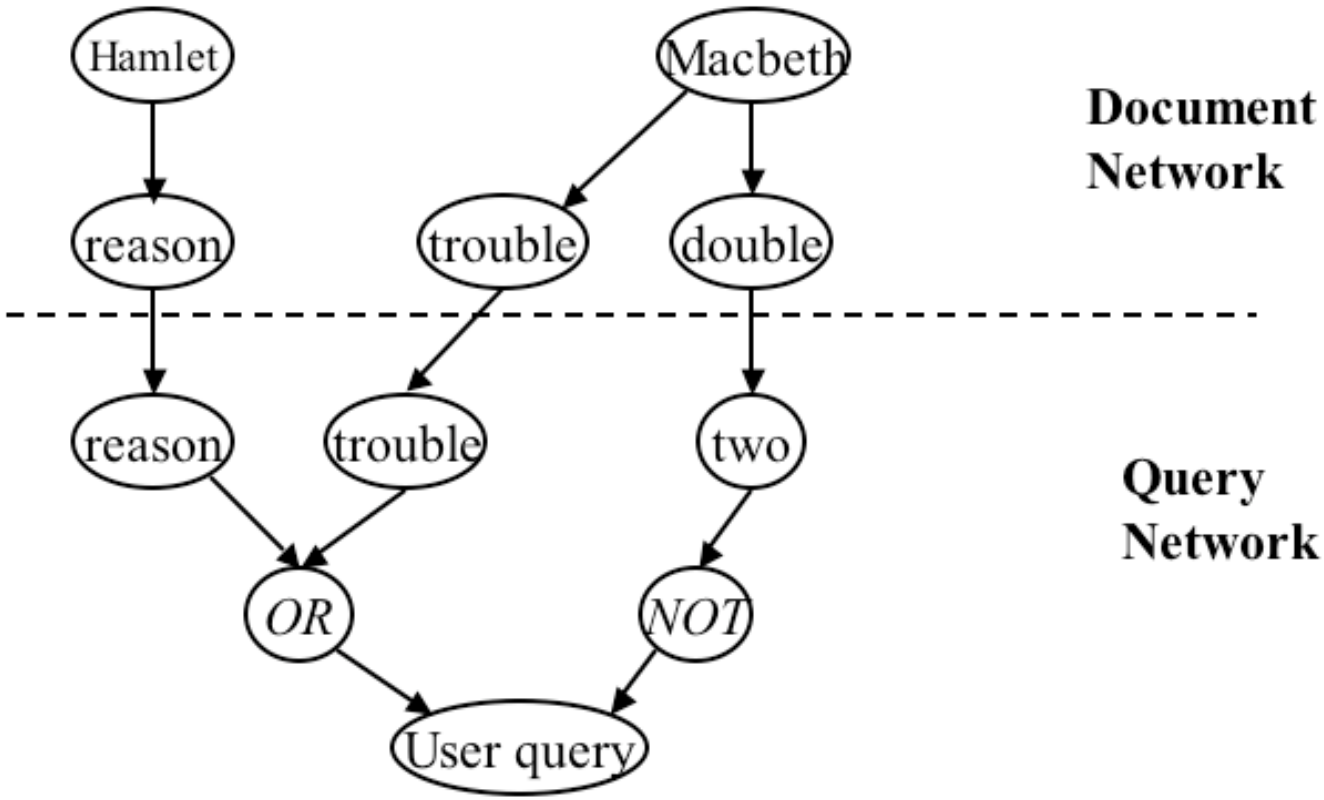
- document nodes:
  - prior document probability
  - usually set to $P(d) = \frac{1}{n}$ for n the number of documents.
- document concept nodes:
  - usually terms, so $r_i = t_i$, or can make use of a thesaurus (for related terms)
  - conditional probability $P(t_i|d)$ based on within document frequency, $tf \times idf$, etc.

