A tutorial to formal models of information retrieval

European Summer School on Information Retrieval, September 1, 2003

Djoerd Hiemstra

Universiteit Twente

hiemstra@cs.utwente.nl

http://www.cs.utwente.nl/~
hiemstra

Goal

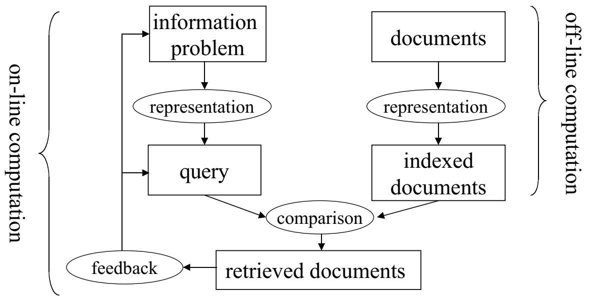
- Gain basic knowledge of IR
  - Intuitive understanding of difficulty of the problem
  - Insight in consequences of modelling assumptions
  - *biased* comparison of formal models

Overview

- PART 1: IR modelling
  - Basic technology
  - An overview of formal models
- PART 2: The Quiz
- PART 3: Language models
  - Retrieval and translation models
  - Advanced models

PART-1: Information Retrieval modelling

Information Retrieval



on-line computation

information problem

representation

query

documents

representation

indexed documents

off-line computation

comparison

retrieved documents

feedback

Full text information retrieval

- Index based on uncontrolled (free) terms (as opposed to controlled terms)
- Every word in a document is a potential index term
- Terms may be linked to specific fragments in a text (title, abstract, preface, image caption, etc.)

Full text information retrieval

- Ranking of documents is essential!
  - *'AltaVista found 32,534,632 documents matching your query'. . .*
  - natural language is ambiguous and vague in contrast with controlled language: (i.e. terms *electrical engineering* vs. UDC 621.3)
- Users are not willing to check out all (millions of) retrieved documents.

Full text information retrieval

- Advantages:
  - fully automatic indexing (saves time and money)
  - less standardisation (tailored to variation in information need of different users)
  - can still be combined (?) with aspects of controlled approach (thesaurus, meta-data)

Full text information retrieval

- Main disadvantage: the (professional) user looses his/her control over the system...
  - because of 'ranking' instead of 'exact matching', the user cannot decrease the size of the retrieved set by entering a more specific query
  - assumptions of stop lists, stemmers, etc. do not hold universally:

    e.g. the query “last will”: are “last” or “will” stop words? should it retrieve “last would”?

Full text information retrieval

- Automatic processing of natural language:
  - statistics (counting words)
  - stop list
  - morphological stemming
  - part-of-speech tagging
  - compound splitting
  - partial parsing: noun phrase extraction
  - other: use of thesaurus, named entity recognition, ...

Full text information retrieval

- stop list
  - remove bad predictors of content
  - e.g. closed word classes (determiners, adverbs, prepositions)
  - not necessarily frequently occurring words
  - example (domain independent):

    about, above, according, accordingly, across, actually, after, again, all, allow, almost, along, already, also, . . .
  - example (domain dependent):

    browse, browser, home, hyper, link, page, web, . . .

Full text information retrieval

- morphological analysis
  - morphology: the way words are build
- rewrite rules (Porter stemmer: inflection and derivation):
  - pakken, pakt, pakte, gepakt → pak

    paars, paarden → ??
- dictionaries (usually only inflection)
  - paarden → paren Verb+3p+past+plural | paard Noun+plural

Full text information retrieval

- compound words
  - word contains more than one morpheme:

    voetbalstadion → voetbal/stadion
    → voet/bal/stadion
    → voet/bal/stad/ion
- fragments or phrases
  - separate words not always good predictors of content
  - e.g. “New York”, “hollandse nieuwe”

Full text information retrieval

access baghdad britain cautiou china council docum dossier drawn franc full hand iraq massiv meet member mix nation page perman present programm reaction remain respons russia secur state sundai uk unit weapon welcom

Iraq dossier meets mixed response

The massive dossier on Iraq's weapons programmes presented to the United Nations has drawn mixed reactions from permanent members of the Security Council.

