Retrieval models III & Evaluation

IN4325 – Information Retrieval

General feedback on assignment 1

- Some groups got on well
- Some struggled with AWS
  - Others did not mention the AWS experiments at all in their report **
- Some did not report the required numbers (results on small test corpus required, no new AWS experiments)
  - Some reported all numbers, but did not specify which corpus they were referring to (small/large would be a good distinction)
  - Some only reported the numbers on the large corpus
- Some did not include the source code (*.class only)

Assignment II

small: $0.095 / hour
xlarge: $0.76 / hour

- Assignment I cost $1400 (we have $3500 in total)
  - xlarge instances are expensive
  - Now everything on max. 8 instances of **m1.small**
    - **~1 hour**
- Hints for assignment II
  - Can be done in one pass
  - Queries can be "hardcoded" into the Mapper / more principled: DistributedCache
  - Use a combiner (very important when emitting a lot of (key,value) pairs)

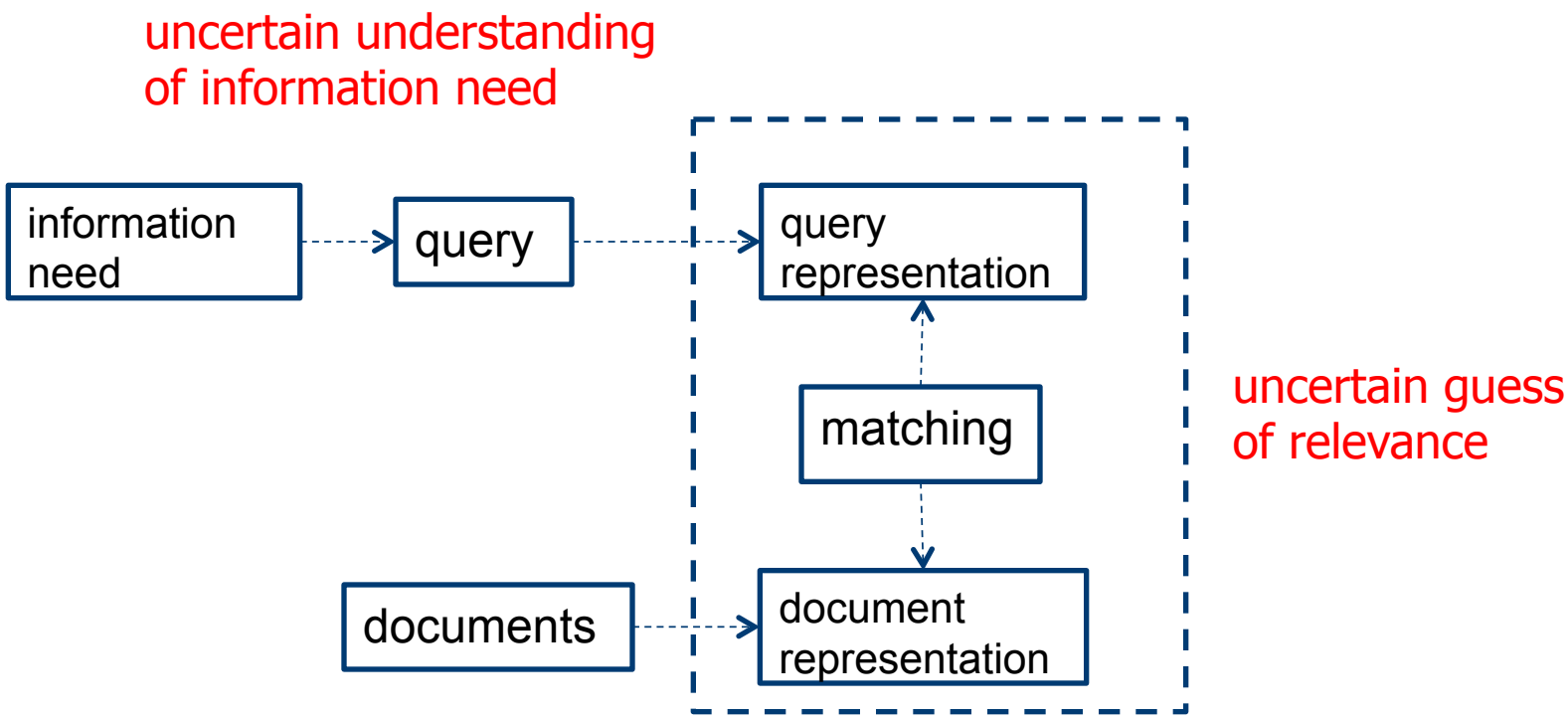
Assignment II

- Hints for assignment II cont.
  - Performance may be very low (depends on normalizing steps taken)
  - Test queries are available now on the website:
    - http://www.st.ewi.tudelft.nl/~hauff/IN4325/
  - Trec_eval can be used to calculate MAP: http://trec.nist.gov/trec_eval/
- Example index will be made available on S3 today
  - Check http://www.st.ewi.tudelft.nl/~hauff/IN4325/

