SIT330-770: Natural Language Processing

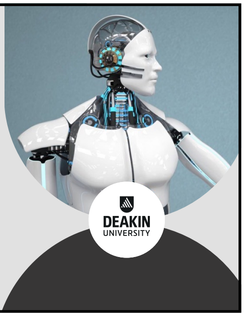
Week 2 - Information Retrieval Part 2

Probabilistic IR & IR Evaluation methods

Dr. Mohamed Reda Bouadjenek

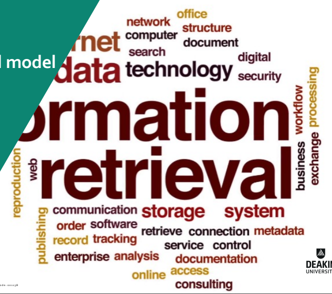
School of Information Technology, Faculty of Sci Eng & Built Env

reda.bouadjenek@deakin.edu.au



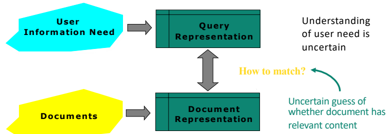
1

Probabilistic retrieval model



2

Why probabilities in IR?

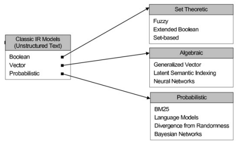


- In traditional IR systems, matching between each document and query is attempted in a semantically imprecise space of index terms
- Probabilities provide a principled foundation for uncertain reasoning.
- Can we use probabilities to quantify our search uncertainties?

3

Probabilistic IR topics

1. Classical probabilistic retrieval model
   - Probability ranking principle, etc.
   - Binary independence model (≈ Naïve Bayes text cat)
   - (Okapi) BM25
2. Bayesian networks for text retrieval
3. Language model approach to IR
   - An important development



- Probabilistic methods are one of the oldest but also one of the currently hot topics in IR
  - Traditionally: neat ideas, but didn't win on performance
  - It seems to be different now

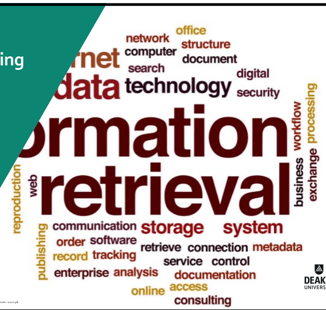
4

The document ranking problem

- We have a collection of documents
- User issues a query
- A list of documents needs to be returned
- **Ranking method is the core of modern IR systems:**
  - **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(R=1|document$_i$, query)

5

The Probability Ranking Principle (PRP)



6

The Probability Ranking Principle (PRP)

- The probabilistic model
  - Tries to estimate the probability that a document will be relevant to a user query
  - Assumes that this probability depends on the query and document representations only
  - The ideal answer set, referred to as **R**, should maximize the probability of relevance
- But,
  - How to compute these probabilities?
  - What is the sample space?

7

7

Recall a few probability basics

- For events A and B:

$$p(A,B) = p(A \cap B) = p(A|B)p(B) = p(B|A)p(A)$$

- **Bayes' Theorem**

$$\underbrace{p(\theta|x)}_{\text{Posterior probability}} = \frac{\overbrace{p(\theta,x)}^{\text{Joint probability}}}{p(x)} = \frac{\overbrace{p(x|\theta)}^{\text{Likelihood probability}}\underbrace{p(\theta)}_{\text{Prior probability}}}{p(x)}$$

- **Theorem of Total Probability**
  - Let $\theta_1, \theta_2 \cdots \theta_n$ be a set of mutually exclusive events (i.e., $\theta_i \cap \theta_j = 0$) and x is the union of N mutually exclusive events, then:

$$p(x) = \sum_{i=0}^{n} p(x|\theta_i)p(\theta_i)$$

- By substitution we get:

$$p(\theta|x) = \frac{p(\theta,x)}{p(x)} = \frac{p(x|\theta)p(\theta)}{\sum_{i=0}^{n} p(x|\theta_i)p(\theta_i)}$$

- Odds:

$$O(A) = \frac{p(A)}{p(\bar{A})} = \frac{p(A)}{1 - p(A)}$$

8

8

The Probability Ranking Principle (PRP)

- Let d represent a document in the collection
- Let $\boldsymbol{R}$ represent **relevance** of a document w.r.t. given (fixed) query and let $\boldsymbol{R = 1}$ represent relevant and $\boldsymbol{R = 1}$ not relevant
- Need to find $p(R = 1|d)$ – probability that a document d is **relevant**
- $p(R = 1|d) = \frac{p(d|R=1)p(R=1)}{p(d)}$
- $p(R = 0|d) = \frac{p(d|R=0)p(R=0)}{p(d)}$
- $p(R = 0|d) + p(R = 1|d) = 1$

$p(R = 1), p(R = 0)$ - prior probability of retrieving a relevant or non-relevant document at random

$p(d|R = 1), p(d|R = 0)$ - probability that if a relevant (not relevant) document is retrieved, it is d