Russia and China welcomed Baghdad's 12,000-page document - which was handed over on Sunday - while Britain and the United States are remaining cautious.

The Security Council has given access to the full dossier to its five permanent members - China, France, Russia, the UK and the US.

Full text information retrieval

bitterli central clear cloudi cloudier coast cold dai east easterli edg flurri forecast frost lead moder northeast part period persist plenti risk shower sleet snow south southern southwestern sunshin todai weather wind wintri

Today's weather forecast

Clear periods leading to a moderate frost in many parts away from the east coast. The northeast will be cloudier, as will the far south, here the risk of a few snow flurries. The bitterly cold easterly wind persisting.

Plenty of sunshine around, but rather cloudy in northeast, here some wintry showers. The south also rather cloudy, perhaps sleet or snow edging into southwestern and central southern parts later in day.

Models of information retrieval

- A model:
  - abstracts away from the real world
  - uses a branch of mathematics
  - possibly: uses a metaphor for searching

Short history of IR modelling

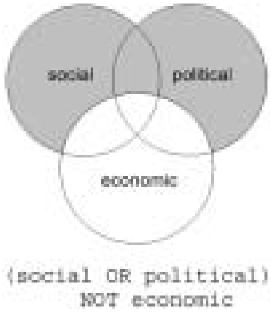
- Boolean model (±1950)
- Document similarity (±1957)
- Probabilistic indexing (±1960)
- Vector space model (±1970)
- Probabilistic retrieval (±1976)
- Fuzzy set models (±1980)
- Inference networks (±1992)
- Language models (±1998)

The Boolean model (±1950)

- Exact matching: data retrieval (instead of *information* retrieval)
  - A term specifies a set of documents
  - Boolean logic to combine terms / document sets
  - AND, OR and NOT

The Boolean model (±1950)

- Venn diagrams



(social OR political)
NOT economic

Statistical similarity between documents (±1957)

- The principle of <u>similarity</u>

  "The more two representations agree in given elements and their distribution, the higher would be the probability of their representing similar information"

  (Luhn 1957)

Statistical similarity between documents (±1957)

- Vector product
  - If the vector has binary components, then the product measures the number of shared terms
  - Vector components might be "weights"

$$score(q,d) = \sum_{k \in \text{matching terms}} q_k \cdot d_k$$

Intermezzo: Term weights??

- *tf.idf* term weighting schemes
  - a family of hundreds (thousands) of algorithms to assign weights that reflect the importance of a term in a document
  - *tf* = term frequency: the number of times a term occurs in a document
  - *idf* = inverse document frequency: usually the logarithm of $^1/_{df}$, where *df* = document frequency: the number of documents that contains the term

Probability ranking (±1960)

- The <u>probability ranking</u> principle

  "If a reference retrieval system's response to each request is a ranking of the documents in the collections in order of decreasing probability of usefulness to the user (...) then the overall effectiveness will be the best that is obtainable on the basis of the data.

  (Robertson 1977)

Probabilistic indexing (±1960)

- *an indexer, which runs through the various possible index terms q that possibly apply to a document, might assign a probability $P(q|D)$ to a term given a document instead of making a yes/no decision.*

  (Maron & Kuhns 1960)

Probabilistic indexing (±1960)

