Chap. 4: Probabilistic IR

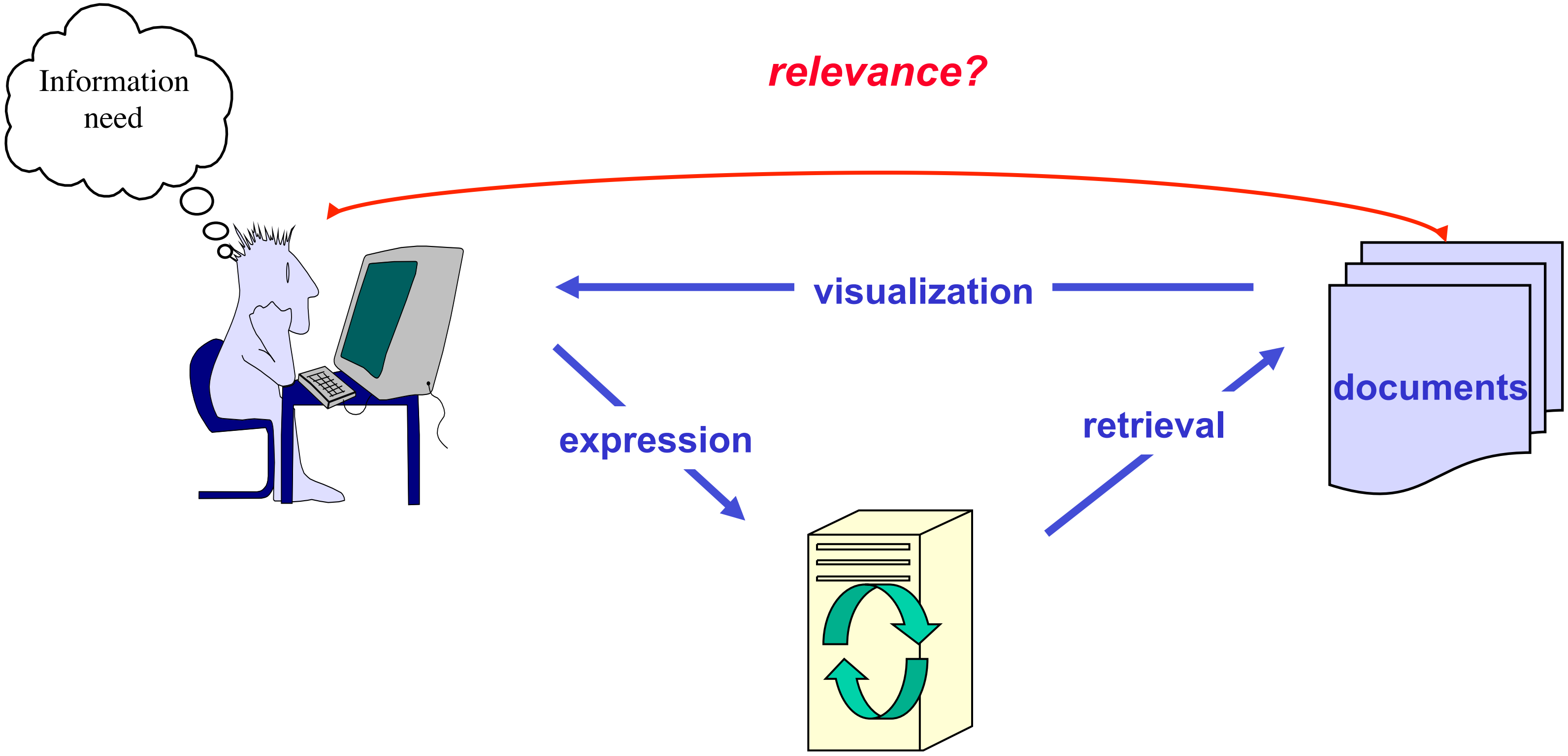
Jean-Pierre Chevallet
Philippe Mulhem
(with some data from Eric Gaussier)
Team MRIM-LIG