9

9

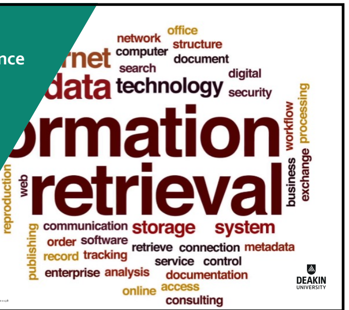
Probabilistic Retrieval Strategy

- First, estimate how each term contributes to relevance
  - How do other things like term frequency and document length influence your judgments about document relevance?
    - Not at all in BIM
    - A more nuanced answer is given by BM25
- Combine to find document relevance probability
- Order documents by decreasing probability
- Theorem: Using the PRP is optimal, in that it minimizes the loss (Bayes risk) under 1/0 loss
  - Provable if all probabilities correct, etc. [e.g., Ripley 1996]

10

10

The Binary Independence Model (BIM)



11

11

Binary Independence Model

- Traditionally used in conjunction with PRP
- **"Binary" = Boolean**: documents are represented as binary incidence vectors of terms:
  - $d = (t_1, \cdots, t_n)$
  - t_i iff term i is present in document d
- **"Independence"**: terms occur in documents independently
- Different documents can be modeled as the same vector

12

12

Binary Independence Model

- Queries: binary term incidence vectors
- Given query ***q***,
  - for each document ***d*** need to compute ***p(R|q,d)***
  - Interested only in ranking
- Will use odds and Bayes' Rule:

$$O(R|q,d) = \frac{p(R=1|q,d)}{p(R=0|q,d)} = \frac{\frac{p(R=1|q)p(d|R=1,q)}{p(d|q)}}{\frac{p(R=0|q)p(d|R=0,q)}{p(d|q)}}$$

13

Deakin University CRICOS Provider Code: 00113B

13

Binary Independence Model

$$O(R|q,d) = \frac{p(R=1|q,d)}{p(R=0|q,d)} = \frac{p(R=1|q)}{p(R=0|q)} \times \frac{p(d|R=1,q)}{p(d|R=0,q)}$$

Constant for a given query

Needs estimation

- Using Independence Assumption:

$$\frac{p(d|R=1,q)}{p(d|R=0,q)} = \prod_{i=1}^{n} \frac{p(t_i|R=1,q)}{p(t_i|R=0,q)}$$

$$O(R|q,d) = O(R|q) \times \prod_{i=1}^{n} \frac{p(t_i|R=1,q)}{p(t_i|R=0,q)}$$

14

Deakin University CRICOS Provider Code: 00113B

14

Binary Independence Model

$$O(R|q,d) = O(R|q) \times \prod_{i=1}^{n} \frac{p(t_i|R=1,q)}{p(t_i|R=0,q)}$$

- Since t_i is either 0 or 1:

$$O(R|q,d) = O(R|q) \times \prod_{t_i=1} \frac{p(t_i|R=1,q)}{p(t_i|R=0,q)} \times \prod_{t_i=0} \frac{p(t_i|R=1,q)}{p(t_i|R=0,q)}$$

- Let $p_i = p(t_i|R=1,q)$ and $u_i = p(t_i|R=0,q)$
- Assume, for all terms not occurring in the query ($q_i = 0$) $p_i = u_i$:

$$O(R|q,d) = O(R|q) \times \prod_{\substack{t_i=1 \\ q_i=1}} \frac{p_i}{u_i} \times \prod_{\substack{t_i=0 \\ q_i=1}} \frac{(1-p_i)}{(1-u_i)}$$

15

Deakin University CRICOS Provider Code: 00113B

15

Binary Independence Model

$$O(R|q,d) = O(R|q) \times \prod_{\substack{t_i=1 \\ q_i=1}} \frac{p_i}{u_i} \times \prod_{\substack{t_i=0 \\ q_i=1}} \frac{(1-p_i)}{(1-u_i)}$$

All matching terms

Non-matching query terms

$$O(R|q,\vec{x}) = O(R|q) \times \prod_{\substack{t_i=1 \\ q_i=1}} \frac{p_i}{u_i} \times \prod_{\substack{t_i=1 \\ q_i=1}} \left(\frac{1-u_i}{1-p_i} \cdot \frac{1-p_i}{1-u_i}\right) \times \prod_{\substack{t_i=0 \\ q_i=1}} \frac{(1-p_i)}{(1-u_i)}$$

$$O(R|q,\vec{x}) = O(R|q) \times \prod_{\substack{t_i=1 \\ q_i=1}} \frac{p_i(1-u_i)}{u_i(1-p_i)} \times \prod_{q_i=1} \frac{(1-p_i)}{(1-u_i)}$$

All matching terms

All query terms

16

Deakin University CRICOS Provider Code: 00113B

16

Binary Independence Model

$$O(R|q,d) = O(R|q) \times \prod_{t_i=q_i=1} \frac{p_i(1-u_i)}{u_i(1-p_i)} \times \prod_{q_i=1} \frac{(1-p_i)}{(1-u_i)}$$

