III.3 Probabilistic Retrieval Models

1. Probabilistic Ranking Principle
2. Binary Independence Model
3. Okapi BM25
4. Tree Dependence Model
5. Bayesian Networks for IR

Based on MRS Chapter 11
TF*IDF vs. Probabilistic IR vs. Statistical LMs

- **TF*IDF** and **VSM** produce sufficiently good results in practice but often criticized for being “too ad-hoc” or “not principled”

- Typically outperformed by **probabilistic retrieval models** and **statistical language models** in IR benchmarks (e.g., TREC)

- **Probabilistic retrieval models**
  - use generative models of documents as bags-of-words
  - explicitly model probability of relevance $P[R \mid d, q]$

- **Statistical language models**
  - use generative models of documents and queries as sequences-of-words
  - consider likelihood of generating query from document model or divergence of document model and query model (e.g., Kullback-Leibler)
Probabilistic Information Retrieval

- **Generative model**
  - **probabilistic mechanism** for producing documents (or queries)
  - usually based on a **family of parameterized probability distributions**

- **Powerful model** but restricted through practical limitations
  - often **strong independence assumptions** required for **tractability**
  - **parameter estimation** has to deal with **sparseness** of available data
    (e.g., collection with $M$ terms has $2^M$ distinct possible documents, but model parameters need to be estimated from $N << 2^M$ documents)
Multivariate Bernoulli Model

• For generating document $d$ from joint (multivariate) term distribution $\Phi$

• consider **binary random variables**: $d_t = 1$ if term in $d$, 0 otherwise

• postulate **independence** among these random variables

$$P[d|\Phi] = \prod_{t \in V} \phi^d_t (1 - \phi^{1-d}_t)$$

$$\phi_t = P[\text{term } t \text{ occurs in a document}]$$

• **Problems**:
  
  • underestimates probability of short documents
  
  • product for absent terms underestimates probability of likely documents
  
  • too much probability mass given to very unlikely term combinations
1. Probability Ranking Principle (PRP)

“If 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, where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data.”

[van Rijsbergen 1979]

- **PRP with costs** [Robertson 1977] defines cost of retrieving \(d\) as the next result in a ranked list for query \(q\) as

  \[
  \text{cost}(d, q) = C_1 P[R|d, q] + C_0 P[\bar{R}|d, q]
  \]

  with **cost constants**

  - \(C_1\) as cost of retrieving a relevant document
  - \(C_2\) as cost of retrieving an irrelevant document

  - For \(C_1 < C_0\), cost is minimized by choosing \(\text{arg max}_d P[R|d, q]\)
Derivation of Probability Ranking Principle

• Consider document \(d\) to be retrieved next, because it is preferred (i.e., has lower cost) over all other candidate documents \(d'\)

\[
\begin{align*}
\text{cost}(d, q) & \leq \text{cost}(d', q) \\
\iff C_1 P[R|d, q] + C_0 P[\bar{R}|d, q] & \leq C_1 P[R|d', q] + C_0 P[\bar{R}|d', q] \\
\iff C_1 P[R|d, q] + C_0 (1 - P[R|d, q]) & \leq C_1 P[R|d', q] + C_0 (1 - P[R|d', q]) \\
\iff C_1 P[R|d, q] - C_0 P[R|d, q] & \leq C_1 P[R|d', q] - C_0 P[R|d', q] \\
\iff (C_1 - C_0) P[R|d, q] & \leq (C_1 - C_0) P[R|d', q] \\
\iff P[R|d, q] & \geq P[R|d', q] \quad \text{(assuming } C_1 < C_0) 
\end{align*}
\]
Probability Ranking Principle (cont’d)

• Probability ranking principle makes **two strong assumptions**
  
  • $P[R | d, q]$ can be **determined accurately**
  
  • $P[R | d, q]$ and $P[R | d’, q]$ are **pairwise independent** for documents $d, d’$

• **PRP without costs** (based on Bayes’ optimal decision rule)
  
  • returns **set of documents** $d$ for which $P[R | d, q] > (1 - P[R | d, q])$
  
  • minimizes the **expected loss** (aka. Bayes’ risk) under the 1/0 loss function
2. Binary Independence Model (BIM)

- **Binary independence model** [Robertson and Spärck-Jones 1976] has traditionally been used with the probabilistic ranking principle