Outline

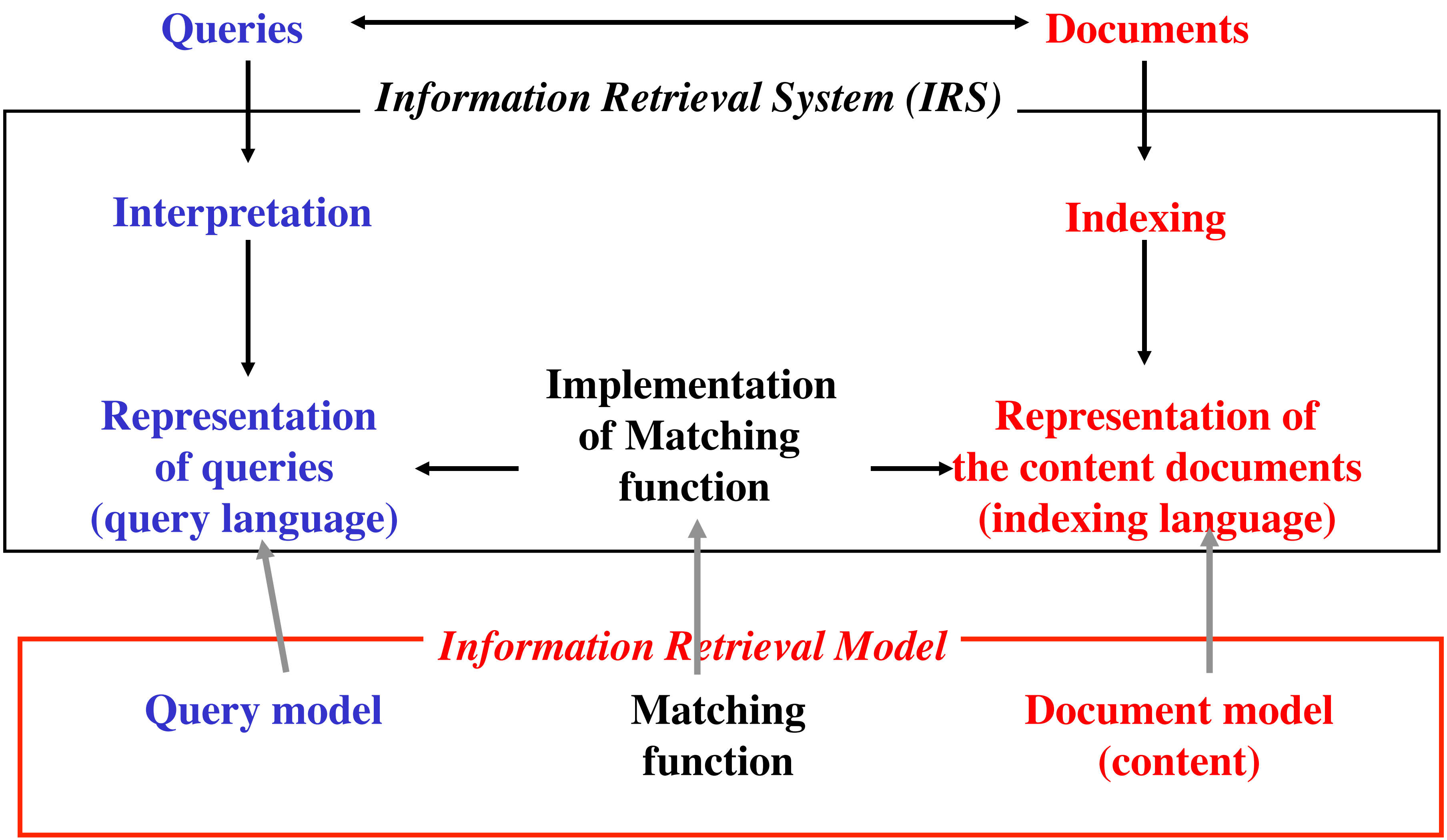
1. Introduction
2. Binary Independent Model
3. Inference Networks
4. Language Models
5. Conclusion

1. Introduction

- Challenge of Information Retrieval:
  - Content base access to documents that satisfy an user's information



1. Introduction



1. Introduction

- Probabilistic Models : capture the IR problem in a probabilistic framework
  - first probabilistic model (Binary Independent Retrieval Model) by Robertson and Spark-Jones in 1976…
  - late 90s, emergence of language models, still hot topic in IR
  - Overall question: "what is the probability for a document to be relevant to a query?"
    - several interpretation of this sentence

1. Introduction

- Classical Models
  - Boolean
    - Strict
    - Weighted
  - Vector Space
  - Probabilistic
    - Classical [Robertson & Spärk-Jones]
    - Inference Network [Tutle & Croft]
    - Language Models [Croft], [Hiemstra][Nie]

1. Introduction

- Points covered by the lessons
  - main probabilistic information retrieval models
    - theoretical aspects
    - examples

1. Introduction

- Probabilistic Model of IR
  - Different approaches of seeing a probabilistic approach for information retrieval
    - Classical approach: probability to have the event *Relevant* knowing one document and one query.
    - Inference Networks approach: probability that the query is true after inference from the content of a document.
    - Language Models approach: probability that a query is generated from a document.

2. Binary Independant Retrieval Model

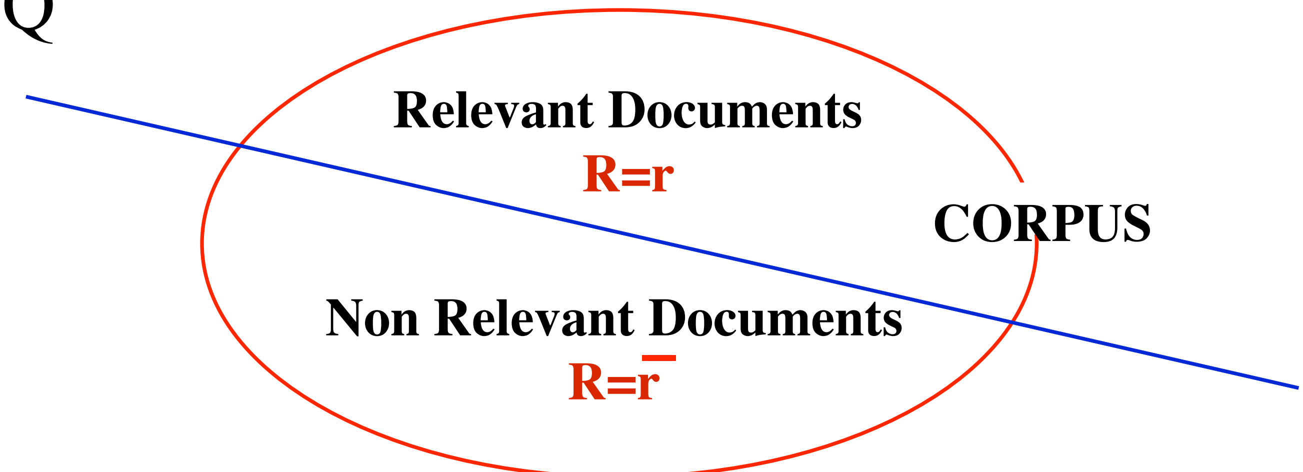
- [Robertson & Spark-Jones 1976]
  - computes the relevance of a document from the relevance known a priori from other documents.
  - achieved by estimating of each indexing term a the document, and by using the Bayes Theorem and a decision rule.

