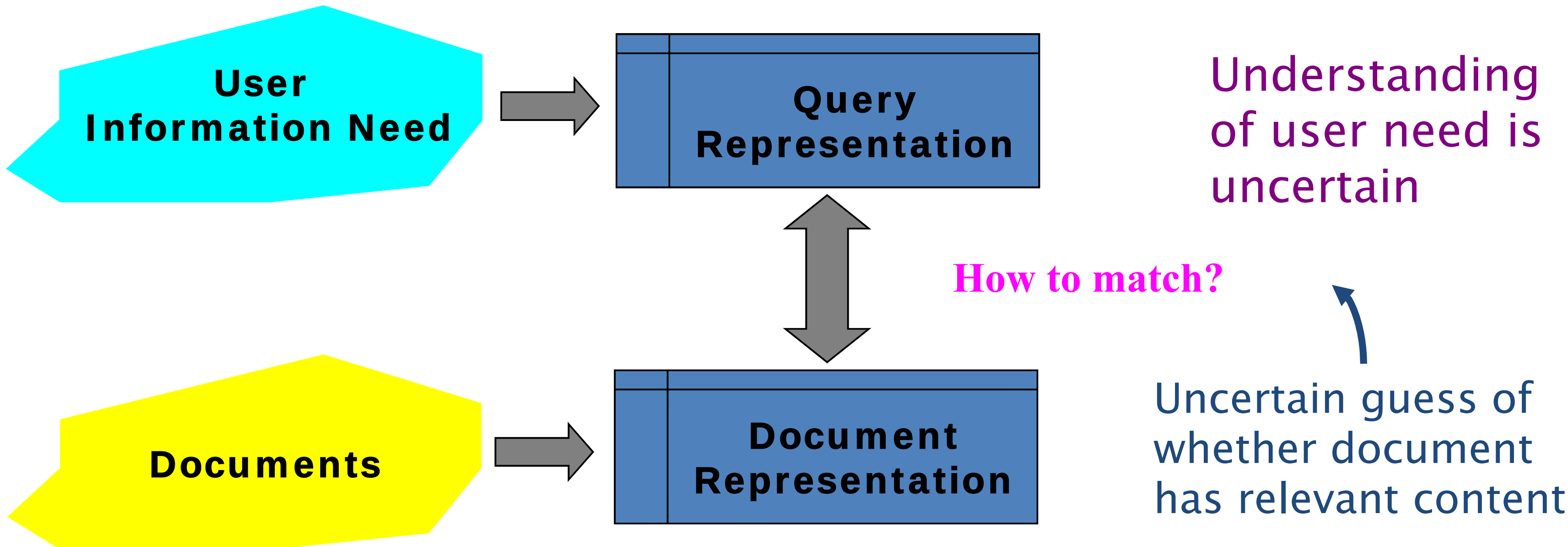
Probabilistic IR - Basic

SEEM5680

Taken from "Introduction to Information Retrieval" by C. Manning, P. Raghavan, and H. Schutze, Cambridge University Press.

Why probabilities in IR?



In traditional IR systems, matching between each document and query is attempted in a semantically imprecise space of index terms.

Probabilities provide a principled foundation for uncertain reasoning. *Can we use probabilities to quantify our uncertainties?*

The document ranking problem

- Recall a typical IR setting
  - We have a collection of documents
  - User issues a query
  - A list of documents needs to be returned
- **Ranking method is core of an IR system:**
  - **In what order do we present documents to the user?**
  - We want the "best" document to be first, second best second, etc….
- **Idea: Rank by probability of relevance of the document w.r.t. information need**
  - P(relevant|document$_i$, query)

The document ranking problem

- An example of a probabilistic model is BM25 which is still a popular IR model recently
- BM25 is also incorporated into tasks other than IR. For example:
  - “Distant Supervision for Multi-Stage Fine-Tuning in Retrieval-Based Question Answering”, The Web Conference (WWW), 2020.
  - “BERT-based Dense Retrievers Require Interpolation with BM25 for Effective Passage Retrieval”, ICTIR, 2021