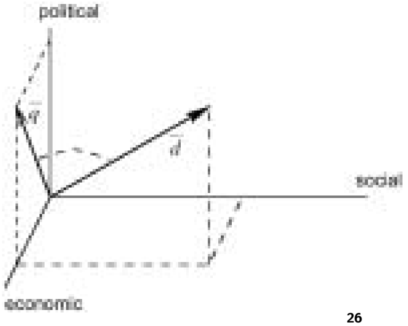
- Structured queries use probabilities as follows (ranking also takes document priors $P(D)$ in account),

$$P(T_1 \text{ AND } T_2|D) = P(T_1|D) \cdot P(T_2|D)$$
$$P(T_1 \text{ OR } T_2|D) = P(T_1|D) + P(T_2|D) - P(T_1|D)P(T_2|D)$$
$$P(\text{NOT } T|D) = 1 - P(T|D)$$

- PRO: Mathematically sound: probability theory; models structured queries
- CON: Assumes manual indexing

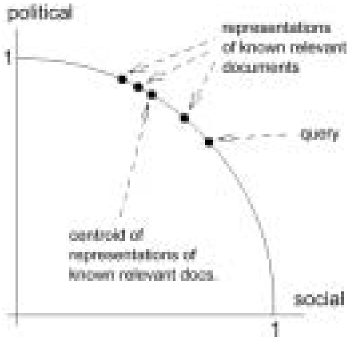
Vector space model (±1970)

- Documents and queries are vectors in a high-dimensional space
- Geometric measures (distances, angles)



Vector space model (±1970)

- Measuring the angle is like normalising the vectors to length 1.
- Relevance feedback: move query on the sphere at length 1.



(Rocchio 1971)

Vector space model (±1970)

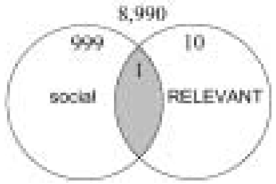
- PRO: Nice metaphor, easily explained;
  Mathematically sound: geometry;
  Great for relevance feedback
- CON: Need term weighting (*tf.idf*);
  Hard to model structured queries

(Salton & McGill 1983)

Probabilistic retrieval (±1976)

- Probability of getting (retrieving) a relevant document from the set of documents indexed by "social".

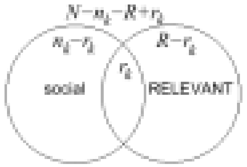
(Robertson & Sparck-Jones 1976)



r = 1 (number of relevant docs containing "social")
R = 11 (number of relevant docs)
n = 1000 (number of docs containing "social")
N = 10000 (total number of

Probabilistic retrieval (±1976)

- Bayes' rule $$P(L \mid D) = \frac{P(D \mid L)P(L)}{P(D)}$$
- Conditional independence $$P(D \mid L) = \prod_k P(D_k \mid L)$$



$$P(D_k{=}1 \mid L{=}1) = r_k/_R$$
$$P(D_k{=}1 \mid L{=}0) = n_k-r_k/_{N-R}$$
$$P(D_k{=}0 \mid L{=}1) = R-r_k/_R$$
$$P(D_k{=}0 \mid L{=}0) = N-n_k-R+r_k/_{N-R}$$

Probabilistic retrieval (±1976)

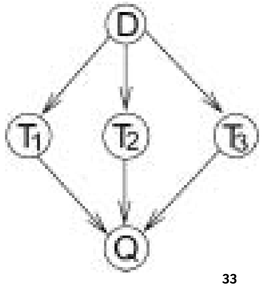
- PRO: does not need term weighting
- CON: within document statistics (*tf's*) do not play a role
  Need results from relevance feedback

Fuzzy set models (±1980)

- Degree of set membership T(a) to represent inexactness and vagueness
  - T(a AND b) = min(T(a), T(b))
  - T(a OR b) = max(T(a), T(b))
  - T(NOT b) = 1 – T(b)
- PRO: structured queries!
- CON: need term weights to define T(a)

Inference networks (±1992)

- Graphical model of conditional dependencies
  - $P(D,T_1,T_2,T_3,Q) =$
    $P(D) \cdot P(T_1 | D)$
    $\cdot P(T_2 | D) \cdot P(T_3 | D)$
    $\cdot P(Q | T_1,T_2,T_3)$