2. BIR

- R: binary random variable
  - R = r : relevant; R = $\bar{r}$: non relevant
  - P(R=r | d, q): probability that R is r for the document and the query considered (noted P(r | d, q))
- probability of relevance depends only from document and query
- term weights are binary (d=(11…100…), w^d_t=0 or 1)
- Each term t is characterized by a a binary variable w_t, indicating the probability that the term occurs. $P(w_t = 1 \mid q, r)$: probability that t occurs in a relevant document. ($P(w_t = 0 \mid q, r) = 1 - P(w_t = 1 \mid q, r)$)
- the terms are conditionnaly independant to R

2. BIR

– For a query Q



with

Corpus = rel $\cup$ nonrel

Relevant Documents $\cap$ Non Relevant Documents = $\emptyset$

$\boxed{P(R=r \mid d, q)}$

Probability for the document d to be a member of the relevant set of documents for q

2. BIR

- Matching function :
  - Use of Bayes theorem

Probability to obtain the description d from observed relevances

P(r,q) is the relevance probability. It is the chance of randomly taking one document from the corpus which is relevant for the query q

$$P(r|d,q) = \frac{P(d|r,q).P(r,q)}{P(d,q)}$$

Probability that the document d belongs to the set of relevant documents of the query q.

probability that the document d is picked for q

2. BIR

Matching function

– Decision: document retrieved if

$$\frac{P(r|d,q)}{P(\bar{r}|d,q)} = \frac{P(d|r,q).P(r,q)}{P(d|\bar{r},q).P(\bar{r},q)} > 1$$

- IR looks for a ranking, so we eliminate $P(r,q)/P(\bar{r},q)$ for a given query (constant)
- In IR, it is more simple to use logs:

$$rsv(d) =_{rank} \log\left(\frac{P(d|r,q)}{P(d|\bar{r},q)}\right)$$

2. BIR

- Matching function
  - Hypothesis of independence between terms (Binary Independance) with weight $w^d{}_t$ for term t in d :

$$P(d|r,q) = P(d=(10...110...)|r,q) = \prod_{w_t^d=1} P(w_t^d=1|r,q).\prod_{w_t^d=0} P(w_t^d=0|r,q)$$

$$P(d|\bar{r},q) = P(d=(10...110...)|\bar{r},q) = \prod_{w_t^d=1} P(w_t^d=1|\bar{r},q).\prod_{w_t^d=0} P(w_t^d=0|\bar{r},q)$$

2. BIR

- Notations $p_t = P(w_t = 1 \mid r, q)$ $\qquad q_t = P(w_t = 1 \mid \bar{r}, q)$

- Then $P(w_t = 0 \mid r, q) = 1 - p_t$ $\qquad P(w_t = 0 \mid \bar{r}, q) = 1 - q_t$

- So

$$rsv(d) =_{rank} \log\left(\frac{P(d \mid r, q)}{P(d \mid \bar{r}, q)}\right) = \log\left(\frac{\prod_{w_t^d=1} p_t \cdot \prod_{w_t^d=0} 1 - p_t}{\prod_{w_t^d=1} q_t \cdot \prod_{w_t^d=0} 1 - q_t}\right) = \log\left(\prod_{w_t^d=1} \frac{p_t}{q_t} \times \prod_{w_t^d=0} \frac{1 - p_t}{1 - q_t}\right)$$

$$rsv(d \mid r, q) =_{rank} \log\left(\prod_{w_t^d=1} \frac{p_t}{q_t}\right) + \log\left(\prod_{w_t^d=0} \frac{1 - p_t}{1 - q_t}\right)$$

2. BIR

- Hypothesis: $p_t = q_t$ for the terms t in the document and absent in the query, because no impact on the relevance of D for Q

$$rsv(d|r,q) =_{rank} \log\left(\prod_{t \in D \cap Q} \frac{p_t}{q_t}\right) + \log\left(\prod_{t \in Q \setminus D} \frac{1-p_t}{1-q_t}\right)$$

2. BIR

$$rsv(d|r,q) =_{rank} \log(\prod_{t \in D \cap Q} \frac{p_t}{q_t}) + \log(\prod_{t \in Q \setminus D} \frac{1-p_t}{1-q_t})$$

$$=_{rank} \log(\prod_{t \in D \cap Q} \frac{p_t}{q_t}) \underbrace{- \log(\prod_{t \in D \cap Q} \frac{1-p_t}{1-q_t})} + \log(\prod_{t \in Q \setminus D} \frac{1-p_t}{1-q_t}) \underbrace{+ \log(\prod_{t \in D \cap Q} \frac{1-p_t}{1-q_t})}$$

$$=_{rank} \log(\prod_{t \in D \cap Q} \frac{p_t}{q_t}) + \log(\prod_{t \in D \cap Q} \frac{1-q_t}{1-p_t}) + \log(\prod_{t \in Q \setminus D} \frac{1-p_t}{1-q_t}) + \log(\prod_{t \in D \cap Q} \frac{1-p_t}{1-q_t})$$

$$= \log(\prod_{t \in D \cap Q} \frac{p_t(1-q_t)}{q_t(1-p_t)}) - \log(\prod_{t \in Q} \frac{1-p_t}{1-q_t})$$

$$= \log(\prod_{t \in D \cap Q} \frac{p_t(1-q_t)}{q_t(1-p_t)})$$

because $\log(\prod_{t_i \in Q} \frac{1-p_i}{1-q_i})$ is a constant for a given query Q.

2. BIR

- Finaly .... !

$$rsv(d|r,q) =_{rank} \sum_{t \in D \cap Q} \log(\frac{p_t(1-q_t)}{q_t(1-p_t)}) = \sum_{t \in D \cap Q} \log(\frac{p_t}{(1-p_t)} . \frac{(1-q_t)}{q_t}) = \sum_{t \in D \cap Q} \log\left(\frac{\frac{p_t}{1-p_t}}{\frac{q_t}{1-q_t}}\right)$$

- Now : how to estimate p_t and q_t ?