Today

- Language modeling
  - More about smoothing
  - Document priors
- Binary independence model
- BM25
- More about evaluations

Probabilistic models in IR



Language modeling

Language modeling

Last lecture in 1 slide

$$P(D \mid Q) = \frac{P(Q \mid D) \times P(D)}{P(Q)}$$

$$P(Q \mid D) = \prod_{q_i \in Q} P(q_i \mid D)$$

central equations in language modeling

- Jelineck-Mercer smoothing

$$P_\lambda(w \mid D) = (1-\lambda) P_{ml}(w \mid D) + \lambda P(w \mid \mathbb{C}), \quad \lambda \in (0,1)$$

can be reversed

parameters control amount of smoothing

- Dirichlet smoothing

$$P_\mu(w \mid D) = \frac{c(w;D) + \mu P(w \mid \mathbb{C})}{\sum_w c(w;D) + \mu}, \; usually \; \mu > 100$$

➔The longer the document, the less smoothing is applied

Smoothing: an experimental study [5]

Reminder: ad hoc TREC topic

TREC 2001 Web adhoc topic

<top>
<num> Number: 503

<title> Vikings in Scotland?

<desc> Description: What hard evidence proves that the Vikings visited or lived in Scotland?

<narr> Narrative: A document that merely states that the Vikings visited or lived in Scotland is not relevant. A relevant document must mention the source of the information, such as relics, sagas, runes or other records from those times.
</top>

title query (short query)

descr. query (long query) rarely used.

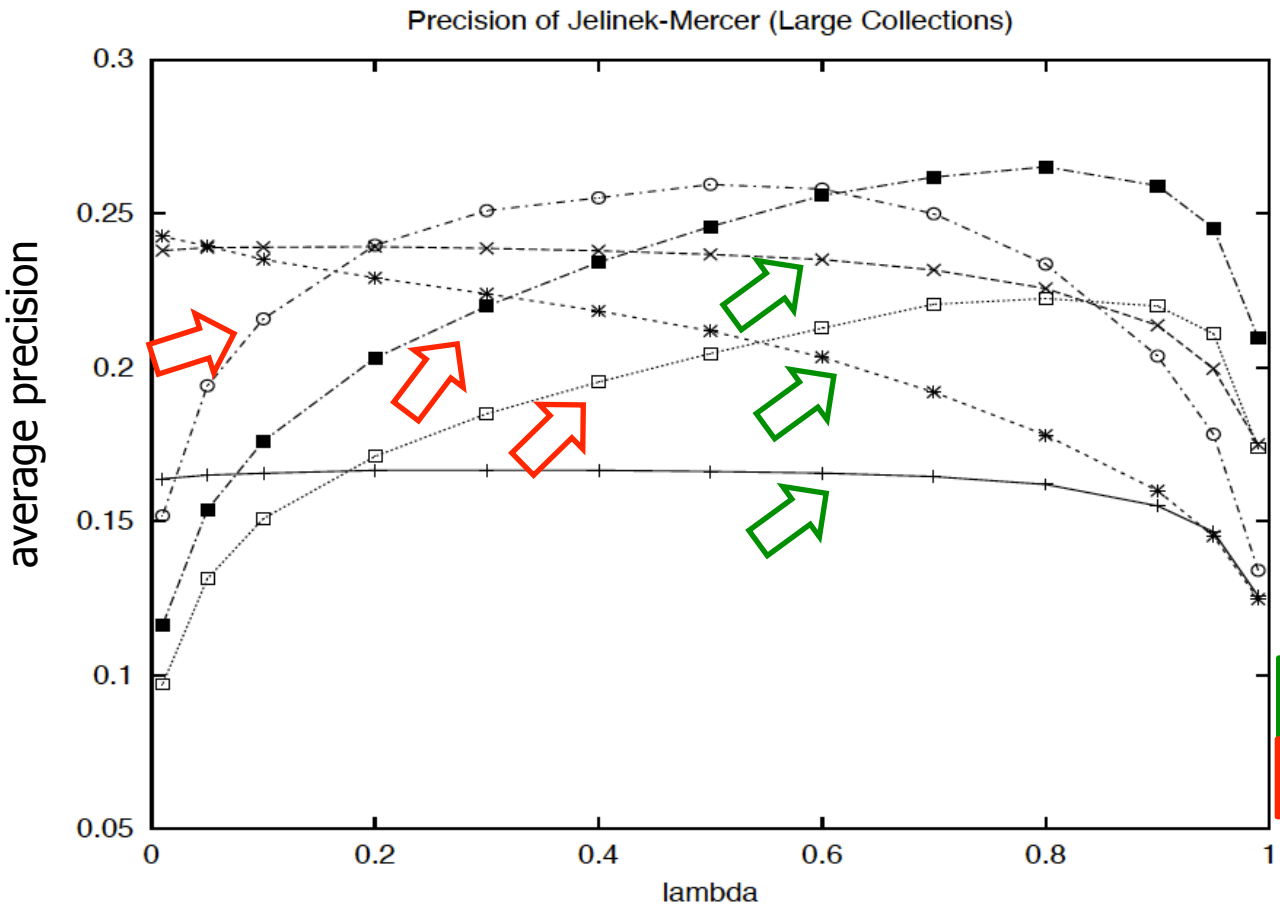
Information need description for assessors.