Constant for each query

Only quantity to be estimated for rankings

- Retrieval Status Value:

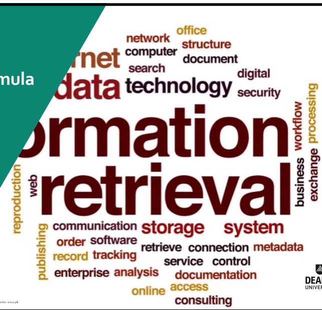
$$RSV = \log \prod_{t_i=q_i=1} \frac{p_i(1-u_i)}{u_i(1-p_i)} = \sum_{t_i=q_i=1} \log \frac{p_i(1-u_i)}{u_i(1-p_i)} \approx Sim(q,d)$$

17

Deakin University CRICOS Provider Code: 00113B

17

The BIM Ranking formula



18

Deakin University CRICOS Provider Code: 00113B

18

Term Incidence Contingency Table

- Retrieval Status Value:

$$Sim(q,d) \approx RSV = \sum_{t_i = q_i = 1} \log \frac{p_i(1-u_i)}{u_i(1-p_i)}$$

- Let,
  - N be the number of documents in the collection
  - n_i be the number of documents that contain term t_i
  - R be the total number of relevant docs to query q
  - r_i be the number of relevant docs that contain term t_i

- Based on these variables, we can build the following contingency table:

	Relevant	Non-relevant	Total
docs that contain t_i	r_i	$n_i - r_i$	n_i
docs that do not contain t_i	$R - r_i$	$N - n_i - (R - r_i)$	$N - n_i$
Total	R	$N - R$	N

If information on the contingency table were available for a given query, we could write:

$p_i = \frac{r_i}{R}$ (the probability of a term appearing in a document relevant to the query)

$u_i = \frac{n_i - r_i}{N-R}$ (the probability of a term appearing in a non-relevant document)

19

Deakin University CRICOS Provider Code: 00113B

19

Ranking Formula

- Then, the equation for ranking computation in the probabilistic model could be rewritten as

$$Sim(q,d) \approx \sum_{t_i = q_i = 1} \log \left(\frac{r_i}{R - r_i} \times \frac{N - n_i - R + r_i}{n_i - r_i} \right)$$

- For handling small values of r_i, we add 0.5 to each of the terms in the formula above, which changes $Sim(q,d)$ into

$$Sim(q,d) \approx \sum_{t_i = q_i = 1} \log \left(\frac{r_i + 0.5}{R - r_i + 0.5} \times \frac{N - n_i - R + r_i + 0.5}{n_i - r_i + 0.5} \right)$$

- This formula is considered as the classic ranking equation for the probabilistic model and is known as the ***Robertson-Sparck Jones*** Equation

20

Deakin University CRICOS Provider Code: 00113B

20

Ranking Formula

- The previous equation cannot be computed without estimates of r_i and R
- One possibility is to assume $R = r_i = 0$, as a way to boostrap the ranking equation, which leads to:

$$Sim(q,d) \approx \sum_{t_i = q_i = 1} \log \left(\frac{N - n_i + 0.5}{n_i + 0.5} \right)$$

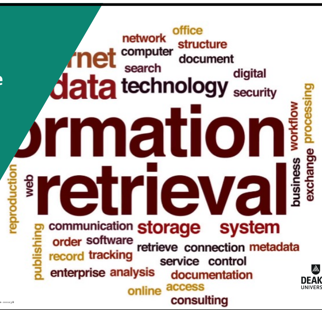
- This equation provides an *idf-like* ranking computation
- In the absence of relevance information, this is the equation for ranking in the probabilistic model

21

Deakin University CRICOS Provider Code: 00113B

21

BIM Ranking Example



22

Deakin University CRICOS Provider Code: 00113B

22

Ranking Example

- Document ranks computed by the previous probabilistic ranking equation for the query ***"to do"***

$$Sim(q,d) \approx \sum_{t_i = q_i = 1} \log \left(\frac{N - n_i + 0.5}{n_i + 0.5} \right)$$

To do is to be.
To be is to do.
d_1

I think therefore I am.
Do be do be do.
d_3

To be or not to be.
I am what I am.
d_2

Do do do, da da da.
Let it be, let it be.
d_4

doc	rank computation	rank
d_1	$\log \frac{4-2+0.5}{2+0.5} + \log \frac{4-3+0.5}{3+0.5}$	- 1.222
d_2	$\log \frac{4-2+0.5}{2+0.5}$	0
d_3	$\log \frac{4-3+0.5}{3+0.5}$	- 1.222
d_4	$\log \frac{4-3+0.5}{3+0.5}$	- 1.222