2. BIR

- Use a set of resolved queries
  - (queries for which we know the answers on the corpus)

	Relevant	Non Relevant	Total
term t present	r_t	$n_t - r_t$	n_t
term t absent	$R_t - r_t$	$N - n_t - (R_t - r_t)$	$N - n_t$
Total	R_t	$N - R_t$	N

  - With
    - r_t: number of relevant documents containing the term t
    - R_t: number of relevant documents for a query that contains the term t
    - N: number of documents in the corpus
    - $n_t - r_t$: number of non relevant documents containing the term t

2. BIR

- Estimation of p_i and q_i on a set of resolved queries

	Relevant	Non Relevant	Total
term t present	r_t	$n_t - r_t$	n_t
term t absent	$R_t - r_t$	$N - n_t - (R_t - r_t)$	$N - n_t$
Total	R_t	$N - R_t$	N

$$p_t = \frac{r_t}{R_t} \qquad 1 - p_t = \frac{R_t - r_t}{R_t}$$

$$q_t = \frac{n_t - r_t}{N - R_t} \qquad 1 - q_t = \frac{N - R_t - n_t + r_t}{N - R_t}$$

2. BIR

- Global formula

$$rsv(D) =_{rank} \sum_{t \in D \cap Q} \log \left(\frac{\frac{r_t / R_t}{(R_t - r_t)/R_t}}{\frac{(n_t - r_t)/(N - R_t)}{(N - R_t - n_t + r_t)/(N - R_t)}} \right) = \boxed{\sum_{t \in D \cap Q} \log \left(\frac{\frac{r_t}{R_t - r_t}}{\frac{n_t - r_t}{N - R_t - n_t + r_t}} \right)}$$

- to avoid problems with 0s:

$$rsv(D) =_{rank} \sum_{t \in D \cap Q} \log \left(\frac{\frac{r_t + 0.5}{R_t - r_t + 0.5}}{\frac{n_t - r_t + 0.5}{N - R_t - n_t + r_t + 0.5}} \right)$$

2. BIR

- Problem of initial probabilities
  - How to deal with terms not in the resolved queries
- Basic model binary and independent

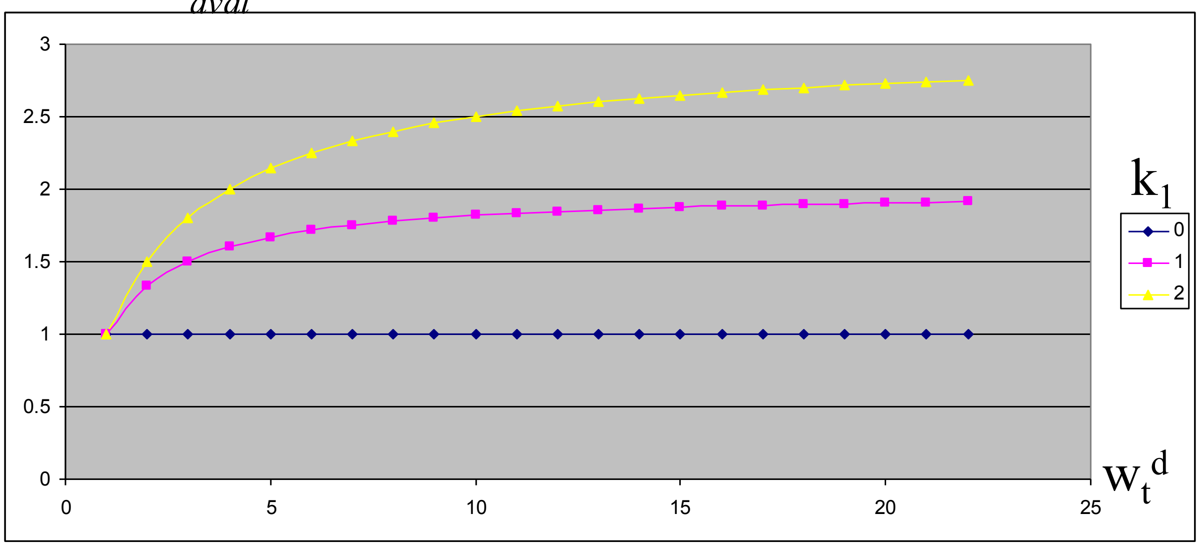
2. BIR

- Extension to weighted terms
  - Best Match [Robertson 1994] : BM25

$$rsv_{BM25}(d|r,q) =_{rank} \sum_{t \in d \cap q} \log\left(\frac{N - n_t + 0.5}{n_t + 0.5}\right) . \frac{(k_1+1)w_t^d}{k_1((1-b) + b.\frac{dl}{avdl}) + w_t^d} . \frac{(k_3+1).w_t^q}{k_3 + w_t^q}$$

$$\frac{(k_1+1)w_t^d}{k_1((1-b) + b.\frac{dl}{avdl}) + w_t^d}$$

dl=avdl, b=1



common values :

k_1 in [1, 2]

b=0.75

k_3 in [0, 1000]

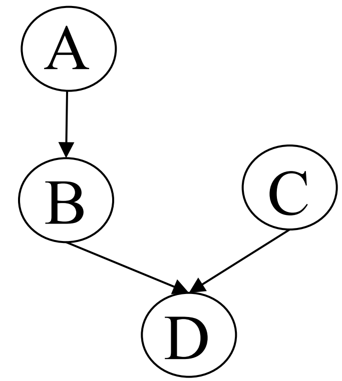
State of the art results

3. Inference Networks IR Models

- [Turtle & Croft 1996]
  - Inspired from Bayesian Belief Networks in Artificial Intelligence
  - Idea: Compute the probability to obtain a query using documents $P(Doc \rightarrow Query)$: combination of evidences
  - Inference Network
    - Nodes: random variables
    - Links: dependencies
    - Direct Acyclic Graph