5 TREC corpora

Language modeling

Jelineck-Mercer smoothing [5]

- λ more sensitive for long queries
- Title queries: good λ=0.1
- Long queries: good λ=0.7



long queries need more smoothing

title queries

T+D+N queries

minimum smoothing

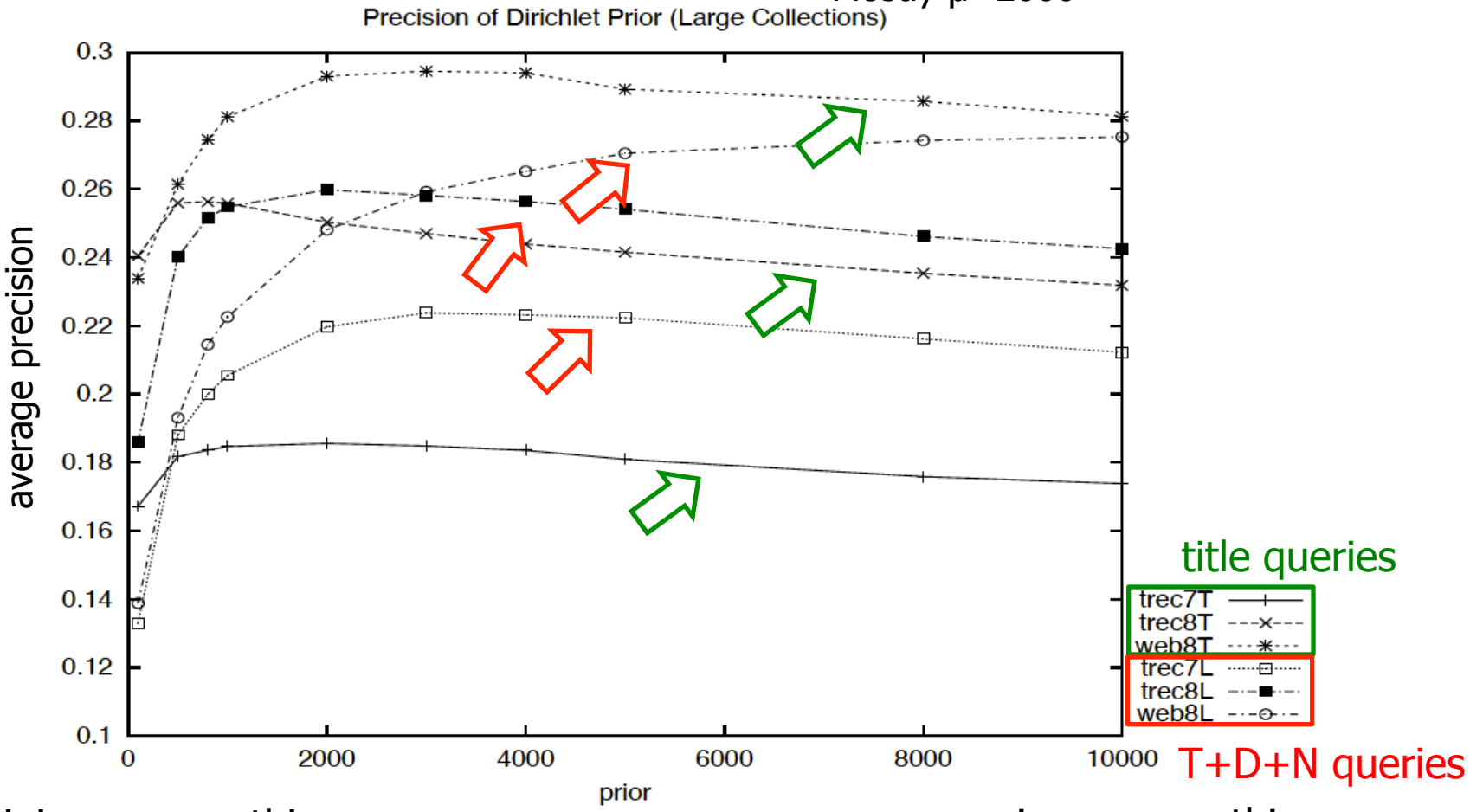
maximum smoothing

Source: [5]

Language modeling

Dirichlet smoothing [5]

- μ more sensitive for long queries
- Optimal value similar for T/L queries
- Mostly μ=2000



minimum smoothing

maximum smoothing

Source: [5]

Language modeling

What about several sources of evidence? [6]