23

Deakin University CRICOS Provider Code: 00113B

23

Ranking Example

- The ranking computation led to negative weights because of the term **"do"**
- Actually, the probabilistic ranking equation produces negative terms whenever $n_i > N/2$
- One possible artifact to contain the effect of negative weights is to change the previous equation to:

$$Sim(q,d) \approx \sum_{t_i = q_i = 1} \log \left(\frac{N + 0.5}{n_i + 0.5} \right)$$

- By doing so, a term that occurs in all documents ($n_i = N$) produces a weight equal to zero

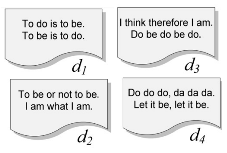
24

Deakin University CRICOS Provider Code: 00113B

24

Ranking Example

- Using this latest formulation, we redo the ranking computation for our example collection for the query "***to do***" and obtain



$$Sim(q,d) \approx \sum_{t_i = q_i = 1} \log\left(\frac{N + 0.5}{n_i + 0.5}\right)$$

doc	rank computation	rank
d_1	$\log \frac{4+0.5}{2+0.5} + \log \frac{4+0.5}{3+0.5}$	1.210
d_2	$\log \frac{4+0.5}{2+0.5}$	0.847
d_3	$\log \frac{4+0.5}{3+0.5}$	0.362
d_4	$\log \frac{4+0.5}{3+0.5}$	0.362

25

Deakin University CRICOS Provider Code: 00113B

25

Improving the BIM ranking

26

26

Estimaging r_i and R

- Our examples above considered that $r_i = R = 0$
- An alternative is to estimate r_i and R performing an initial search:
  - select the top 10-20 ranked documents
  - inspect them to gather new estimates for r_i and R
  - remove the 10-20 documents used from the collection
  - rerun the query with the estimates obtained for r_i and R
- Unfortunately, procedures such as these require human intervention to initially select the relevant documents

27

Deakin University CRICOS Provider Code: 00113B

27

Improving the Initial Ranking

- Consider the equation

$$Sim(q,d) \approx \sum_{t_i = q_i = 1} \log \frac{p_i(1-u_i)}{u_i(1-p_i)}$$

- How obtain the probabilities p_i and u_i?
- Estimates based on assumptions:
  - $p_i = 0.5$
  - $u_i = \frac{n_i}{N}$ where n_i is the number of docs that contain t_i
  - Use this initial guess to retrieve an initial ranking
  - Improve upon this initial ranking

p_i : the probability of a term appearing in a document relevant to the query

u_i : the probability of a term appearing in a non-relevant document

28

Deakin University CRICOS Provider Code: 00113B

28

Improving the Initial Ranking

- Substituting p_i and u_i into the previous Equation, we obtain:

$$Sim(q,d) \approx \sum_{t_i = q_i = 1} \log\left(\frac{N - n_i}{n_i}\right)$$

- That is the equation used when no relevance information is provided, without the 0.5 correction factor
- Given this initial guess, we can provide an initial probabilistic ranking

29

Deakin University CRICOS Provider Code: 00113B

29

Improving the Initial Ranking

- We can attempt to improve this initial ranking as follows
- Let
  - D: set of docs initially retrieved
  - D_i: subset of docs retrieved that contain t_i
- Re-evaluate estimates:
  - $p_i = \frac{D_i}{D}$
  - $u_i = \frac{n_i - D_i}{N - D}$
- This process can then be repeated recursively

- To avoid $D = 0$ and $D_i = 0$:
  - $p_i = \frac{D_i + 0.5}{D + 1}$
  - $u_i = \frac{n_i - D_i + 0.5}{N - D + 1}$
- Also,
  - $p_i = \frac{D_i + \frac{n_i}{N}}{D + 1}$
  - $u_i = \frac{n_i - D_i + \frac{n_i}{N}}{N - D + 1}$

30

Deakin University CRICOS Provider Code: 00113B

30

Pluses and Minuses

- Advantages:
  - Docs ranked in decreasing order of probability of relevance
- Disadvantages:
  - need to guess initial estimates for p_i
  - method does not take into account tf factors
  - the lack of document length normalization

31

Comparison of Classic Models

- Boolean model does not provide for partial matches and is considered to be the weakest classic model
- There is some controversy as to whether the probabilistic model outperforms the vector model
- Croft suggested that the probabilistic model provides a better retrieval performance
- However, Salton *et al* showed that the vector model outperforms it with general collections
- This also seems to be the dominant thought among researchers and practitioners of IR

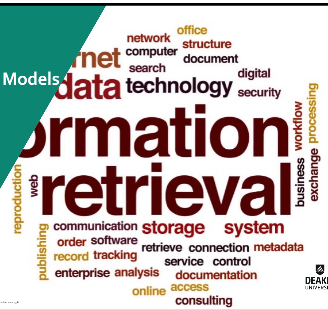
32

Comparison of Classic Models

- Boolean model does not provide for partial matches and is considered to be the weakest classic model
- There is some controversy as to whether the probabilistic model outperforms the vector model
- Croft suggested that the probabilistic model provides a better retrieval performance
- However, Salton *et al* showed that the vector model outperforms it with general collections
- This also seems to be the dominant thought among researchers and practitioners of IR