3. Inference Networks IR Models

Example:



Uncertain inference

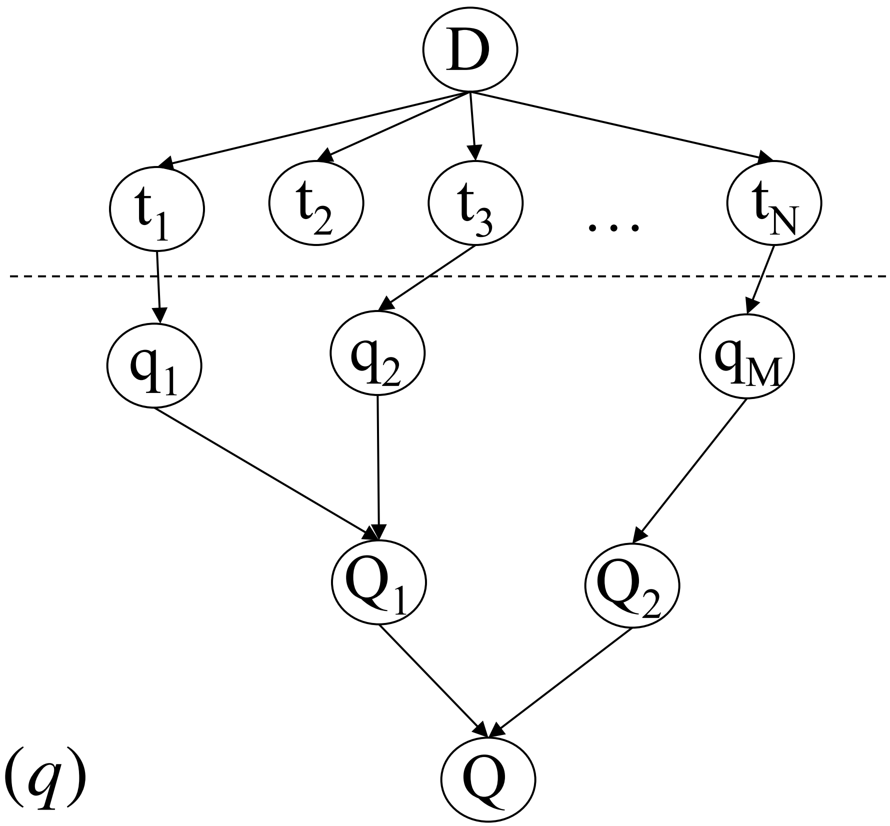
$X = true \equiv x \qquad X = false \equiv \bar{x}$

$$P(d) = P(d/b,c).P(b).P(c) + P(d/\bar{b},c).P(\bar{b}).P(c)$$
$$+ P(d/b,\bar{c}).P(b).P(\bar{c}) + P(d/\bar{b},\bar{c}).P(\bar{b}).P(\bar{c})$$

$P(b) = P(b/a).P(a) + P(b/\bar{a}).p(\bar{a}) \qquad P(\bar{b}) = P(\bar{b}/a).p(a) + P(\bar{b}/\bar{a}).p(\bar{a})$

3. Inference Networks IR Models

- In IR:
  - Binary nodes
  - Example
  - Inference



$$prob(d \rightarrow q) = prob(q)$$

$$= prob(q/q_1,q_2).p(q_1).p(q_2) + prob(q/\overline{q_1},q_2).p(\overline{q_1}).p(q_2)$$

$$+ prob(q/q_1,\overline{q_2}).p(q_1).p(\overline{q_2}) + prob(q/\overline{q_1},\overline{q_2}).p(\overline{q_1}).p(\overline{q_2})$$

3. Inference Networks IR Models

- Use in IR
  - Example:
    - $P(D) = 1/|Corpus|$
    - $P(t_i/D) = tf_{i,D}.idf_i$ if node from D, and $p(t_i)=0$ othewise
    - $P(q_j/t_i)=1$ if link, and $p(q_i) = 0$ othewise
    - Operators for the Q_i with #and, #or, …
    - $P(Q/Q_k)=1$

3. Inference Networks IR Models

- More a framework for IR than a theoretical model.
- Problem of initial probabilities not solved (in fact tf.idf…)

- System: Inquery

4. Language Models of IR

- Consider two dices d1 and d2 so that :
  - for d1 $P(1) = P(3) = P(5) = \frac{1}{3} - \varepsilon \quad P(2) = P(4) = P(6) = \varepsilon$
  - for d2 $P(1) = P(3) = P(5) = \varepsilon \quad P(2) = P(4) = P(6) = \frac{1}{3} - \varepsilon$

- Suppose that we observe the sequence Q={1,3,3,2}.
- What dice is likely to have generated this sequence?

4. Language Models of IR

$$P(Q|d1) = (\frac{1}{3} - \varepsilon)^3 . \varepsilon \qquad P(Q|d2) = (\frac{1}{3} - \varepsilon) . \varepsilon^3$$

$$if \ \varepsilon = 0.01$$

$$P(Q|d1) = 3.38E - 4 \qquad P(Q|d2) = 2.99E - 6$$

4. Language Models of IR

- In IR
  - the documents are the dices, we will represent documents as "documents models"
  - the query is the sequence

4. Language Models of IR

- Comes from speech understanding theory
- Idea : Use of statistical techniques to estimate both document models and the matching score of document for a query
  - Document model ?
    - A document is a « bag of terms »
    - A language model of a document is a probability function of its terms. The terms being part of the indexing vocabulary.