Recall a few probability basics

- For events *a* and *b:*
- Bayes' Rule

$$p(a,b) = p(a \cap b) = p(a \mid b)\,p(b) = p(b \mid a)\,p(a)$$

$$p(\overline{a} \mid b)\,p(b) = p(b \mid \overline{a})\,p(\overline{a})$$

$$p(a \mid b) = \frac{p(b \mid a)\,p(a)}{p(b)} = \frac{p(b \mid a)\,p(a)}{\sum_{x=a,\overline{a}} p(b \mid x)\,p(x)}$$

Prior: $p(a)$

Posterior: $p(a \mid b)$

- Odds:

$$O(a) = \frac{p(a)}{p(\overline{a})} = \frac{p(a)}{1 - p(a)}$$

The Probability Ranking Principle

- A reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request.
- The overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data.
- The probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose.

Probability Ranking Principle

Let *x* be a document in the collection.
Let *R* represent **relevance** of a document w.r.t. given (fixed) query and let *NR* represent **non-relevance.**

R={0,1} vs. NR/R

Need to find p(*R*|*x*) - probability that a document *x* is **relevant.**

$$p(R \mid x) = \frac{p(x \mid R)\,p(R)}{p(x)}$$

$$p(NR \mid x) = \frac{p(x \mid NR)\,p(NR)}{p(x)}$$

p(*R*),p(*NR*) - prior probability of retrieving a (non) relevant document

$$p(R \mid x) + p(NR \mid x) = 1$$

p(*x*|*R*), p(*x*|*NR*) - probability that if a relevant (non-relevant) document is retrieved, it is *x*.

Probability Ranking Principle

- How do we compute all those probabilities?
  - Do not know exact probabilities, have to use estimates
  - Binary Independence Retrieval (BIR) – is the simplest model

Binary Independence Model

- **"Binary" = Boolean**: documents are represented as binary incidence vectors of terms (cf. lecture 1):

  $\vec{x} = (x_1, \ldots, x_n)$

  $x_i = 1$ <u>iff</u> term *i* is present in document *x*,

  otherwise 0
- **"Independence":** terms occur in documents independently
- Different documents can be modeled as same vector

Binary Independence Model

- Queries: binary term incidence vectors
- Given query ***q***,
  - for each document ***d*** need to compute ***p(R|q,d).***
  - replace with computing ***p(R|q,x)*** where ***x*** is binary term incidence vector representing ***d***
  - interested only in ranking
- Will use odds and Bayes' Rule:

$$O(R \mid q, \vec{x}) = \frac{p(R \mid q, \vec{x})}{p(NR \mid q, \vec{x})} = \frac{\dfrac{p(R \mid q)\, p(\vec{x} \mid R, q)}{p(\vec{x} \mid q)}}{\dfrac{p(NR \mid q)\, p(\vec{x} \mid NR, q)}{p(\vec{x} \mid q)}}$$

Binary Independence Model

- An example of a query vector and a document vector

$\boldsymbol{q}$	0	1	0	0	0	1	0	1
$\vec{x}$	0	1	1	0	1	0	0	1

- The notion of relevance (R) and non-relevance (NR) only depends on the concerned document.

Binary Independence Model

$$O(R \mid q, \vec{x}) = \frac{p(R \mid q, \vec{x})}{p(NR \mid q, \vec{x})} = \frac{p(R \mid q)}{p(NR \mid q)} \cdot \frac{p(\vec{x} \mid R, q)}{p(\vec{x} \mid NR, q)}$$

Constant for a given query

Needs estimation

- Using **Independence** Assumption:

$$\frac{p(\vec{x} \mid R, q)}{p(\vec{x} \mid NR, q)} = \prod_{i=1}^{n} \frac{p(x_i \mid R, q)}{p(x_i \mid NR, q)}$$

- So : $O(R \mid q, d) = O(R \mid q) \cdot \prod_{i=1}^{n} \frac{p(x_i \mid R, q)}{p(x_i \mid NR, q)}$

Binary Independence Model

$$O(R \mid q, d) = O(R \mid q) \cdot \prod_{i=1}^{n} \frac{p(x_i \mid R, q)}{p(x_i \mid NR, q)}$$