33

The BM (Best Match) Models



34

The BM (Best Match) Models

- A good term weighting is based on three principles
  - inverse document frequency
  - term frequency
  - document length normalization
- The classic probabilistic model covers only the first of these principles
- This reasoning led to a series of experiments, which led to new formulas

35

BM1, BM11 and BM15 Formulas

- At first, the Okapi system used the Equation below as ranking formula

$$Sim(q,d) \approx \sum_{t_i=q_i=1} \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

which is the equation used in the probabilistic model, when no relevance information is provided

- It was referred to as the BM1 formula (***Best Match 1***)

36

BM1, BM11 and BM15 Formulas

- The first idea for improving the ranking was to introduce a term-frequency factor $\mathcal{F}_{t,d}$ in the BM1 formula
- This factor, after some changes, evolved to become

$$\mathcal{F}_{t,d} = S_1 \times \frac{tf_{t,d}}{K_1 + tf_{t,d}}$$

Where
  - $tf_{t,d}$ is the frequency of term t within document d
  - K_1 is a constant setup experimentally for each collection
  - S_1 is a scaling constant, normally set to $S_1 = (K_1+1)$
- If $K_1 = 0$ this whole factor becomes equal to and bears no effect in the ranking

37

37

BM1, BM11 and BM15 Formulas

- The next step was to modify the $\mathcal{F}_{t,d}$ factor by adding document length normalization to it, as follows:

$$\grave{\mathcal{F}}_{t,d} = S_1 \times \frac{tf_{t,d}}{\frac{K_1 \times len(d)}{avg\_doclen} + tf_{t,d}}$$

Where
  - $len(d)$ is the length of document d (computed, for instance, as the number of terms in the document)
  - avg_doclen is the average document length for the collection

38

38

BM1, BM11 and BM15 Formulas

- Next, a correction factor $\mathcal{G}_q$ dependent on the document and query lengths was added

$$\mathcal{G}_q = K_2 \times len(q) \times \frac{avg\_doclen - len(d)}{avg\_doclen + len(d)}$$

Where
  - $len(q)$ is the query length (number of terms in the query)
  - K_2 is a constant

39

39

BM1, BM11 and BM15 Formulas

- A third additional factor, aimed at taking into account term frequencies within queries, was defined as

$$\mathcal{F}_{t,q} = S_3 \times \frac{tf_{t,q}}{K_3 + tf_{t,q}}$$

Where
  - $tf_{t,d}$ is the frequency of term t within query q
  - K_3 is a constant
  - S_3 is a scaling constant related to K_3, normally set to $S_3 = (K_3 + 1)$

40

40

BM1, BM11 and BM15 Formulas

- Introduction of these three factors led to various BM (Best Matching) formulas, as follows:

$$Sim_{BM1}(q,d) \approx \sum_{t_i = q_i = 1} \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

$$Sim_{BM15}(q,d) \approx \mathcal{G}_q + \sum_{t_i = q_i = 1} \mathcal{F}_{t,d} \times \mathcal{F}_{t,q} \times \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

$$Sim_{BM11}(q,d) \approx \mathcal{G}_q + \sum_{t_i = q_i = 1} \grave{\mathcal{F}}_{t,d} \times \mathcal{F}_{t,q} \times \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

41

41

BM1, BM11 and BM15 Formulas

- Experiments using TREC data have shown that BM11 outperforms BM15
- Further, empirical considerations can be used to simplify the previous equations, as follows:
  - Empirical evidence suggests that a best value of K_2 is 0, which eliminates the $\mathcal{G}_q$ factor from these equations
  - Further, good estimates for the scaling constants S_1 and S_3 are $K_1 + 1$ and $K_3 + 1$, respectively
  - Empirical evidence also suggests that making K_3 very large is better. As a result, the $\mathcal{F}_{t,q}$ factor is reduced simply to $tf_{t,q}$
  - For short queries, we can assume that $tf_{t,q}$ is 1 for all terms

42

42

BM1, BM11 and BM15 Formulas

- These considerations lead to simpler equations as follows:

$$Sim_{BM1}(q,d) \approx \sum_{t_i = q_i = 1} \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

$$Sim_{BM15}(q,d) \approx \sum_{t_i = q_i = 1} \frac{(K_1+1)tf_{t,d}}{K_1 + tf_{t,d}} \times \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

$$Sim_{BM11}(q,d) \approx \sum_{t_i = q_i = 1} \frac{(K_1+1)tf_{t,d}}{\frac{K_1 len(d)}{avg\_doclen} + tf_{t,d}} \times \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

43

The BM25 Model

44

BM25 Ranking Formula

- BM25: combination of the BM11 and BM15
- The motivation was to combine the BM11 and BM25 term frequency factors as follows:

$$\mathcal{B}_{t,d} = \frac{(K_1+1)tf_{t,q}}{K_1\left[(1-b) + b\frac{len(d)}{avg\_doclen}\right] + tf_{t,q}}$$