4. Language Models of IR

– Models

- Probability P of occurrence of a word or a word sequence in one language
  - – Consider a sequence s composed of words : $m_1, m_2, \ldots, m_l$.
  - – The probability P(s) may be computed by

$$P(s) = \prod_{i=1}^{l} P(m_i | m_1 \ldots m_{i-1} m_1 \ldots m_{i-1})$$

  - – For complexity reasons, we simplify by considering only the n-1 preceding words of a word (namely a *ngram* model)

$$P(m_i | m_1 \ldots m_{i-1}) = P(m_i | m_{i-n+1} \ldots m_{i-1})$$

4. Language Models of IR

– Models

- Unigram $$P(s) = \prod_{i=1}^{l} P(m_i)$$

- Bigram $$P(s) = \prod_{i=1}^{l} P(m_i | m_{i-1}) = \prod_{i=1}^{l} \frac{P(m_{i-1}\, m_i)}{P(m_{i-1})}$$

- Trigram $$P(s) = \prod_{i=1}^{l} P(m_i | m_{i-2}\, m_{i-1}) = \prod_{i=1}^{l} \frac{P(m_{i-2}\, m_{i-1}\, m_i)}{P(m_{i-2}\, m_{i-1})}$$

- In IR, most approaches use unigrams

4. Language Models of IR

- Basic idea :

$$P(R=r \mid d,q) = P(q \mid \theta_d, R=r) \quad \textit{noted} \quad P(q \mid \theta_d)$$

meaning : what is the probability that a user who likes the document d should use the query q (to retrieve d)?

… but … how to estimate θ_d ?

4. Language Models of IR

- Several probability laws may be used for θ_d
  - Multinomial distribution
    - example : one urn with several marbles of c colors, several marbles of each color may appear. A sequence of colors selected (marble selected and put back) is modelled by a multinomial law of probability:

      p(c1, c2, c2)=p(c1)*p(c2)*p(c2)
    - we have $\sum_c p(c) = 1$
  - Multinomial distribution for documents [Song and Fei]:
    - here we compute the probability that the query terms get selected from the document
    - each word occurrence is independant
    - with V the vocabulary: $\sum_{t \in V} p(t|\theta_d) = 1$

$$P(q|\theta_d) = \frac{|q|!}{\prod_{t \in V}\left(\left|w_t^q\right|!\right)} \prod_{t \in V} p(t|\theta_d)^{w_t^q} \propto \prod_{t \in V} p(t|\theta_d)^{w_t^q}$$

4. Language Models of IR

- Several probability laws may be used for θ_d
  - Multiple Bernoulli
    - define a binary random variable X_t for each term t that indicates whether the term is present (X_t=1) or absent (X_t=0) in the query.
    - each word is considered independant
    - we have for each t: $p(X_t = 1|\theta_d) + p(X_t = 0|\theta_d) = 1$
    - the parameters are: $\theta_d = \{p(X_t = 1|\theta_d)\}_{t \in V}$

$$p(q|\theta_d) = \prod_{t \in q} P(X_t = 1|\theta_d) . \prod_{t \notin q} (1 - P(X_t = 1|\theta_d))$$

4. Language Models of IR

- We focus here on the Multinomial model (good results and more used in litterature)
- How to estimate the parameters of the model?
  - A simple solution: use the Maximum Likelihood estimate (MLE) to fit the statistical model to the data: We look for the $p(t|\theta_d)$ that maximize the probability to observe the document.

$$P_{ML}(t|\theta_d) = \frac{w_d^t}{\sum_{t \in V} w_d^t} = \frac{w_d^t}{|d|}$$

with w^t_d the count of t in d

respects the "multinomial constraint" : $\sum_{t \in V} P_{ML}(t|\theta_d) = \frac{\sum_{t \in V} w_d^t}{|d|} = \frac{|d|}{|d|} = 1$

4. Language Models of IR

- Is it done, so? Not really... consider
  - a vocabulary V={"day", "night", "sky"}
  - a document d so that θ_d={$p_{ML}(day|\ \theta_d)=0.67$, $p_{ML}(night|\ \theta_d)=0.33$, $p_{ML}(sky|\ \theta_d)=0$}
  - a query q="day sky"
  - then: $p(q|\ \theta_d) \propto\ p_{ML}(day|\ \theta_d)^1 * p_{ML}(sky|\ \theta_d)^1$

$$= 0.67 * 0$$

$$= 0 \quad \ldots!$$

even is the document matches partially the query!

4. Language Models of IR

- This problems comes from the fact that we used only the document source to model the probability distribution, and the document is not large enough to really contain all the needed data...

- So, P_{ML} is itself not sufficient for the language model of documents.

- One solution is to integrate data from a larger set, the <u>collection of documents</u>.

4. Language Models of IR

- The solution is to achieve probability *smoothing*
  - we smooth the p_{ML} by a probability coming from the corpus
  - mainly the probability coming from the corpus is defined as

$$P(t|C) = \frac{\sum_{d \in C} w_d^t}{\sum_{d \in C} \sum_{t \in V} w_d^t} = \frac{\sum_{d \in C} w_d^t}{\sum_{d \in C} |d|}$$

- Several smoothings exist (with different impact on retrieval quality), corresponding to several ways to manage the integration between the data from thre documents and the corpus

4. Language Models of IR

- Jelinek-Mercer smoothing
  - fixed coefficient interpolation

$$P_\lambda(t|\hat{\theta}_d) = (1-\lambda).P_{ML}(t|\theta_d) + \lambda.P(t|C)$$

  - one λ in [0, 1] for all the documents
  - when $\lambda=0$, we get P_{ML}, when $\lambda=1$ all document models are the same as the collection model.
  - Estimation with several values λ on one test collection.
  - simple to compute, good results.