Query network

- represent the submitted query
  - query concept nodes: framework of nodes that represent the required concepts
  - query operator nodes, e.g. #and, #not, etc which permits statistical operators and statistical approximations of the Boolean operators
  - user information need node

#and	AND the terms
#or	OR the terms
#not	Negate the term (incoming belief)
#sum	Sum of the incoming beliefs
#wsum	Weighted sum of the incoming beliefs
#max	Maximum of the incoming beliefs

Example


Hamlet
Macbeth
reason
trouble
double
reason
trouble
two
OR
NOT
User query
Document Network
Query Network

Link Matrix

Computation of $P(I|d)$ by propagation of probabilities in the network.
We assume two concepts (terms), t_1 and t_2
Link matrix L.

$$\begin{pmatrix} P(I|d) \\ P(\bar{I}|d) \end{pmatrix} = L \cdot \begin{pmatrix} P(t_1, t_2|d) \\ P(t_1, \bar{t_2}|d) \\ P(\bar{t_1}, t_2|d) \\ P(\bar{t_1}, \bar{t_2}|d) \end{pmatrix}$$

$$L_{\#or} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Link Matrix

Link matrix L contains probabilities of the form

$$\begin{pmatrix} P(I|t_1,t_2) & P(I|t_1,\bar{t_2}) & P(I|\bar{t_1},t_2) & P(I|\bar{t_1},\bar{t_2}) \\ P(\bar{I}|t_1,t_2) & P(\bar{I}|t_1,\bar{t_2}) & P(\bar{I}|\bar{t_1},t_2) & P(\bar{I}|\bar{t_1},\bar{t_2}) \end{pmatrix}$$

A special setting of L with $w_i, i = 1,2$ be query term weights, and

$$w_s = w_1 + w_2$$

$$\begin{pmatrix} \frac{w_1+w_2}{w_s} & \frac{w_1}{w_s} & \frac{w_2}{w_s} & 0 \\ 0 & \frac{w_2}{w_s} & \frac{w_1}{w_s} & \frac{w_1+w_2}{w_s}) \end{pmatrix}$$

Probabilistic Inference Network (PIN) Model

- Special setting of the link matrix using weights $w_s = w_1 + w_2$ for two query terms leads to:
- (Here q, is what is called I in this model, as there can be several queries, or variants, or sub-parts)

$$R(d,q) = P(I|d) = \frac{1}{w_s} \cdot (w_1 \cdot P(t_1|d) + w_2 \cdot P(t_2|d))$$

- Complexity linear with number of query terms.
- System: INQUERY

Page rank

PR	page rank reflecting its "authority", which can be used to rank page
u, v	two pages
U	set of pages
$N(v)$	number of outgoing links in page v

$$PR(u) := d + (1 - d) \cdot \sum_{v \to u} \frac{PR(v)}{N(v)}$$

- the summation is over the set of pages v that have a link to page u
- damping factor reflection the probability that a random (web) surfer arrives at a page
- page rank is used in Google

(See lecture on Web Search)

Fuzzy set model ("vagueness")

- Query term = fuzzy set
- Document = has a membership (between 0 and 1) to that set
- **Fuzzy set theory:**
  - set A (in universe U) whose boundaries are not well defined
  - (e.g., tall, nice, relevant)
  - membership function $\mu_A : U \mapsto [0, 1]$

Fuzzy Model

- Set-membership function: $\mu_A : U \rightarrow [0; 1]$ where A is a term and U is a set of documents.
  - sailing = { (0.9, d1), (0.8, d2) }
  - boats = { (0.5, d1), (0.8, d2) }
- Membership based on tf, idf, or correlation matrix C:
  - $C_{i,j} := \frac{n_{i,j}}{n_i + n_j - n_{i,j}}$
  - $n_{i,j}$: Number of documents in which t_i and t_j occur
  - n_i: Number of documents in which t_i occurs
  - $\mu_i(d) := 1 - \prod_{t_j \in d}(1 - C_{i,j})$
- The weight of a document in the set of term t_i is computed as the disjunction of all document terms related to term t_i.