Inference networks (±1992)

- Problem: the specification of $P(Q|T_1,T_2,T_3)$ needs 2^{n+1} probabilities(!)
- Canonical forms, computed in linear time:
  - $P_{\text{AND}}(Q|T_1,T_2,T_3) = P(T_1) \cdot P(T_2) \cdot P(T_3)$
  - $P_{\text{OR}}(Q|T_1,T_2,T_3) = 1 - (1 - P(T_1)) \cdot (1 - P(T_2)) \cdot (1 - P(T_3))$
  - $P_{\text{NOT}}(Q|T) = 1 - P(T)$
  - $P_{\text{SUM}}(Q|T_1,T_2,T_3) = (P(T_1) + P(T_2) + P(T_3)) / 3$
  - $P_{\text{WSUM}}(Q|T_1,T_2,T_3) = \text{w}_1 P(T_1) + \text{w}_2 P(T_2) + \text{w}_3 P(T_3)$

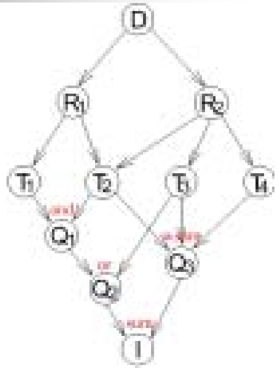
NB AND, OR and NOT as in probabilistic

Inference networks (±1992)

PRO: combine evidence in a complex way

CON: – need term weighting scheme to specify $P(T_i)$
– Learning is still intractible with canonical forms

NB when using the network, ignore "document part"



Language models (±1998)

- Let's 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 "ESSIR", "summer" and "school"?
- Compute the probability, and use it to rank the documents.

NB this is like a trivial Maron & Kuhns – like "probabilistic indexer"

Language models (±1998)

- Given a query $T_1, T_2, \ldots, T_n$, rank the documents according to the following probability measure:

$$P(T_1, T_2, \ldots, T_n \mid D) = \prod_{i=1} ((1-\lambda_i)P(T_i) + \lambda_i P(T_i \mid D))$$

- Linear combination of document model and background model

  λ_i : probability of document model

  $1-\lambda_i$: probability of background model

  $P(T_i \mid D)$: document model

  $P(T_i)$: background model

Language models (±1998)

- Probability theory / hidden Markov model theory
- Successfully applied to speech recognition, and:
  - optical character recognition, part-of-speech tagging, stochastic grammars, spelling correction, machine translation, etc.

(Ponte & Croft 1998; Hiemstra 1998)

References

- Hiemstra, D. (1998). A linguistically motivated probabilistic model of information retrieval. In Proceedings of the 2nd European Conference on Research and Advanced Technology for Digital Libraries (ECDL), 569-584.
- Lee, J.H. (1995). Analyzing the effectiveness of extended Boolean models in information retrieval. Technical Report TR95-1501, Cornell University. http://cs-tr.cs.cornell.edu/
- Luhn, H.P. (1957). A statistical approach to mechanised encoding and searching of literary information. IBM Journal of Research and Development 1(4), 309-317.
- Maron, M.E. and J.L. Kuhns (1960). On relevance, probabilistic indexing and information retrieval. Journal of the Association for Computing Machinery 7, 216-244.
- Ponte, J.M. and W.B. Croft (1998). A language modeling approach to information retrieval. In Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval (SIGIR'98).
- Robertson, S.E. (1977). The probability ranking principle in IR

References

- Robertson, S.E. and K. Sparck-Jones (1976). Relevance weighting of search terms. Journal of the American Society for Information Science 27, 129-146.
- Rocchio, J.J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), The Smart Retrieval System: Experiments in Automatic Document Processing, pp. 313-323. Prentice Hall.
- Salton, G. and M.J. McGill (1983). Introduction to Modern Information Retrieval. McGraw-Hill.
- Savino, P. and F. Sebastiani (1998). Essential bibliography on multimedia information retrieval, categorisation and filtering. Technical Report CNR Pisa.
- Shannon, C.E. (1948). A mathematical theory of communication. Bell System Technical Journal 27, 379-423, 623-656.
- Turtle, H. and W.B. Croft (1991). Evaluation of an inference network-based retrieval model. ACM Transactions on Information Systems 9(3), 187-222.