- Since x_i is either 0 or 1:

$$O(R \mid q, d) = O(R \mid q) \cdot \prod_{x_i=1} \frac{p(x_i = 1 \mid R, q)}{p(x_i = 1 \mid NR, q)} \cdot \prod_{x_i=0} \frac{p(x_i = 0 \mid R, q)}{p(x_i = 0 \mid NR, q)}$$

- Let $p_i = p(x_i = 1 \mid R, q); \quad r_i = p(x_i = 1 \mid NR, q);$

Binary Independence Model

- An example of a query vector and a document vector

$\boldsymbol{q}$	0	1	0	0	0	1	0	1
$\vec{x}$	0	1	1	0	1	0	0	1

- Recall:

  $p_i = p(x_i = 1|R, q)$; $r_i = p(x_i = 1|NR, q)$

- Assume:

For all terms not occurring in the query, i.e. $q_i = 0$,
then $p_i = r_i$ (also implying that $1 - p_i = 1 - r_i$)

- We can focus on terms appeared in the query, i.e. $q_i = 1$

Binary Independence Model

q	0	1	0	0	0	1	0	1
$\vec{x}$	0	1	1	0	1	0	0	1

$$O(R \mid q, \vec{x}) = O(R \mid q) \cdot \prod_{x_i = q_i = 1} \frac{p_i}{r_i} \cdot \prod_{\substack{x_i = 0 \\ q_i = 1}} \frac{1 - p_i}{1 - r_i}$$

All matching terms

Non-matching query terms

$$= O(R \mid q) \cdot \prod_{x_i = q_i = 1} \frac{p_i (1 - r_i)}{r_i (1 - p_i)} \cdot \prod_{q_i = 1} \frac{1 - p_i}{1 - r_i}$$

All matching terms

$x_i = 0,1$

All query terms

Binary Independence Model

$$O(R \mid q, \vec{x}) = O(R \mid q) \cdot \prod_{x_i = q_i = 1} \frac{p_i (1 - r_i)}{r_i (1 - p_i)} \cdot \prod_{q_i = 1} \frac{1 - p_i}{1 - r_i}$$

Constant for each query

Only quantity to be estimated for rankings

- Retrieval Status Value:

$$RSV = \log \prod_{x_i = q_i = 1} \frac{p_i (1 - r_i)}{r_i (1 - p_i)} = \sum_{x_i = q_i = 1} \log \frac{p_i (1 - r_i)}{r_i (1 - p_i)}$$

Binary Independence Model

- All boils down to computing RSV.

$$RSV = \log \prod_{x_i = q_i = 1} \frac{p_i(1-r_i)}{r_i(1-p_i)} = \sum_{x_i = q_i = 1} \log \frac{p_i(1-r_i)}{r_i(1-p_i)}$$

$$RSV = \sum_{x_i = q_i = 1} c_i; \qquad c_i = \log \frac{p_i(1-r_i)}{r_i(1-p_i)}$$

- All we need to do is to estimate c_i

Probability Estimates in Practice

- Recall:

  $p_i = p(x_i = 1|R, q)$; $r_i = p(x_i = 1|NR, q)$
- If non-relevant documents are approximated by the whole collection, then r_i (prob. of occurrence in non-relevant documents for query) is df_i/N and

  $log(1 - r_i) / r_i = log(N - df_i) / df_i = \log(N/ df_i) = \mathrm{IDF}$
- p_i (prob. of occurrence in relevant documents) can be set to a constant, e.g. Croft and Harper's combination match – set to 0.5.
- Finally the ranking is determined by which query terms occur in documents scaled by their idf weighting:

$$RSV = \sum_{x_i = q_i = 1} log \frac{N}{df_i}$$

BM25

- We can improve the basic probabilistic model by factoring in the frequency of each term and document length:

$$RSV = \sum_{x_i=q_i=1} log\left(\frac{N}{df_i}\right)\frac{(k_1+1)tf_i}{k_1\left((1-b)+b\left({}^{L}/_{L_{ave}}\right)\right)+tf_i}$$

where tf_i is the frequency of term i. L and L_{ave} are the length of the current document and the average document length for the whole collection. k_i is a positive tuning parameter that calibrates the document term frequency scaling. b is a parameter which determines the scaling by document length.