- On the Web (or elsewhere), several sources of information to estimate content models
  - E.g. the content of the Web page + the anchor texts of all hyperlinks pointing to the document
  - N potentially very different representations of the same document

Why Publishers Don't See Google As A Friend
thenoisychannel.com/.../why-publishers-dont-see-google-as-a-friend/
12 Apr 2009 – Where the real money ends up is at the one point in the system where traffic is concentrated: the Google **search engine. Google's** overriding ...
Daniel Tunkelang shared this

$$P(D \mid Q) \propto P(D) \prod_{i=1}^{n} \Big((1-\lambda-\mu) P(q_i \mid \mathbb{C}) + \lambda P_{content}(q_i \mid D) + \mu P_{anchor}(q_i \mid D) \Big)$$

Language modeling

Cluster-based retrieval [7]

- Smooth documents with a mixture of the document's topical cluster and the corpus

$$P(w \mid D) = (1-\lambda)P_{ml}(w \mid D) + \lambda[(1-\beta)P_{ml}(w \mid Cluster) + \beta P(w \mid \mathbb{C})],$$
$$with\ \lambda, \beta \in (0,1)$$

1. Cluster model smoothed with corpus model
2. Document model is smoothed with smoothed cluster model

- Retrieval effectiveness of cluster-based smoothing has been shown to improve upon standard LM
- Issue: parameter estimation of the clustering approaches

Language modeling

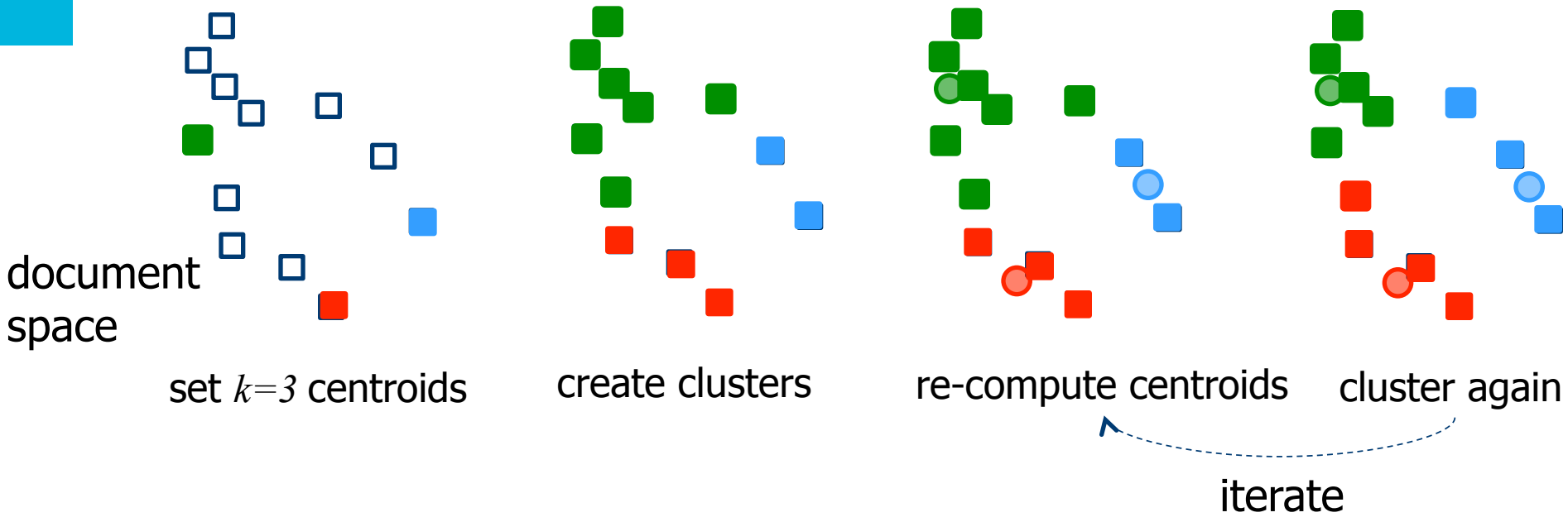
Cluster-based retrieval [7]

- The corpus documents need to be clustered
- 2 step process
  - Determine a suitable pairwise measure of document similarity (or distance)
  - Group documents based on their similarity (distance)
- Popular similarity measures: **cosine similarity**, Dice & Jaccard coefficients, overlap coefficient, Kullback-Leibler divergence
- Grouping: partitioning (e.g. **k-means**), hierarchical agglomerative clustering (e.g. **single linkage**)

K-means clustering

A short detour

Chapters 16 & 17 in the course book!!