- **Assumptions:**
  - relevant and irrelevant documents **differ in their term distribution**
  - probabilities of term occurrences are **pairwise independent**
  - documents are **sets of terms**, i.e., **binary term weights** in \{0,1\}
  - **non-query terms** have the same probability of occurring in relevant and non-relevant documents
  - **relevance** of a document is **independent** of relevance **others document**
Ranking Proportional to Relevance Odds

\[ O(R|d) = \frac{P[R|d]}{P[R|d]} \quad \text{(odds for ranking)} \]

\[ = \frac{P[d|R] \times P[R]}{P[d|\bar{R}] \times P[R]} \quad \text{(Bayes’ theorem)} \]

\[ \propto \frac{P[d|R]}{P[d|R]} \quad \text{(rank equivalence)} \]

\[ = \prod_{t \in V} \frac{P[d_t|R]}{P[d_t|\bar{R}]} \quad \text{(independence assumption)} \]

\[ = \prod_{t \in q} \frac{P[d_t|R]}{P[d_t|\bar{R}]} \quad \text{(non-query terms)} \]

\[ = \prod_{t \in d} \frac{P[D_t|R]}{P[D_t|R]} \times \prod_{t \notin d} \frac{P[\bar{D}_t|R]}{P[D_t|R]} \]

with \( d_t \) indicating if document \( d \) includes term \( t \)
and \( D_t \) indicating if random document includes term \( t \)
Ranking Proportional to Relevance Odds (cont’d)

\[
\begin{align*}
&= \prod_{t \in q} \frac{P[D_t|R]}{P[D_t]} \times \prod_{t \notin q} \frac{P[D_t|R]}{P[D_t]} \\
&= \prod_{t \in q} \frac{p_t}{q_t} \times \prod_{t \notin q} \frac{1-p_t}{1-q_t} \\
&= \prod_{t \in q} \frac{p_t^{d_t}}{q_t^{d_t}} \times \prod_{t \notin q} \frac{(1-p_t)^{1-d_t}}{(1-q_t)^{1-d_t}} \\
\propto \sum_{t \in q} \log \left( \frac{p_t^{d_t} (1-p_t)}{(1-p_t)^{d_t}} \right) - \log \left( \frac{q_t^{d_t} (1-q_t)}{(1-q_t)^{d_t}} \right) \\
&= \sum_{t \in q} d_t \log \frac{p_t}{1-p_t} + \sum_{t \in q} d_t \log \frac{1-q_t}{q_t} + \sum_{t \in q} \log \frac{1-p_t}{1-q_t} \\
\propto \sum_{t \in q} d_t \log \frac{p_t}{1-p_t} + \sum_{t \in q} d_t \log \frac{1-q_t}{q_t} \\
&= \left( \text{shortcuts } p_t \text{ and } q_t \right)

\]
Estimating $p_t$ and $q_t$ with a Training Sample

• We can estimate $p_t$ and $q_t$ based on a **training sample** obtained by evaluating the query $q$ on a **small sample of the corpus** and asking the user for **relevance feedback** about the results.

• Let $N$ be the # documents in our sample
  - $R$ be the # relevant documents in our sample
  - $n_t$ be the # documents in our sample that contain $t$
  - $r_t$ be the # relevant documents in our sample that contain $t$

we estimate

\[
p_t = \frac{r_t}{R} \quad q_t = \frac{n_t - r_t}{N - R}
\]

or with **Lidstone smoothing** ($\lambda = 0.5$)

\[
p_t = \frac{r_t + 0.5}{R + 1} \quad q_t = \frac{n_t - r_t + 0.5}{N - R + 1}
\]
Smoothing (with Uniform Prior)

- Probabilities $p_t$ and $q_t$ for term $t$ are estimated by MLE for Binomial distribution
  - repeated coin tosses for term $t$ in relevant documents ($p_t$)
  - repeated coin tosses for term $t$ in irrelevant documents ($q_t$)
- Avoid overfitting to the training sample by smoothing estimates
  - Laplace smoothing (based on Laplace’s law of succession)
    \[
    p_t = \frac{r_t + 1}{R + 2} \quad q_t = \frac{n_t - r_t + 1}{N - R + 2}
    \]
  - Lidstone smoothing (heuristic generalization with $\lambda > 0$)
    \[
    p_t = \frac{r_t + \lambda}{R + 2 \lambda} \quad q_t = \frac{n_t - r_t + \lambda}{N - R + 2 \lambda}
    \]
Binary Independence Model (Example)