Where b is a is a constant with values in the interval [0,1]

- If $b = 0$, it reduces to the BM15 term frequency factor
- If $b = 1$, it reduces to the BM11 term frequency factor
- For values of b between 0 and 1, the equation provides a combination of BM11 with BM15

45

BM25 Ranking Formula

- The ranking equation for the BM25 model can then be written as:

$$Sim_{BM25}(q,d) \approx \sum_{t_i = q_i = 1} \mathcal{B}_{t,d} \times \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

where K_1 and b are empirical constants

- $K_1 = 1$ works well with real collections
- b should be kept closer to 1 to emphasize the document length normalization effect present in the BM11 formula
- For instance, $b = 0.75$ is a reasonable assumption
- Constants values can be fine tunned for particular collections through proper experimentation

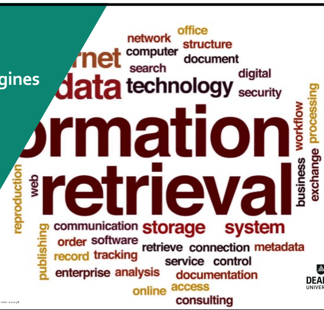
46

BM25 Ranking Formula

- Unlike the probabilistic model, the BM25 formula can be computed without relevance information
- There is consensus that BM25 outperforms the classic vector model for general collections
- Thus, it has been used as a baseline for evaluating new ranking functions, in substitution to the classic vector model

47

Evaluating search engines



48

Measures for a search engine

- How fast does it index
  - Number of documents/hour
  - (Average document size)
- How fast does it search
  - Latency as a function of index size
- Expressiveness of query language
  - Ability to express complex information needs
  - Speed on complex queries
- Uncluttered UI
- Is it free?

49

Measures for a search engine

- All of the preceding criteria are *measurable*: we can quantify speed/size
  - we can make expressiveness precise
- The key measure: user happiness
  - What is this?
  - Speed of response/size of index are factors
  - But blindingly fast, useless answers won't make a user happy
- Need a way of quantifying user happiness

50

Measuring user happiness

- Issue: who is the user we are trying to make happy?
  - Depends on the setting
- Web engine:
  - User finds what s/he wants and returns to the engine
    - Can measure rate of return users
  - User completes task – search as a means, not end
  - See Russell http://dmrussell.googlepages.com/JCDL-talk-June-2007-short.pdf
- eCommerce site: user finds what s/he wants and buys
  - Is it the end-user, or the eCommerce site, whose happiness we measure?
  - Measure time to purchase, or fraction of searchers who become buyers?

51

Measuring user happiness

- Enterprise (company/govt/academic): Care about "user productivity"
  - How much time do my users save when looking for information?
  - Many other criteria having to do with breadth of access, secure access, etc.

52

Happiness: elusive to measure

- Most common proxy: *relevance* of search results
- But how do you measure relevance?
- We will detail a methodology here, then examine its issues
- Relevance measurement requires 3 elements:
  1. A benchmark document collection
  2. A benchmark suite of queries
  3. A usually binary assessment of either Relevant or Nonrelevant for each query and each document
     - Some work on more-than-binary, but not the standard

53

Evaluating an IR system

- Note: the **information need** is translated into a **query**
- Relevance is assessed relative to the **information need** *not* the **query**
- E.g., Information need: *I'm looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine.*
- Query: ***wine red white heart attack effective***
- Evaluate whether the doc addresses the information need, not whether it has these words

54

Standard relevance benchmarks

- TREC - National Institute of Standards and Technology (NIST) has run a large IR test bed for many years
- Reuters and other benchmark doc collections used
- "Retrieval tasks" specified
  - sometimes as queries
- Human experts mark, for each query and for each doc, Relevant or Nonrelevant
  - or at least for subset of docs that some system returned for that query

55

Boolean Evaluating Metrics

56

Unranked retrieval evaluation: Precision and Recall

- **Precision**: fraction of retrieved docs that are relevant = P(relevant|retrieved)
- **Recall**: fraction of relevant docs that are retrieved = P(retrieved|relevant)

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

- Precision P = tp/(tp + fp)
- Recall R = tp/(tp + fn)

57

Should we instead use the accuracy measure for evaluation?

- Given a query, an engine classifies each doc as "Relevant" or "Nonrelevant"
- The **accuracy** of an engine: the fraction of these classifications that are correct:

  (tp + tn) / (tp + fp + fn + tn)
- **Accuracy** is a commonly used evaluation measure in machine learning classification work
- Why is this not a very useful evaluation measure in IR?

58

Why not just use accuracy?

- How to build a 99.9999% accurate search engine on a low budget....

snoogle.com

Search for:

0 matching results found.

- People doing information retrieval *want to find something* and have a certain tolerance for junk.