PART-2: The formal IR model Quiz

Question 1

- In the Boolean model: how many different sets of documents can be specified with 3 query terms?
  - 8
  - 9
  - 256
  - unlimited

Question 2

- In the vector space model: Given 2 documents D1 and D2. Suppose the similarity between D1 and D2 is 0.08, what will be the similarity between D2 and D1? (i.e. if we interchange the documents)
  - smaller than 0.08
  - equal: 0.08
  - bigger than 0.08
  - it depends document's contents

Question 3

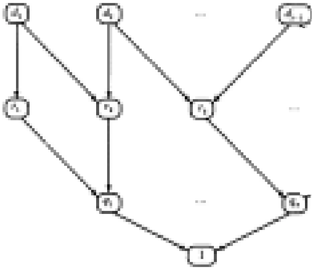
- In the probabilistic model: suppose we query for `essir`, and D1 has more occurrences of `essir` than D2, which document will be ranked first?
  - D1 will be ranked before D2
  - D2 will be ranked before D1
  - it depends on the model's implementation

Question 4

- In the fuzzy set model: suppose we query for `(essir OR NOT(essir)) AND NOT(impotence))`, which documents will be ranked first?
  - documents satisfying `essir`
  - documents satisfying `NOT(essir)`
  - documents satisfying `NOT(impotence)`
  - it depends on the member functions

Question 5

- Which arrows in the inference net is properly computed as "conditional de



- all
- only the upper layer (document network)
- only the lower layers (query network)

Question 6

- In the language model: let's assume document D consisting of 100 words in total, contains 4 times the word "ESSIR", what is $P(T=\text{"ESSIR"}|D)$? (ignoring the background model)
  - smaller than 4/100 = 0.04
  - equal to 4/100 = 0.04
  - bigger than 4/100 = 0.04
  - it depends of the *tf.idf* weights

PART-3: Statistical language models

Statistical language models

- Noisy channel paradigm (Shannon 1948)

I (input)

noisy channel

O (output)

- hypothesise all possible input texts I and take the one with the highest probability, symbolically:

$$\hat{I} = \underset{I}{\operatorname{argmax}}\, P(I\,|\,O)$$
$$= \underset{I}{\operatorname{argmax}}\, P(I) \cdot P(O\,|\,I)$$

Statistical language models

- Noisy channel paradigm (Shannon 1948)

D (document)

noisy channel

$T_1, T_2, \ldots$ (query)

- hypothesise all possible documents D and take the one with the highest probability, symbolically:

$$\hat{D} = \underset{D}{\operatorname{argmax}}\, P(D\,|\,T_1, T_2, \cdots)$$
$$= \underset{D}{\operatorname{argmax}}\, P(D) \cdot P(T_1, T_2, \cdots\,|\,D)$$

Statistical language models

- Given a query $T_1, T_2, \ldots, T_n$, rank the documents according to the following probability measure:

$$P(T_1, T_2, \ldots, T_n\,|\,D) = \prod_{i=1} ((1-\lambda_i)P(T_i) + \lambda_i P(T_i\,|\,D))$$

λ_i : probability that the term on position i is important

$1-\lambda_i$: probability that the term is unimportant

$P(T_i\,|\,D)$: probability of an important term

$P(T_i)$: probability of an unimportant term

Statistical language models

- Definition of probability measures:

$$P(T_i = t_i\,|\,D = d) = \frac{tf(t_i, d)}{\sum_t tf(t,d)} \quad \text{(important term)}$$

$$P(T_i = t_i) = \frac{df(t_i)}{\sum_t df(t)} \quad \text{(unimportant term)}$$

Statistical language models

- How to estimate value of λ_i ?
  - For ad-hoc retrieval (i.e. no previously retrieved documents to guide the search)