4. Language Models of IR

- Jelinek smoothing guaranties the contraint related to multinomial distribution $\sum_{t \in V} p_\lambda(t|\hat{\theta}_d) = 1$?

- We have $$p_\lambda(t|\hat{\theta}_d) = (1-\lambda)\frac{w_d^t}{\sum_{t \in V} w_d^t} + \lambda \frac{\sum_{d \in C} w_d^t}{\sum_{d \in C}\sum_{t \in V} w_d^t}$$

- So $$\sum_{t \in V} p\lambda(t|\hat{\theta}_d) = (1-\lambda)\frac{\sum_{t \in V} w_d^t}{\sum_{t \in V} w_d^t} + \lambda \frac{\sum_{t \in V}\sum_{d \in C} w_d^t}{\sum_{d \in C}\sum_{t \in V} w_d^t}$$

$$= (1-\lambda) + \lambda$$

$$= 1$$

4. Language Models of IR

- Dirichlet smoothing
  - interpolation dependant of each document, with one parameter μ
  - considers that the corpus adds pseudo occurrences of terms (non integer) :

$$P_\mu(t|\hat{\theta}_d) = \frac{w_d^t + \mu P(t|C)}{|d| + \mu}$$

4. Language Models of IR

- Dirichlet smoothing
  - do we still get multinomial distributions?

$$P_{\mu}(t|\hat{\theta}_d) = \frac{w_d^t + \mu P(t|C)}{\sum_{t \in V} w_d^t + \mu}$$

  - Yes:

$$\sum_{t \in V} P_{\mu}(t|\hat{\theta}_d) = \frac{1}{\sum_{t \in V} w_d^t + \mu} . \sum_{t \in V} (w_d^t + \mu P(t|C))$$

$$= \frac{1}{\sum_{t \in V} w_d^t + \mu} . (\sum_{t \in V} w_d^t + \mu \sum_{t \in V} P(t|C))$$

$$= \frac{1}{\sum_{t \in V} w_d^t + \mu} . (\sum_{t \in V} w_d^t + \mu) = 1$$

4. Language Models of IR

- Dirichlet smoothing
  - relationship with Jelinek-Mercer smoothing

$$P_\mu(t|\hat{\theta}_d) = \frac{w_d^t + \mu P(t|C)}{|d| + \mu} = \frac{|d|}{|d| + \mu} . \frac{w_d^t}{|d|} + \frac{\mu}{|d| + \mu} P(t|C)$$

$$= \frac{|d|}{|d| + \mu} . P_{ML}(t|\theta_d) + \underbrace{\frac{\mu}{|d| + \mu}}_{\approx\lambda} P(t|C)$$

  - long documents have less smoothing (because more data)
  - Dirichlet smoothing: very good results (values around 1000 or greater).

4. Language Models of IR

- Why smoothing is important?
  - In fact, smoothing makes a link with IDF [Lafferty & Zhai 2001]
  - consider that a general smoothing is of the form

$$P_\mu(t|\hat{\theta}_d) = \begin{cases} p_s(t|\theta_d) & \text{if t in document d} \\ \alpha_d \mathrm{p}(\mathrm{t}|\mathrm{C}) & \text{otherwise} \end{cases}$$

method	$P_s(w\|\theta_d)$	α_d	Parameter
Jelinek-Mercer	$(1-\lambda).P_{ML}(t\|\theta_d)+\lambda.P(t\|C)$	λ	λ
Dirichlet	$\dfrac{w_d^t+\mu P(t\|C)}{\sum_{t\in V} w_d^t+\mu}$	$\dfrac{\mu}{\sum_{t\in V} w_d^t+\mu}$	μ

4. Language Models of IR

- Why smoothing is important?

$$
\begin{aligned}
\log P(q|\hat{\theta}_d) &=_{rank} \sum_{t\in V} w_t^q .\log p(t|\hat{\theta}_d) \\
&=_{rank} \sum_{t\in d} w_t^q .\log p_s(t|\theta_d) + \sum_{t\notin d} w_t^q .\log \alpha_d\, p(t|C) \\
&=_{rank} \sum_{t\in d} w_t^q .\log p_s(t|\theta_d) + \sum_{t\in V} w_t^q .\log \alpha_d\, p(t|C) - \sum_{t\in d} w_t^q .\log \alpha_d\, p(t|C) \\
&=_{rank} \sum_{t\in d} w_t^q .\log \frac{p_s(t|\theta_d)}{\alpha_d\, p(t|C)} + \underline{\sum_{t\in V} w_t^q .\log \alpha_d} + \cancel{\sum_{t\in V} w_t^q .\log p(t|C)}
\end{aligned}
$$

"similar" to TF.IDF

4. Language Models of IR

- Generalization of the original matching function, negative Kullback-Leibler divergence:

$$-KL(\theta_q \mid \hat{\theta}_d) = -\sum_{t \in V} P(t \mid \theta_q) \log \frac{P(t \mid \theta_q)}{P(t \mid \hat{\theta}_d)}$$

- KL divergence compares two probabilities distributions (relative entropy: how to code one distribution with another one)

4. Language Models of IR

- KL divergence on multinomial distributions of query and document and MLE similar to original matching:

$$-KL(\theta_q \| \hat{\theta}_d) = -\sum_{t \in V} P(t|\theta_q) \log \frac{P(t|\theta_q)}{P(t|\hat{\theta}_d)}$$

$$= -\sum_{t \in V} \frac{w_t^q}{|q|} \log P(t|\theta_q) + \sum_{t \in V} \frac{w_t^q}{|q|} \log P(t|\hat{\theta}_d)$$

$$=_{rank} \sum_{t \in V} w_t^q \log P(t|\hat{\theta}_d)$$

$$=_{rank} \log \prod_{t \in V} P(t|\hat{\theta}_d)^{w_t^q}$$

$$=_{rank} P(q|\hat{\theta}_d)$$

4. Language Models of IR

- The KL divergence considers by definition comparison of distributions, which seems closer to the usual meaning of matching in IR.