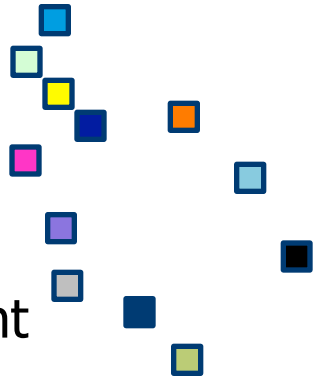


Goal: partition the N elements into k disjoint sets S_j with minimized sum of squares:

$$\sum_{j=1}^{k} \sum_{n \in S_j} \left| x_n - \mu_j \right|^2$$

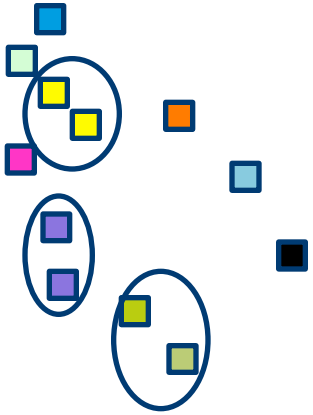
Hierarchical agglomerative clustering

A short detour

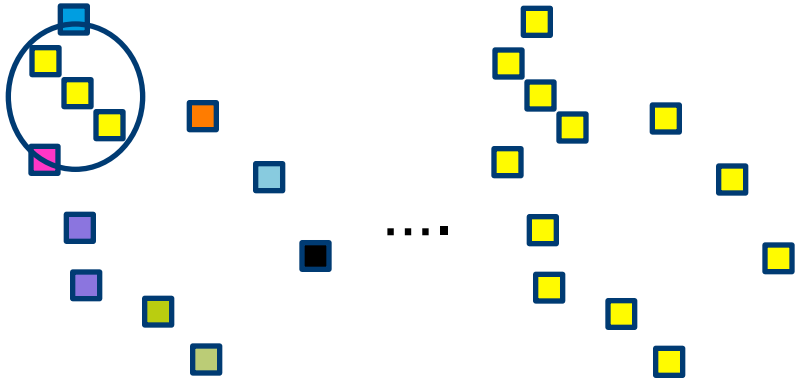


document space

every document is in its own cluster



merge the two most similar clusters iteratively



Single linkage: cluster similarity based On two most similar documents

eventually

cutoff point needs to be determined (when to stop merging)

Language modeling: P(D)

The document prior

- So far: $P(D)$ is assumed to be uniform
  - Each document is equally likely to be drawn for a query
- What can influence the probability of a document being relevant to an unseen query?
  - Document length
  - Document quality (PageRank, HITS, etc.)
  - Document source (Wikipedia pages receive a high prior)
  - Recency
  - Language
  - ….

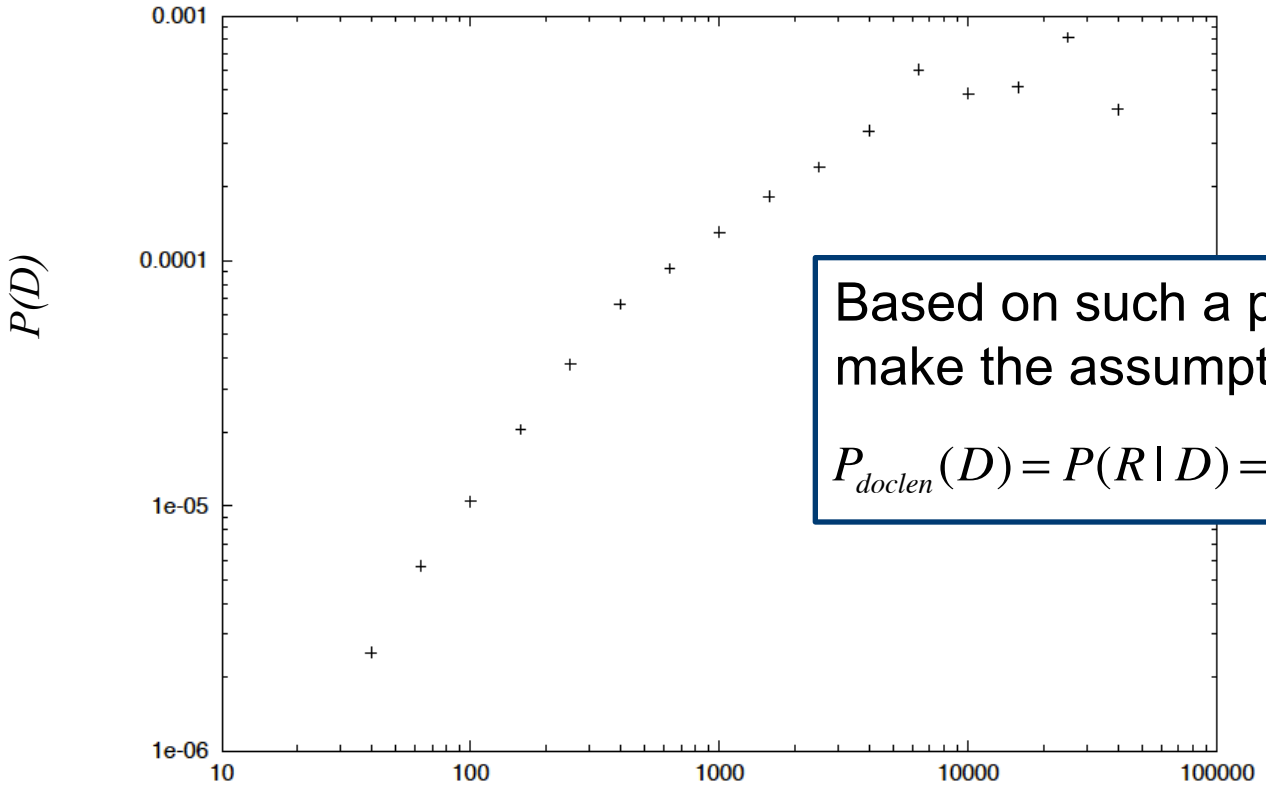
A case study in language modeling: P(D)