    $\lambda_i = constant$ (i.e. each term equally important)
  - Note that for extreme values:

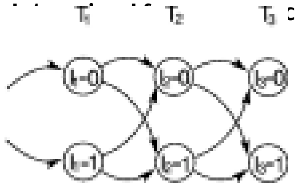
    $\lambda_i = 0$: term does not influence ranking

    $\lambda_i = 1$: term is mandatory in retrieved docs.

    $\lim \lambda_i \to 1$: docs containing n query terms are ranked above docs containing $n-1$

Statistical language models

- Presentation as hidden Markov model
  - finite state machine: probabilities governing transitions
  - sequence of state transitions cannot be [illegible]ce of output symbols



Statistical language models

- Re-estimate the value of λ_i from relevant documents (relevance feedback)
  - Expectation Maximisation algorithm
  - Estimate different value of λ_i for each term (i.e. different importance of each term.)

Statistical language models

- Implementation

$$P(T_1,T_2,\cdots,T_n\,|\,D)=\prod_{i=1}^{n}((1-\lambda_i)P(T_i)+\lambda_i P(T_i\,|\,D))$$

$$\vdots$$

$$P(T_1,T_2,\cdots,T_n\,|\,D)\propto\sum_{i=1}^{n}\log(1+\frac{\lambda_i P(T_i\,|\,D)}{(1-\lambda_i)P(T_i)})$$

Statistical language models

- Implementation as vector product:

$$score(q,d) \quad = \sum_{k\in\text{ matching terms}} q_k\cdot d_k$$

$$q_k = tf(k,q)$$

$$d_k = \log(1+\frac{tf(k,d)\sum_t df(t)}{df(k)\sum_t tf(t,d)}\cdot\frac{\lambda_k}{1-\lambda_k})$$

Statistical language models

- Implementation as vector product:

$$score(q,d) \quad = \sum_{k\in\text{ matching terms}} q_k\cdot d_k$$

$$q_k = tf(k,q)$$

$$d_k = \log(1+\frac{\bigcirc{tf(k,d)}\bigcirc{\sum_t df(t)}}{\bigcirc{df(k)}\bigcirc{\sum_t tf(t,d)}}\cdot\boxed{\frac{\lambda_k}{1-\lambda_k}})$$

Language models & translation

- Cross-language information retrieval (CLIR):
  - Enter query in one language (language of choice) and retrieve documents in one or more other languages.
  - The system takes care of automatic translation

Cross-language IR

<u>cross-language information retrieval</u>

–

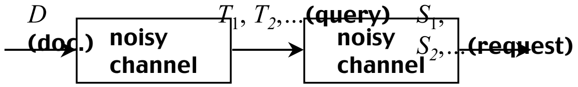
<u>zoeken in anderstalige informatie</u>

–

<u>recherche d'informations multilingues</u>

Language models & translation

- Noisy channel paradigm


D (doc.) → noisy channel → $T_1, T_2,\ldots$ (query) → noisy channel → $S_1, S_2,\ldots$ (request)

- hypothesise all possible documents D and take the one with the highest probability:

$$\hat{D} = \operatorname*{argmax}_D P(D\,|\,S_1,S_2,\cdots)$$

$$= \operatorname*{argmax}_D P(D)\cdot\sum_{T_1,T_2,\cdots}P(T_1,T_2,\cdots;S_1,S_2,\cdots\,|\,D)$$