Fuzzy set model

$$\mu_i(d) := 1 - \prod_{t_j \in d} (1 - C_{i,j})$$

- Document d belongs to fuzzy set (term) t_i if its own terms (the t_js) are related to t_i:
  - one term t_j in document d very related to t_i ($C_{ij} \sim 1$):
    membership of d to fuzzy set t_i close to 1 ($\mu_i(d) \sim 1$)
  - no term t_j in document d related to t_i (all $C_{ij} \sim 0$):
    membership of d to fuzzy set t_i close to 0 ($\mu_i(d) \sim 0$)

Fuzzy Model

- Retrieval is based on fuzzy operations:
  - if query is q="A OR B": $R(d,q) = \max(\mu_A(d), \mu_B(d))$
  - if query is q="A AND B": $R(d,q) = \min(\mu_A(d), \mu_B(d))$
  - if query is q="A AND NOT B":
    $R(d,q) = \max(\mu_A(d), \mu_B(d)) - \min(\mu_A(d), \mu_B(d))$

Extended Boolean model

- Deal with problems:
  - t_1 AND t_2 (too few documents retrieved) $\rightarrow$ affect recall
  - t_1 OR t_2 (too many documents retrieved) $\rightarrow$ affect precision
- Use weights $d = (w_1, w_2)$ for terms t_1 and t_2
- OR-queries:

$$R(d, q) = \sqrt{\frac{w_1^2 + w_2^2}{2}}$$

- AND-queries:

$$R(d, q) = 1 - \sqrt{\frac{(1 - w_1)^2 + (1 - w_2)^2}{2}}$$

Extended Boolean model (Examples)

- t_1 and t_2 in document ($w_1 = 1$ and $w_2 = 1$):
  OR-query $= 1$
  AND-query $= 1$
- t_1 and t_2 not in document ($w_1 = 0$ and $w_2 = 0$):
  OR-query $= 0$
  AND-query $= 0$
- t_1 in document and t_2 not in document ($w_1 = 1$ and $w_2 = 0$):
  OR-query $= 1/\sqrt{2} = 0.707 < 1$
  AND-query $= 1 - 1/\sqrt{2} = 0.29 > 0$

Uncertain Inference (Logical) Model

- Model the notion of relevance as inference.
- Document d is relevant to query $q \iff d \rightarrow q$
- The inference will be uncertain, because the IR process is uncertain (based on estimates)

IR as an inference

- document and query: logical formulae d and q
- retrieval: search for document which implies the query: $d \rightarrow q$

	logical view
$d = \{t_1, t_2, t_3\}$ $q = \{t_1, t_3\}$	$d = t_1 \wedge t_2 \wedge t_3$ $q = t_1 \wedge t_3$
$d \rightarrow q$	

- Advantage: from term-based retrieval to knowledge-based retrieval
- d = square, q = rectangle, and thesaurus: square $\rightarrow$ rectangle: document d relevant to query q

IR as an uncertain inference

- document and query: logical formulae d and q
- d = quadrangle, q = rectangle
- document d maybe relevant to query q:
- uncertain knowledge is required:

  quadrangle, $\rightarrow$ rectangle with uncertainty 0.3
- Retrieval: Estimating the probability that document infer the query:

$$R(d, q) = P(d \rightarrow q)$$

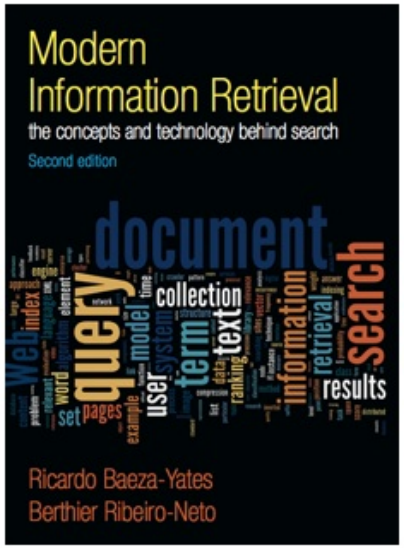
Uncertain Inference Model

- What is the difference between $P(d \rightarrow q)$ and $P(q \rightarrow d)$?
  - View d and q as conjunctions of the terms.
  - $P(d \rightarrow q) = 1$, if d contains *all* query terms. q is a subset of d; d is exhaustive.
  - $P(q \rightarrow d) = 1$, if d contains *only* query terms. d is a subset of q. d is specific.
- Lots of work 1990-2000 (Logic-based IR, FERMI EU project)
- Direct application to XML retrieval (exhaustivity and specificity-based definition relevance at INEX)

- What is Information Retrieval
- TF-IDF
- Precision and Recall
- Retrieval Models
  - **Classical Models** (Bolean, TF-IDF and Length Normalization, Vector Space, Probabilistic Model (BIRM))
  - **Alternative Set Theoretic Models** (Set-based, Extended Boolean, Fuzzy Set)
  - **Alternative Algebraic Models** (Generalized Vector Space, Latent Semantic Indexing (LSI), Neural Network)
  - **Alternative Probabilistic Models** (BM25, Language Models, Divergence from Randomness (DFR), Bayesian Network)
- Other models (XML, Multimedia, Enterprise, Web, etc)

These lectures are based on lecture and presentation slides of the following persons, who are greatly acknowledged:

- Thomas Roelleke, Queen Mary University of London
- Arjen de Vries, CWI Amsterdam and University of Delf
- Iadh Ounis, University of Glasgow
- Gianni Amati, Fondazione Ugo Bordoni, Rome



- Introduction (Chapter 1)
- IR models (Chapter 3)
- Query reformulation and relevance feedback (Chapter 5)
- Evaluation - brief (Chapter 4)
- Indexing - brief (Chapter 6)
- TF-IDF (Chapter 3)

Aizawa, A. (2003).

An information-theoretic perspective of tf-idf measures.
Information Processing and Management, 39:45–65.

Amati, G. and Rijsbergen, C. J. (2002).

Term frequency normalization via Pareto distributions.
In Crestani, F., Girolami, M., and Rijsbergen, C. J., editors, *24th BCS-IRSG European Colloquium on IR Research, Glasgow, Scotland*.

Amati, G. and van Rijsbergen, C. J. (2002).

Probabilistic models of information retrieval based on measuring the divergence from randomness.
ACM Transaction on Information Systems (TOIS), 20(4):357–389.

Baeza-Yates, R. and Ribeiro-Neto, B. (2011).

Modern Information Retrieval - the concepts and technology behind search.
Addison Wesley.

Belew, R. K. (2000).

Finding out about.
Cambridge University Press.

Bookstein, A. (1980).

Fuzzy requests: An approach to weighted Boolean searches.
Journal of the American Society for Information Science, 31:240–247.

Brin, S. and Page, L. (1998).

The anatomy of a large-scale hypertextual web search engine.
Computer Networks, 30(1-7):107–117.

Church, K. and Gale, W. (1995).

Inverse document frequency (idf): A measure of deviation from Poisson.
In *Proceedings of the Third Workshop on Very Large Corpora*, pages 121–130.

Cooper, W. (1991).

Some inconsistencies and misnomers in probabilistic IR.
In Bookstein, A., Chiaramella, Y., Salton, G., and Raghavan, V., editors, *Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 57–61, New York.

Cooper, W. S. (1988).
Getting beyond Boole.
Information Processing and Management, 24(3):243–248.

Cooper, W. S. (1994).
Triennial acm sigir award presentation and paper: The formalism of probability theory in ir: A foundation for an encumbrance.
In [Croft and van Rijsbergen, 1994], pages 242–248.

Craswell, N., Hawking, D., and Robertson, S. E. (2001).
Effective site finding using link anchor information.
In *SIGIR*, pages 250–257.

Crestani, F. and van Rijsbergen, C. J. (1995).
Probability kinematics in information retrieval.
In Fox, E., Ingwersen, P., and Fidel, R., editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 291–299, New York. ACM.