Kraaij et al. [6]

- Another TREC task: Entry page search
  - Find an entry page (homepage) of an organisation
- Ad hoc retrieval systems purely based on content perform poorly
- Priors (or other model components) can be
  - Estimated from training data
  - Defined based on some general modelling assumptions

Language modeling: P(D)

Kraaij et al. [6]



Based on such a plot, one can make the assumption:

$$P_{doclen}(D) = P(R \mid D) = C \times doclen(D)$$

Figure 1: Prior probability of relevance given document length on the Ad Hoc task

Language modeling: P(D)

Kraaij et al. [6]

- What about page priors? Which ones might be successful?
  - Page length?
  - Number of web pages pointing to the target page?
  - URL form?

Language modeling: P(D)

Kraaij et al. [6]

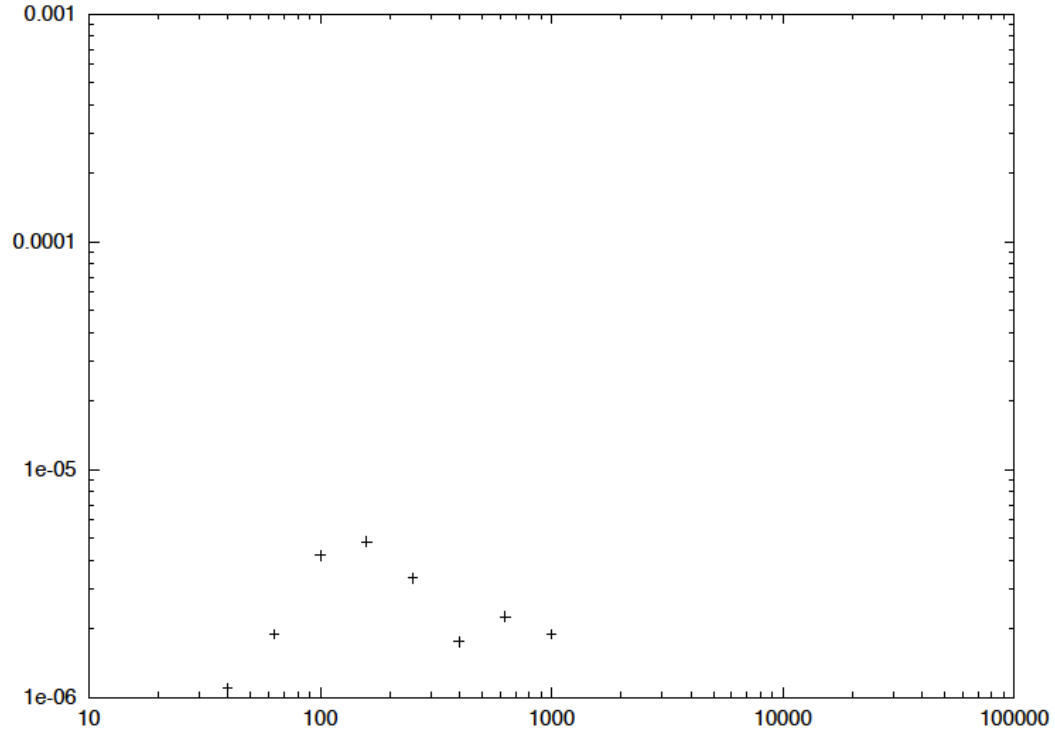


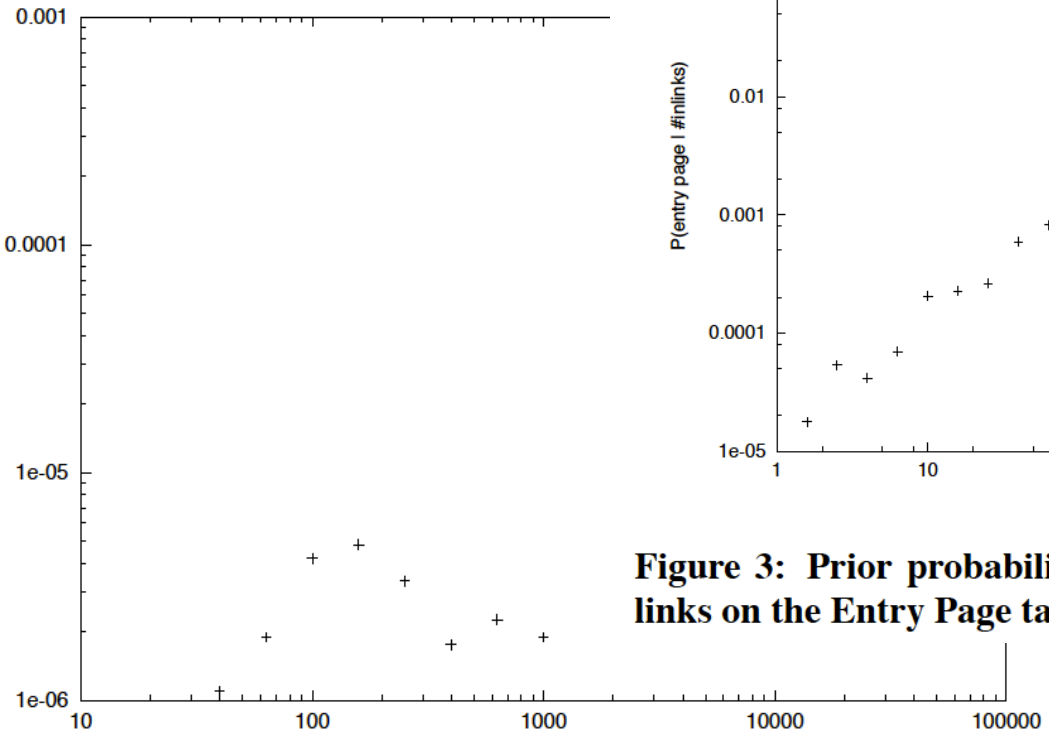
Figure 2: Prior probability of relevance given document length on the Entry Page task ($P(entry\ page|doclen)$)

successful?

e?

Language modeling: P(D)

Kraaij et al. [6]



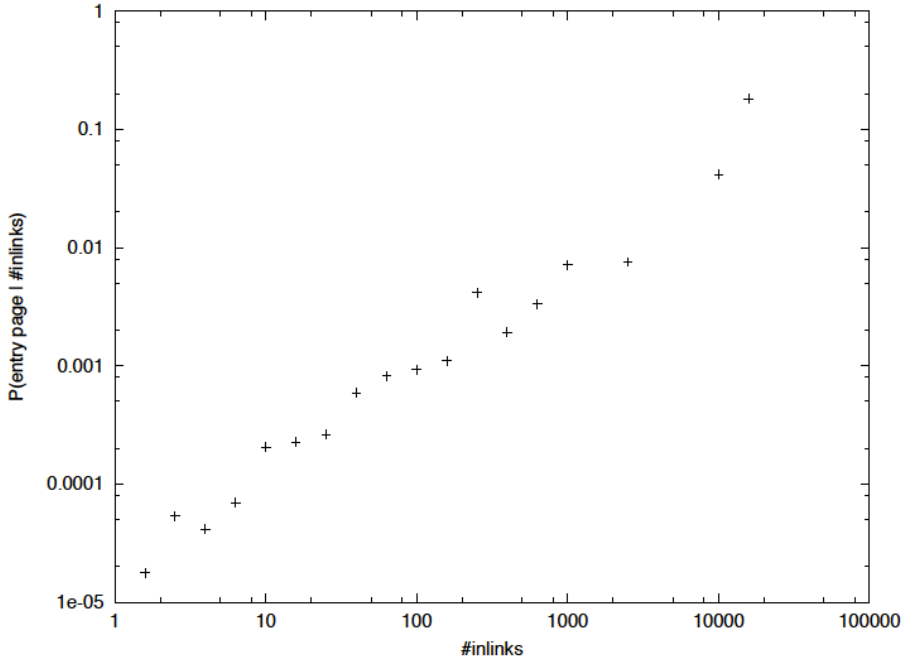


Figure 3: Prior probability of relevance given number of inlinks on the Entry Page task ($P(entry\ page|inlinkCount)$)

Figure 2: Prior probability of relevance given document length on the Entry Page task ($P(entry\ page|doclen)$)

Language modeling: P(D)

Kraaij et al. [6]