Language models & translation

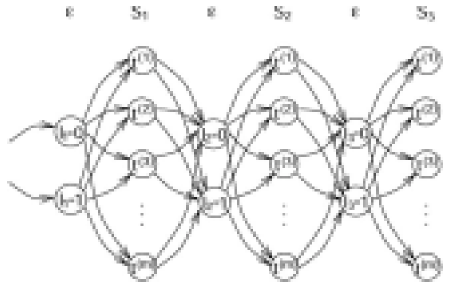
- Cross-language information retrieval :
  - Assume that the translation of a word/term does not depend on the document in which it occurs.
  - if: $S_1, S_2,\ldots, S_n$ is a Dutch query of length n
  - and $t_{i1}, t_{i2},\ldots, t_{im}$ are m English translations of the Dutch query term S_i

$$P(S_1,S_2,\ldots,S_n\,|\,D) =$$

$$\prod_{i=1}^{n}\sum_{j=1}^{m_i}P(S_i\,|\,T_i=t_{ij})((1-\lambda_i)P(T_i=t_{ij})+\lambda_i P(T_i=t_{ij}\,|\,D))$$

Language models & translation

- Presentation as hidden Markov model



Language models & translation

- How does it work in practice?
  - Find for each Dutch query term N_i the possible translations $t_{i1}, t_{i2},\ldots, t_{im}$ and translation probabilities
  - Combine them in a structured query
  - Process structured query

Language models & translation

- Example:
  - Dutch query: *gevaarlijke stoffen*
  - Translations of *gevaarlijke* : *dangerous* (0.8) or *hazardous* (0.2)
  - Translations of *stoffen* : *fabric* (0.3) or *chemicals* (0.3) or *dust* (0.4)
  - Structured query:

    $((0.8\ dangerous \cup 0.2\ hazardous)\ ,$

    $(0.3\ fabric \cup 0.3\ chemicals \cup 0.4\ dust))$

Language models & translation

- Other applications using the translation model
  - On-line stemming
  - Synonym expansion
  - Spelling correction
  - 'fuzzy' matching
  - Extended (ranked) Boolean retrieval

Language models & translation

- Note that:
  - $\lambda_i = 1, \text{ for all } 0 \le i \le n$: Boolean retrieval
  - Stemming and on-line morphological generation give exact same results:

    $P(\text{funny} \cup \text{funnies, table} \cup \text{tables} \cup \text{tabled}) =$

    $P(\texttt{funni, tabl})$

Advanced applications

- High precision searches (literal strings)
- Highly structured documents (XML)
- Priors

Advanced applications (1)

- Extension for adjacent words (index needs position information)
- Given a query $T_1, T_2,\ldots, T_n$,rank the documents by the following measure:

$$P(T_1, T_2,\ldots,T_n \mid D) =$$

$$\prod_{i=1}^{n}((1-\lambda_i-\mu_i)P(T_i)+\lambda_i P(T_i \mid D)+\mu_i P(T_i \mid T_{i-1}, D))$$

Advanced applications (2)

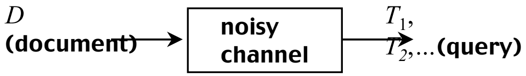
- Extension for record fields, e.g. title (index should support structured documents)
- Given a query $T_1, T_2,\ldots, T_n$,rank the documents by the following measure:

$$P(T_1, T_2,\ldots,T_n \mid D) =$$

$$\prod_{i=1}^{n}((1-\lambda_i-\mu_i)P(T_i)+\lambda_i P(T_i \mid D)+\mu_i P(T_i \mid F = \text{title}, D))$$

Advanced applications (3): about priors

- Noisy channel paradigm (Shannon 1948)