Croft, B. and Lafferty, J., editors (2003).
Language Modeling for Information Retrieval.
Kluwer.

Croft, W. and Harper, D. (1979).
Using probabilistic models of document retrieval without relevance information.
Journal of Documentation, 35:285–295.

Croft, W. and Turtle, H. (1992).
Retrieval of complex objects.
In Pirotte, A., Delobel, C., and Gottlob, G., editors, *Advances in Database Technology — EDBT'92*, pages 217–229, Berlin et al. Springer.

Croft, W. B. and van Rijsbergen, C. J., editors (1994).

Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, London, et al. Springer-Verlag.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990).

Indexing by latent semantic analysis.

Journal of the American Society for Information Science, 41(6):391–407.

Dumais, S. T., Furnas, G. W.and Landauer, T. K., and Deerwester, S. (1988).

Using latent semantic analysis to improve information retrieval.

pages 281–285.

Fang, H. and Zhai, C. (2005).

An exploration of axiomatic approaches to information retrieval.

In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 480–487, New York, NY, USA. ACM.

Fuhr, N. (1992).

Probabilistic models in information retrieval.

The Computer Journal, 35(3):243–255.

Grossman, D. A. and Frieder, O. (1998).

Information Retrieval: Algorithms and Heuristics.

Kluwer, Massachusetts.

Grossman, D. A. and Frieder, O. (2004).

Information Retrieval. Algorithms and Heuristics, 2nd ed., volume 15 of *The Information Retrieval Series*.

Springer.

He, B. and Ounis, I. (2005).

Term frequency normalisation tuning for BM25 and DFR models.

In *ECIR*, pages 200–214.

Hiemstra, D. (2000).

A probabilistic justification for using tf.idf term weighting in information retrieval.
International Journal on Digital Libraries, 3(2):131–139.

Joachims, T. (2000).

Estimating the generalization performance of an svm efficiently.
In [Langley, 2000], pages 431–438.

Kleinberg, J. (1999).

Authoritative sources in a hyperlinked environment.
Journal of ACM, 46.

Klinkenberg, R. and Joachims, T. (2000).

Detecting concept drift with support vector machines.
In [Langley, 2000], pages 487–494.

Lafferty, J. and Zhai, C. (2003).

Probabilistic Relevance Models Based on Document and Query Generation, chapter 1.
In [Croft and Lafferty, 2003].

Langley, P., editor (2000).

Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Standord, CA, USA, June 29 - July 2, 2000. Morgan Kaufmann.

Lavrenko, V. and Croft, W. B. (2001).

Relevance-based language models.
In *SIGIR*, pages 120–127.

Luk, R. W. P. (2008).

On event space and rank equivalence between probabilistic retrieval models.
Inf. Retr., 11(6):539–561.

Manning, C. D., Raghavan, P., and Schuetze, H., editors (2008).

Introduction to Information Retrieval.
Cambridge University Press.

Margulis, E. (1992).

N-poisson document modelling.
In Belkin, N., Ingwersen, P., and Pejtersen, M., editors, *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 177–189, New York.

Maron, M. and Kuhns, J. (1960).

On relevance, probabilistic indexing, and information retrieval.
Journal of the ACM, 7:216–244.

Meghini, C., Sebastiani, F., Straccia, U., and Thanos, C. (1993).

A model of information retrieval based on a terminological logic.
In Korfhage, R., Rasmussen, E., and Willett, P., editors, *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 298–308, New York. ACM.

Metzler, D. and Croft, W. B. (2004).

Combining the language model and inference network approaches to retrieval.
Information Processing & Management, 40(5):735–750.

Nie, J. (1992).

Towards a probabilistic modal logic for semantic-based information retrieval.
In Belkin, N., Ingwersen, P., and Pejtersen, M., editors, *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 140–151, New York.

Ponte, J. and Croft, W. (1998).