- Consider query \( q = \{t_1, \ldots, t_6\} \) and sample of four documents

<table>
<thead>
<tr>
<th></th>
<th>( t_1 )</th>
<th>( t_2 )</th>
<th>( t_3 )</th>
<th>( t_4 )</th>
<th>( t_5 )</th>
<th>( t_6 )</th>
<th>( R )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d_1 )</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( d_2 )</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( d_3 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( d_4 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( n_t )</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>( r_t )</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>( p_t )</td>
<td>( 5/6 )</td>
<td>( 1/2 )</td>
<td>( 1/2 )</td>
<td>( 5/6 )</td>
<td>( 1/2 )</td>
<td>( 1/6 )</td>
<td></td>
</tr>
<tr>
<td>( q_t )</td>
<td>( 1/6 )</td>
<td>( 1/6 )</td>
<td>( 1/2 )</td>
<td>( 1/2 )</td>
<td>( 1/2 )</td>
<td>( 1/6 )</td>
<td></td>
</tr>
</tbody>
</table>

- For document \( d_6 = \{t_1, t_2, t_6\} \) we obtain

\[
P[R|d_6, q] \propto \log 5 + \log 1 + \log \frac{1}{5} + \log 5 + \log 5 + \log 5
\]
Estimating $p_t$ and $q_t$ without a Training Sample

- When **no training sample** is available, we estimate $p_t$ and $q_t$ as

$$p_t = (1 - p_t) = \frac{1}{2} \quad q_t = \frac{df_t}{|D|}$$

- $p_t$ reflects that we have **no information about relevant documents**
- $q_t$ under the assumption that 
  \[
  \# \text{ relevant documents} \ll \# \text{ documents}
  \]

- When we plug in these estimates of $p_t$ and $q_t$, we obtain

$$P[R|d, q] = \sum_{t \in q} d_t \log 1 + \sum_{t \in q} d_t \log \frac{|D| - df_t}{df_t} \approx \sum_{t \in q} d_t \log \frac{|D|}{df_t}$$

which can be seen as **TF*IDF** with binary term frequencies and logarithmically dampened inverse document frequencies
Poisson Model

• For generating document \( d \) from joint (multivariate) term distribution \( \Phi \)

• consider **counting random variables**: \( d_t = tf_{t,d} \)

• postulate **independence** among these random variables

• **Poisson model** with term-specific parameters \( \mu_t \):

\[
P[d|\mu] = \prod_{t \in V} \frac{e^{-\mu_t} \cdot \mu_t^{d_t}}{d_t!} = e^{-\sum_{t \in V} \mu_t} \prod_{t \in d} \frac{\mu_t^{d_t}}{d_t!}
\]

• MLE for \( \mu_t \) from \( n \) sample documents \( \{d_1, \ldots, d_n\} \): \( \hat{\mu}_t = \frac{1}{n} \sum_{i=1}^{n} t_{f_{t,d_i}} \)

• no penalty for absent words

• no control of document length
3. Okapi BM25

• Generalizes term weight

\[ w = \log \frac{p(1 - q)}{q(1 - p)} \]

into

\[ w = \log \frac{p_{tf}q_0}{q_{tf}p_0} \]

where \( p_i \) and \( q_i \) denote the probability that term occurs \( i \) times in a relevant or irrelevant document, respectively.

• Postulates Poisson (or 2-Poisson-mixture) distributions for terms

\[ p_{tf} = e^{-\lambda} \frac{\lambda^{tf}}{tf!} \quad q_{tf} = e^{-\mu} \frac{\mu^{tf}}{tf!} \]
Okapi BM25 (cont’d)

• Reduces the number of parameters that have to be learned and approximates Poisson model by similarly-shaped function

\[ w = \frac{tf}{k_1 + tf} \log \frac{p(1 - q)}{q(1 - p)} \]

• Finally leads to Okapi BM25 as state-of-the-art retrieval model (with top-ranked results in TREC)

\[ w_{t,d} = \frac{(k_1 + tf_{t,d})}{k_1((1 - b) + b \frac{|d|}{avdl}) + tf_{t,d}} \log \frac{|D| - df_j + 0.5}{df_j + 0.5} \]

• \( k_1 \) controls impact of term frequency (common choice \( k_1 = 1.2 \))

• \( b \) controls impact of document length (common choice \( b = 0.75 \))
Okapi BM25 (Example)

- 3D plot of a simplified BM25 scoring function using $k_1 = 1.2$ as parameter (DF mirrored for better readability)

- Scores for $df_t > N/2$ are negative

$$w_t = \frac{(k_1 + 1) \cdot tf_{t,d}}{k_1 + tf_{t,d}} \log \frac{|D| - df_t + 0.5}{df_t + 0.5}$$
4. Tree Dependence Model

- Consider term correlations in documents (with binary RV $X_i$) requires estimating \textit{m-dimensional} probability distribution

$$P[X_1 = .., \ldots, X_m = ..] = f_X(X_1, \ldots, X_m)$$

- \textbf{Tree dependence model} [van Rijsbergen 1979]