D (document)

noisy channel

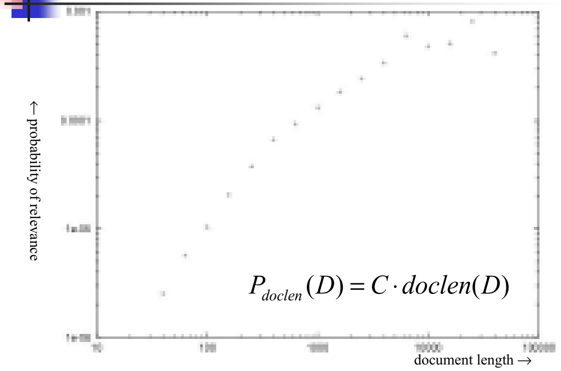
$T_1, T_2, \ldots$ (query)

- hypothesise all possible documents D and take the one with the highest probability, symbolically:

$$\hat{D} = \underset{D}{\operatorname{argmax}}\, P(D \mid T_1, T_2, \cdots)$$

$$= \underset{D}{\operatorname{argmax}}\, \boxed{P(D)} \cdot P(T_1, T_2, \cdots \mid D)$$

Prior probability of relevance on ad-hoc search task

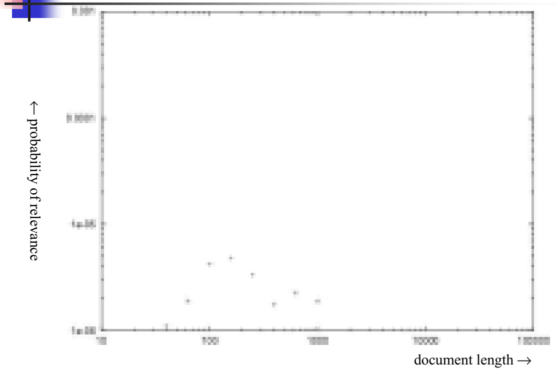


$$P_{doclen}(D) = C \cdot doclen(D)$$

Priors in Entry Page Search

- Sources of Information
  - Document length
  - Number of links pointing to a document
  - The depth of the URL
  - Occurrence of cue words (‘welcome’,’home’)
  - number of links in a document
  - page traffic

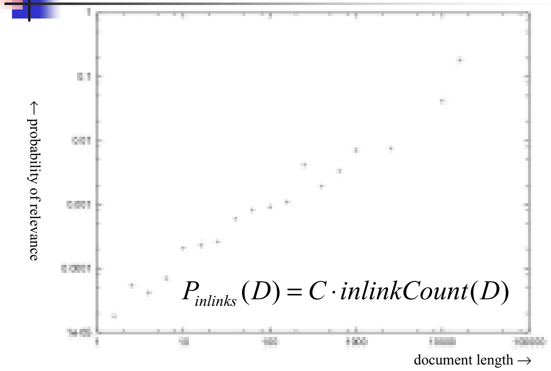
Priors in Entry Page Search



Priors in Entry Page Search

- Assumption
  - Entry pages referenced more often
- Different types of inlinks
  - From other hosts (recommendation)
  - From same host (navigational)
- Both types point often to entry pages

Priors in Entry Page Search



$$P_{inlinks}(D) = C \cdot inlinkCount(D)$$

Priors in Entry Page Search URL depth

- Top level documents are often entry pages
- Four types of URLs
  - root: `www-clips.imag.fr`
  - subroot: `www-clips.imag.fr/mrim/`
  - path: `www-clips.imag.fr/mrim/essir03/`
  - file: `www-clips.imag.fr/mrim/essir03/main.html`

Priors in Entry Page Search results

method	Content	Anchors
$P(Q\|D)$	0.3375	0.4188
$P(Q\|D)P_{doclen}(D)$	0.2634	0.5600
$P(Q\|D)P_{inlink}(D)$	0.4974	0.5365
$P(Q\|D)P_{URL}(D)$	0.7705	0.6301

Language models conclusion

- Simple model: like *tf.idf* weighting in vector model
- Translation model: accounts for multiple query representations (e.g. CLIR or stemming)
- Advanced models: account for multiple document representations and or position information
- Document priors: account for "non-content" information
- Only PRO's, no CON's ☺