A language modeling approach to information retrieval.
In Croft, W. B., Moffat, A., van Rijsbergen, C. J., Wilkinson, R., and Zobel, J., editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, New York. ACM.

Robertson, S. (2004).

Understanding inverse document frequency: On theoretical arguments for idf.
Journal of Documentation, 60:503–520.

Robertson, S. (2005).

On event spaces and probabilistic models in information retrieval.

Information Retrieval Journal, 8(2):319–329.

Robertson, S. and Sparck Jones, K. (1976).

Relevance weighting of search terms.

Journal of the American Society for Information Science, 27:129–146.

Robertson, S. E. and Walker, S. (1994).

Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval.

In [Croft and van Rijsbergen, 1994], pages 232–241.

Robertson, S. E., Walker, S., and Hancock-Beaulieu, M. (1995).

Large test collection experiments on an operational interactive system: Okapi at TREC.

Information Processing and Management, 31:345–360.

Rocchio, J. (1966).

Document retrieval systems - optimization and evaluation.

Report ISR-10 to the NSF, Computation Laboratory, Harvard University.

Rocchio, J. (1971).

Relevance feedback in information retrieval.

In [Salton, 1971].

Roelleke, T., Tsikrika, T., and Kazai, G. (2006).

A general matrix framework for modelling information retrieval.

Journal on Information Processing & Management (IP&M), Special Issue on Theory in Information Retrieval, 42(1).

Roelleke, T. and Wang, J. (2006).

A parallel derivation of probabilistic information retrieval models.

In *ACM SIGIR*, pages 107–114, Seattle, USA.

Roelleke, T. and Wang, J. (2008).

TF-IDF uncovered: A study of theories and probabilities.

In *ACM SIGIR*, pages 435–442, Singapore.

Roelleke, T., Wu, H., Wang, J., and Azzam, H. (2008).

Modelling retrieval models in a probabilistic relational algebra with a new operator: The relational Bayes.
VLDB Journal, 17(1):5–37.

Salton, G., editor (1971).

The SMART Retrieval System - Experiments in Automatic Document Processing.
Prentice Hall, Englewood, Cliffs, New Jersey.

Salton, G., Fox, E., and Wu, H. (1983).

Extended Boolean information retrieval.
Communications of the ACM, 26:1022–1036.

Salton, G., Wong, A., and Yang, C. (1975).

A vector space model for automatic indexing.
Communications of the ACM, 18:613–620.

Sebastiani, F. (2002).

Machine learning in automated text categorization.
ACM Comput. Surv., 34(1):1–47.

Turtle, H. and Croft, W. (1991a).

Efficient probabilistic inference for text retrieval.
In *Proceedings RIAO 91*, pages 644–661, Paris, France.

Turtle, H. and Croft, W. (1991b).

Evaluation of an inference network-based retrieval model.
ACM Transactions on Information Systems, 9(3):187–222.

Turtle, H. and Croft, W. (1992).

A comparison of text retrieval models.
The Computer Journal, 35.

Turtle, H. and Croft, W. B. (1990).

Inference networks for document retrieval.

In Vidick, J.-L., editor, *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, pages 1–24, New York. ACM.

van Rijsbergen, C. J. (1979).

Information Retrieval.
Butterworths, London, 2. edition.
http://www.dcs.glasgow.ac.uk/Keith/Preface.html.

van Rijsbergen, C. J. (1986).

A non-classical logic for information retrieval.
The Computer Journal, 29(6):481–485.

van Rijsbergen, C. J. (1989).

Towards an information logic.
In Belkin, N. and van Rijsbergen, C. J., editors, *Proceedings of the Twelfth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 77–86, New York.

Wong, S. and Yao, Y. (1995).

On modeling information retrieval with probabilistic inference.
ACM Transactions on Information Systems, 13(1):38–68.

Zaragoza, H., Hiemstra, D., and Tipping, M. (2003).

Bayesian extension to the language model for ad hoc information retrieval.
In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval*, pages 4–9, New York, NY, USA. ACM Press.