59

Precision/Recall

- You can get high recall (but low precision) by retrieving all docs for all queries!
- Recall is a non-decreasing function of the number of docs retrieved
- In a good system, precision decreases as either the number of docs retrieved or recall increases
  - This is not a theorem, but a result with strong empirical confirmation

60

Difficulties in using precision/recall

- Should average over large document collection/query ensembles
- Need human relevance assessments
  - People aren't reliable assessors
- Assessments have to be binary
  - Nuanced assessments?
- Heavily skewed by collection/authorship
  - Results may not translate from one domain to another

61

A combined measure: *F*

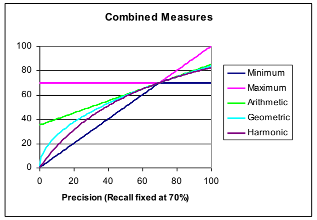
- Combined measure that assesses precision/recall tradeoff is **F measure** (weighted harmonic mean):

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

- People usually use balanced F_1 measure
  - i.e., with $\beta = 1$ or $\alpha = \frac{1}{2}$
- Harmonic mean is a conservative average
  - See CJ van Rijsbergen, *Information Retrieval*

62

F_1 and other averages



63

Ranked evaluation metrics

64

Evaluating ranked results

- Up until now we've been considering metrics for boolean (set-based) retrieval
  - Precision, Recall, F_1
- But users don't really care about all results
- Users care about getting results near top of ranking…

65

Metrics: Ranking

- Ranking results matters for human consumption of data

1. **Precision @ k (P@k)**
   - Percent of relevant results (out of top k)
2. **Average Precision (AP or AveP)**
   - Weights higher ranks more
   - More on the exact definition shortly…

Precision at k: P@10 does not distinguish between the two results

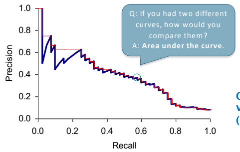
Average Precision: prefers Result 2 to Result 1

Rank	Result 1 ($P@k=0.5$, $AP=0.68$)	Result 2 ($P@k=0.5$, $AP=1.00$)
1	✓	✓
2		✓
3	✓	✓
4		✓
5	✓	✓
6		
7	✓	
8		
9	✓	
10		

66

Evaluating ranked results

- Sometimes we don't want to fix top "k"
  - The system can return any number of results
  - We can evaluate performance for a range of k by looking at the **precision-recall curve**



One way to generate is to vary the length k of a (ranked) results list.

67

67

Evaluation

- Graphs are good, but people want summary measures!
  - P@k good for most of web search… why? what k?
  - But P@k averages badly
    - If only 10 relevant docs, max P@100 is 0.1
    - Also has an arbitrary parameter of *k*
    - Sometimes **R-Prec** is better
      - R-Prec definition: P@k with k=#relevant docs (for query)
      - max R-Prec is 1.0, why?
  - *But P@k and R-Prec still use a fixed k. Does any ranking metric approximate area under precision-recall curve?*
    - Well yes, average precision does just that…

68

68

Definition of (Mean) Average Precision

- **Average Precision (AveP or AP) and Mean AP (MAP)**

$$\text{MAP} = \frac{\sum_{q=1}^{Q} \text{AveP(q)}}{Q} \qquad \text{AveP} = \frac{\sum_{k=1}^{n} (P(k) \times \text{rel}(k))}{\text{number of relevant documents}}$$

  - AP = higher ranked docs are counted more often
    - Unlike P@k, ordering matters!
  - AP ≈ area under precision-recall curve when n→#all docs!
    - Good discussion in IIR book and on Wikipedia https://en.wikipedia.org/wiki/Information_retrieval#Performance_and_correctness_measures
  - Mean AP (MAP) = mean over queries
    - Note: this is **macro-averaging**: queries weighted equally
    - Empirically correlates with human evaluation of retrieval systems

69

69

Variance

- For a test collection, it is usual that a system does crummily on some information needs (e.g., MAP = 0.1) and excellently on others (e.g., MAP = 0.7)
- Indeed, it is usually the case that the variance in performance of the same system across queries is much greater than the variance of different systems on the same query.
- **There are easy information needs and hard ones!**

70

70

Test collection for IR evaluation

71

71

Test Collections

TABLE 4.3 Common Test Corpora

Collection	*NDocs*	*NQrys*	*Size (MB)*	*Term/Doc*	*Q-D RelAss*
ADI	82	35			
AIT	2109	14	2	400	>10,000
CACM	3204	64	2	24.5	
CISI	1460	112	2	46.5	
Cranfield	1400	225	2	53.1	
LISA	5872	35	3		
Medline	1033	30	1		
NPL	11,429	93	3		
OSHMED	34,8566	106	400	250	16,140
Reuters	21,578	672	28	131	
TREC	740,000	200	2000	89-3543	» 100,000

72

72

From document collections to test collections

- Still need
  - Test queries
  - Relevance assessments
- Test queries
  - Must be germane to docs available
  - Best designed by domain experts
  - Random query terms generally not a good idea
- Relevance assessments
  - Human judges, time-consuming
  - Are human panels perfect?