  - considers \textbf{only 2-dimensional probabilities} for term pairs $(i,j)$

$$f_{ij}(X_i, X_j) = P[X_i = .., X_j = ..] = \sum_{X_1} \ldots \sum_{X_i-1} \sum_{X_i+1} \ldots \sum_{X_j-1} \sum_{X_j+1} \ldots \sum_{X_m} P[X_1 = .., \ldots, X_m = ..]$$

  - estimates for each $(i,j)$ the \textbf{error made by independence assumptions}
  
  - constructs a tree with \textbf{terms as nodes} and \textbf{$m-1$ weighted edges} connecting the \textbf{highest-error term pairs}
Two-Dimensional Term Correlations

- **Kullback-Leibler divergence** estimates error of approximating $f$ by $g$ assuming pairwise term independence

$$\epsilon(f, g) = \sum_{\bf{x} \in \{0,1\}^m} f(\bf{x}) \log \frac{f(\bf{x})}{g(\bf{x})} = \sum_{\bf{x} \in \{0,1\}^m} f(\bf{x}) \log \frac{f(\bf{x})}{\prod_{i=1}^m g(X_i)}$$

- **Correlation coefficient** for term pairs

$$\rho(X_i, X_j) = \frac{\text{Cov}(X_i, X_j)}{\sqrt{\text{Var}(X_i)} \sqrt{\text{Var}(X_j)}}$$

- $p$-values of $X^2$ test of independence
Kullback-Leibler Divergence (Example)

- Given are documents \( d_1 = (1,1), d_2 = (0,0), d_3 = (1,1), d_4 = (0,1) \)

- **2-dimensional probability distribution** \( f \):
  \[
  f(1,1) = P[X_1 = 1, X_2 = 1] = \frac{2}{4} \\
  f(0,0) = \frac{1}{4}, f(0,1) = \frac{1}{4}, f(1,0) = 0
  \]

- **1-dimensional marginal distributions** \( g_1 \) and \( g_2 \)
  \[
  g_1(1) = P[X_1 = 1] = \frac{2}{4}, g_1(0) = \frac{2}{4} \\
  g_2(1) = P[X_2 = 1] = \frac{3}{4}, g_2(0) = \frac{1}{4}
  \]

- **2-dimensional probability distribution** assuming independence
  \[
  g(1,1) = g_1(1) g_2(1) = \frac{3}{8} \\
  g(0,0) = \frac{1}{8}, g(0,1) = \frac{3}{8}, g(1,0) = \frac{1}{8}
  \]

- **approximation error** \( \varepsilon \) (Kullback-Leibler divergence)
  \[
  \varepsilon = \frac{2}{4} \log \frac{4}{3} + \frac{1}{4} \log 2 + \frac{1}{4} \log \frac{2}{3} + 0
  \]
Constructing the Term Dependence Tree

- **Input**: Complete graph \((V, E)\) with \(m\) nodes \(X_i \in V\) and \(m^2\) undirected edges \((i, j) \in E\) with weights \(\varepsilon\)

- **Output**: Spanning tree \((V, E')\) with maximum total edge weight

- **Algorithm**:
  - **Sort** \(m^2\) edges in descending order of weights
  - \(E' = \emptyset\)
  - **Repeat until** \(|E'| = m-1\)
    - \(E' = E' \cup \{(i, j) \in E \setminus E' \mid (i, j)\) has maximal weight and \(E'\) remains acyclic\}