- KL is implemented as Language Model matching in Terrier and Lemur.

5. Conclusion

- Language models are state of the art IR
  - Multinomial
  - Dirichlet smoothing
  - Strong fundamentals, links to heuristics in IR (TF, IDF)

- Many extentions
  - cluster-based smoothing
  - other probability models (Poisson)
  - other smoothings

- LM state of the art, competing with BM 25.

Bibliography

- C. Zhai, Statistical Language Models for Information Retrieval, Morgan&Claypool, 2009
- Zhai&Lafferty, A Study of Smoothing Methods for Language Models Appplied to Ad Hoc Information Retrieval, ACM SIGIR 2001, pp334-342
- F. Song and W.B. Croft. A general language model for information retrieval. In Proceedings of Eighth International Conference on Information and Knowledge Management (CIKM'99), 1999.
- H. Turtle , W. B. Croft, Inference networks for document retrieval, Proceedings of the 13th annual international ACM SIGIR, p.1-24, September 05-07, 1990.
- J. Ponte and W.B. Croft, A Language Modeling Approach to Information Retrieval, ACM SIGIR 1998.

Bibliography

- S. Robertson and K. Spark Jones (1976), Relevance weighting of search terms. Journal of the American Society for Information Science. n°27. pp. 129-146.
- S. Robertson et al., Okapi at TREC-3, TREC-3 conference, 1994.
- A. Singhal, Modern Information Retrieval: A Brief Overview, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2001.
- M. Boughanem, W. Kraaj an J.Y. Nie, Modèles de langue pour la recherche d'information, *in* les systèmes de recherche d'information : modèles conceptuels, Hermes 2004.

– Use in IR – Model of Hiemstra

- Idea : $Score(D,Q) = P(D/Q) = P(D/t_1t_2...t_n)$ with $Q=t_1t_2\ldots t_n$

$$= P(D)\frac{P(t_1t_2...t_n/D)}{P(t_1t_2...t_n)}$$

- Hypotheses :
  - Independent query terms
- Notation : $P(t_1t_2\ldots t_n)=1/c$
- We obtain: $Score(D,Q) = cP(D)\prod_{t_i\in Q} P(t_i/D)$
  - We define

$$P(D) = \frac{|D|}{|C|}$$: Probability of the document

$$P(t_i/D) = \alpha_1.P_{ML}(t_i/D) + (1-\alpha_1).P_{ML}(t_i/C)$$

: Probability of a term knowing a document

– Use in IR – Model of Hiemstra

- Expansion of $P(t_i/D)$

$$P(t_i / D) = \alpha_1 . \frac{tf(t_i)}{\sum_t tf(t)} + (1-\alpha_1)\frac{df(t_i)}{\sum_t df(t)}$$

$$= (\alpha_1 . \frac{tf(t_i)}{\sum_t tf(t)} . \frac{\sum_t df(t)}{(1-\alpha_1).df(t_i)} + 1).(1-\alpha_1)\frac{df(t_i)}{\sum_t df(t)}$$

- So

$$Score(D,Q) = c.\frac{|D|}{|C|}.\prod_{t_i \in Q}\left((\alpha_1 . \frac{tf(t_i)}{\sum_t tf(t)} . \frac{\sum_t df(t)}{(1-\alpha_1).df(t_i)} + 1).(1-\alpha_1)\frac{df(t_i)}{\sum_t df(t)} \right)$$

– Used in IR – Model of Hiemstra

- We use logs

$$Score(D,Q) = c.\frac{|D|}{|C|}.\prod_{t_i \in Q}\left((\alpha_1.\frac{tf(t_i)}{\sum_t tf(t)}.\frac{\sum_t df(t)}{(1-\alpha_1).df(t_i)}+1).(1-\alpha_1)\frac{df(t_i)}{\sum_t df(t)}\right)$$

$$\log - Score(D,Q) = \log(c.\frac{|D|}{|C|}.\prod_{t_i \in Q}\left((\alpha_1.\frac{tf(t_i)}{\sum_t tf(t)}.\frac{\sum_t df(t)}{(1-\alpha_1).df(t_i)}+1).(1-\alpha_1)\frac{df(t_i)}{\sum_t df(t)}\right))$$

  – Constants elements for one query

$$\log - Score(D,Q) = \log(c) + \log(\frac{|D|}{|C|}) + \sum \log(\alpha_1.\frac{tf(t_i)}{\sum_t tf(t)}.\frac{\sum_t df(t)}{(1-\alpha_1).df(t_i)}+1) + \sum_{t_i \in Q}\log((1-\alpha_1)\frac{df(t_i)}{\sum_t df(t)})$$

  – So

$$\log(c), \quad \log(\frac{|D|}{|C|}), \qquad \text{and} \quad \sum_{t_i \in Q}\log((1-\alpha)\frac{df(t_i)}{\sum_i df(t)}$$

$$\log - Score(D,Q) \propto \boxed{\sum_{t_i \in Q}\log(\alpha_1.\frac{tf(t_i)}{\sum_t tf(t)}.\frac{\sum_t df(t)}{(1-\alpha_1).df(t_i)}+1)}$$

– Use in IR – Model of Hiemstra

- Typical value for α_1 : 0.15
- Defines a strong formal framework for IR
- Comparable results than the vector space model but possible extensions (example : good results on web pages)