73

TREC

- TREC Ad Hoc task from first 8 TRECs is standard IR task
  - 50 detailed information needs a year
  - Human evaluation of pooled results returned
  - More recently other related things: Web track, HARD
- A TREC query (TREC 5)

<top>
<num> Number: 225
<desc> Description:
What is the main function of the Federal Emergency Management Agency (FEMA) and the funding level provided to meet emergencies? Also, what resources are available to FEMA such as people, equipment, facilities?
</top>

74

Qrels example

225 0 EP-1003076 2
225 0 EP-0926545 2
225 0 EP-0952483 2
226 0 EP-0852279 2
226 0 EP-1254842 1
227 0 EP-0999033 2
227 0 EP-0855703 2
228 0 EP-0459510 2
228 0 EP-0192015 2
228 0 EP-1348868 2
229 0 EP-0157442 2
229 0 EP-1249171 2
229 0 EP-0554468 1
230 0 EP-0926230 2
230 0 EP-0931578 2
230 0 EP-1008559 2

75

Can we avoid human judgment?

- No
- Makes experimental work hard
  - Especially on a large scale
- In some very specific settings, can use proxies
  - E.g.: for approximate vector space retrieval, we can compare the cosine distance closeness of the closest docs to those found by an approximate retrieval algorithm
- But once we have test collections, we can reuse them (so long as we don't overtrain too badly)

76

Evaluation at large search engines

- Search engines have test collections of queries and hand-ranked results
- Recall is difficult to measure on the web
- Search engines often use precision at top k, e.g., k = 10
- . . . or measures that reward you more for getting rank 1 right than for getting rank 10 right.
  - NDCG (Normalized Cumulative Discounted Gain)
- Search engines also use non-relevance-based measures.
  - Clickthrough on first result
    - Not very reliable if you look at a single clickthrough … but pretty reliable in the aggregate.
  - Studies of user behavior in the lab
  - A/B testing

77

A/B testing

- Purpose: Test a single innovation
- Prerequisite: You have a large search engine up and running.
- Have most users use old system
- Divert a small proportion of traffic (e.g., 1%) to the new system that includes the innovation
- Evaluate with an "automatic" measure like clickthrough on first result
- Now we can directly see if the innovation does improve user happiness.
- Probably the evaluation methodology that large search engines trust most
- In principle less powerful than doing a multivariate regression analysis, but easier to understand

78

Results presentation

79

Result Summaries

- Having ranked the documents matching a query, we wish to present a results list
- Most commonly, a list of the document titles plus a short summary, aka "10 blue links"

80

Summaries

- The title is often automatically extracted from document metadata. What about the summaries?
  - This description is crucial.
  - User can identify good/relevant hits based on description.
- Two basic kinds:
  - Static
  - Dynamic
- A **static summary** of a document is always the same, regardless of the query that hit the doc
- A **dynamic summary** is a *query-dependent* attempt to explain why the document was retrieved for the query at hand

81

Static summaries

- In typical systems, the static summary is a subset of the document
- Simplest heuristic: the first 50 (or so – this can be varied) words of the document
  - Summary cached at indexing time
- More sophisticated: extract from each document a set of "key" sentences
  - Simple NLP heuristics to score each sentence
  - Summary is made up of top-scoring sentences.
- Most sophisticated: NLP used to synthesize a summary
  - Seldom used in IR; cf. text summarization work

82

Dynamic summaries

- Present one or more "windows" within the document that contain several of the query terms
  - "KWIC" snippets: Keyword in Context presentation

83

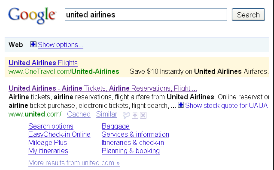
Techniques for dynamic summaries

- Find small windows in doc that contain query terms
  - Requires fast window lookup in a document cache
- Score each window wrt query
  - Use various features such as window width, position in document, etc.
  - Combine features through a scoring function
- Challenges in evaluation: judging summaries
  - Easier to do pairwise comparisons rather than binary relevance assessments

84

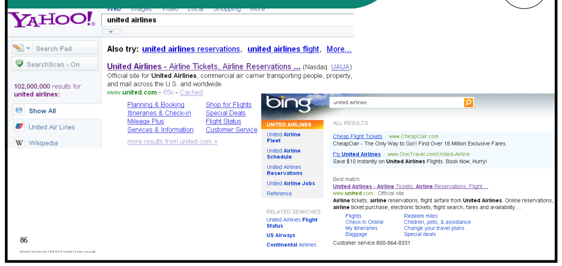
Quicklinks

- For a *navigational query* such as ***united airlines*** user's need likely satisfied on www.united.com
- Quicklinks provide navigational cues on that home page



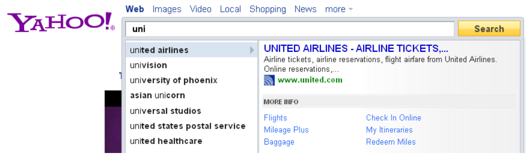
85

Quicklinks



86

Alternative results presentations?



87