- **Example**:

```
+---+---+---+
| web | surf | net |
+---+---+---+
| 0.7| 0.5 | 0.9 |
+---+---+---+
| 0.5| 0.1 | 0.1 |
+---+---+---+
| 0.3| 0.1 | 0.1 |
+---+---+---+
```

```
+---+---+
| web | surf |
+---+---+
| net | swim |
+---+---+
```

```
Estimation with Term Dependence Tree

- Given a term dependence tree \((V = \{X_1, \ldots, X_m\}, E')\) with preorder-labeled nodes (i.e., \(X_1\) is root) and assuming that \(X_i\) and \(X_j\) are independent for \((i, j) \notin E'\)

\[
P[X_1 = \ldots, X_m = .] = P[X_1 = .] P[X_2 = \ldots, X_m = .| X_1 = .] \quad \text{(conditional probability)}
\]

\[
= \prod_{i=1}^{m} P[X_i = .| X_1 = \ldots, X_{i-1} = .] \quad \text{(chain rule)}
\]

\[
= P[X_1] \prod_{(i,j) \in E'} P[X_j|X_i] \quad \text{(independence assumption)}
\]

\[
= P[X_1] \prod_{(i,j) \in E'} \frac{P[X_j, X_i]}{P[X_i]} \quad \text{(conditional probability)}
\]

- Example:

\[
P[\text{web, net, surf, swim}] = P[\text{web}] P[\text{net}|\text{web}] P[\text{surf}|\text{web}] P[\text{swim}|\text{surf}]
\]
5. Bayesian Networks

• A Bayesian network (BN) is a **directed, acyclic graph** \((V, E)\) with

  • Vertices \(V\) representing **random variables**
  
  • Edges \(E\) representing **dependencies**
  
  • For a root \(R \in V\) the BN captures the **prior probability** \(P[R = \ldots]\)

  • For a vertex \(X \in V\) with **parents** \(\text{parents}(x) = \{P_1, \ldots, P_k\}\)
    the BN captures the **conditional probability** \(P[X|P_1, \ldots, P_k]\)

  • The vertex \(X\) is **conditionally independent** of a non-parent node \(Y\)
    given its parents \(\text{parents}(x) = \{P_1, \ldots, P_k\}\), i.e.:

    \[
    P[X|P_1, \ldots, P_k, Y] = P[X|P_1, \ldots, P_k]
    \]
Bayesian Networks (cont’d)

• We can determine any **joint probability** using the BN

\[ P[X_1, \ldots, X_n] \]

\[ = P[X_1 | X_2, \ldots, X_n] P[X_2, \ldots, X_n] \]

\[ = \prod_{i=1}^{n} P[X_i | X_{i+1}, \ldots, X_n] \quad \text{(chain rule)} \]

\[ = \prod_{i=1}^{n} P[X_i | \text{parents}(X_i), \text{other nodes}] \quad \text{(conditional independence)} \]

\[ = \prod_{i=1}^{n} P[X_i | \text{parents}(X_i)] \]
Bayesian Networks (Example)

<table>
<thead>
<tr>
<th>Cloudy</th>
<th>P[C]</th>
<th>P[⋯]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sprinklers</th>
<th>P[S]</th>
<th>P[⋯]</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>T</td>
<td>0.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rain</th>
<th>C</th>
<th>P[R]</th>
<th>P[⋯]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>0.8</td>
<td>0.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wet</th>
<th>S</th>
<th>R</th>
<th>P[W]</th>
<th>P[⋯]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F</td>
<td>F</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>T</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>F</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>T</td>
<td>1.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

\[
P[C, S, \bar{R}, W] = P[C] P[S|C] P[\bar{R}|C] P[W|S, \bar{R}] = 0.5 \times 0.1 \times 0.2 \times 0.9
\]
Bayesian Networks for IR

\[ P[q|parents(q)] = \begin{cases} 
1 & : \exists t \in parents(q) : rel(t, q) \\
0 & : \text{otherwise}
\end{cases} \]

\[ P[q, d_j] = \sum_{(t_1, \ldots, t_M)} P[q, d_j, t_1, \ldots, t_M] \]
\[ = \sum_{(t_1, \ldots, t_M)} P[q|d_j, t_1, \ldots, t_M] P[d_j, t_1, \ldots, t_M] \]
\[ = \sum_{(t_1, \ldots, t_M)} P[q|t_1, \ldots, t_M] P[t_1, \ldots, t_M|d_j] P[d_j] \]

\[ P[d_j] = 1/N \]

\[ P[t_i|d_j] = \begin{cases} 
1 & : t_i \in d_j \\
0 & : \text{otherwise}
\end{cases} \]
Advanced Bayesian Networks for IR

- BN not widely adopted in IR due to challenges in parameter estimation, representation, efficiency, and practical effectiveness

\[ P[c_k | t_i, t_l] \approx \frac{df_{il}}{df_i + df_l - df_{il}} \]
Summary of III.3

- **Probabilistic IR** as a family of (more) principled approaches relying on generative models of documents as bags of words

- **Probabilistic ranking principle** as the foundation establishing that ranking documents by $P[R \mid d, q]$ is optimal

- **Binary independence model** puts that principle into practice based on a multivariate Bernoulli model

- **Smoothing** to avoid overfitting to the training sample

- **Okapi BM25** as a state-of-the-art retrieval model based on an approximation of a 2-Poisson mixture model

- **Term dependence model** and **Bayesian networks** can consider term correlations (but are often intractable)
Additional Literature for III.3


- S.E. Robertson, K. Spärck Jones: Relevance Weighting of Search Terms, JASIS 27(3), 1976

- S.E. Robertson, S. Walker: Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval, SIGIR 1994


- K. J. van Rijsbergen: Information Retrieval, University of Glasgow, 1979 http://www.dcs.gla.ac.uk/Keith/Preface.html