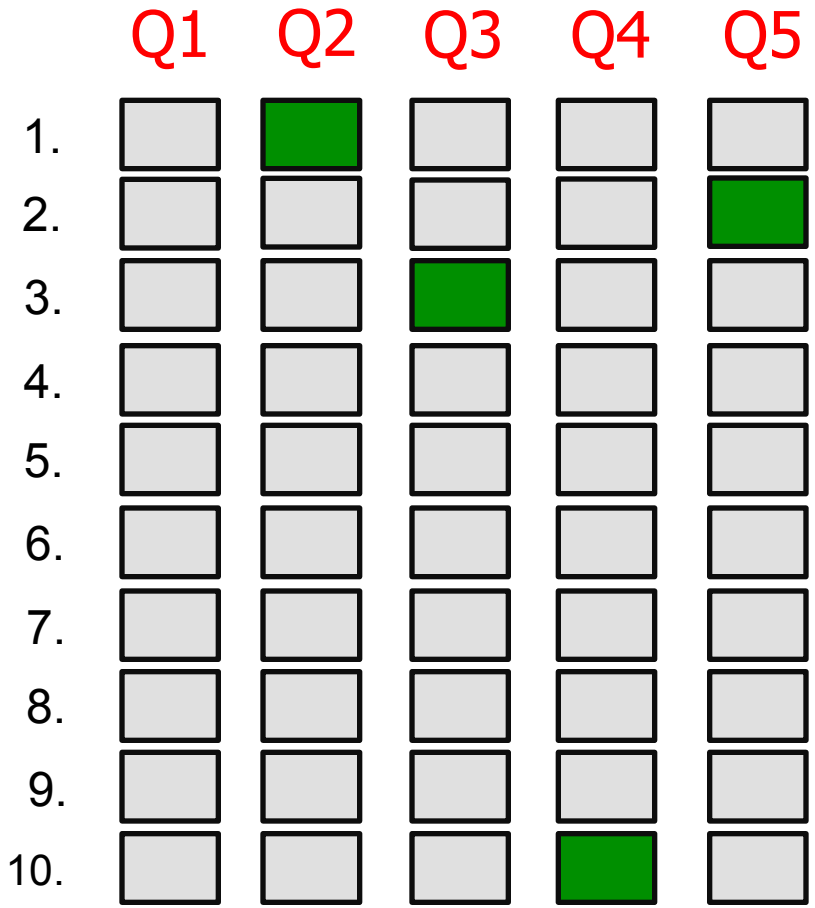
$$P(Entry\ page \mid root) = 6.44 \times 10^{-3}$$
$$P(Entry\ page \mid subroot) = 3.95 \times 10^{-4}$$
$$P(Entry\ page \mid path) = 9.55 \times 10^{-5}$$
$$P(Entry\ page \mid file) = 3.85 \times 10^{-6}$$

- URL type
  - Root: http://www.sigir.org
  - Subroot: http://www.sigir.org/sigirlist
  - Path: http://www.sigir.org/sigirlist/issues/
  - File: http://www.sigir.org/resources.html

URL type	Entry page	WT10g
root	79 (73.1%)	12,258 (0.7%)
subroot	15 (13.9%)	37,959 (2.2%)
path	8 (7.4%)	83,734 (4.9%)
file	6 (5.6%)	1,557,719 (92.1%)

Mean Reciprocal Rank

One system, five queries.



One relevant document per query.

$$RR = \frac{1}{\text{rank of relevant document}}$$

	Q1	Q2	Q3	Q4	Q5
RR	0.0	1.0	0.33	0.01	0.5

MRR=0.369

Language modeling: P(D)

Kraaij et al. [6]

- Results in MRR

small amounts of smoothing

Ranking	Content (λ=0.1)	Anchors (λ=0.1)
$P(Q\|D)$	0.3375	0.4188
$P(Q\|D)P_{doclen}(D)$	0.2634	0.5600
$P(Q\|D)P_{URL}(D)$	**0.7705**	0.6301
$P(Q\|D)P_{inlink}(D)$	0.4974	0.5365

Binary independence model and BM25(F)

Probability Ranking Principle

Stephen Robertson

- Theoretical basis for probabilistic IR
  - Optimizes results for ad hoc retrieval
- Ad hoc retrieval setup:
  - Corpus, user query
  - Wanted: a ranked list of documents
- In what order should the documents be retrieved?
  - In LM we rank by $P(q|d)$
- Binary notion of relevance
  - indicator variable: $R_{D,Q}=\{0,1\}$

relevant
non-relevant

no explicit notion of relevance in LM

Probability Ranking Principle

Stephen Robertson

- Retrieve documents in decreasing order of their estimated probability of relevance
  - At each rank position i the system should select D_i

$$D_i = \arg\max_{D \in RE \setminus RA} P(R_{D,Q} = 1 \mid D, Q)$$

retrieved documents ranked documents

- "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** […], the overall effectiveness of the system to its user will be **the best that is obtainable**…" [2]

Probability Ranking Principle

Stephen Robertson

- *Bayes optimal decision rule* for **set** retrieval (in place of ranked retrieval)

$$D \text{ is relevant iff } P(R=1 \mid D,Q) > P(R=0 \mid D,Q)$$

- PRP assumptions
  - Each document's relevance is independent of all other relevance assessments
  - High accuracy in the probability of relevance

- Question: how to estimate $P(R=1|D,Q)$ and $P(R=0|D,Q)$

Binary independence model (BIM)

- Classic model used with PRP
- Simplifying assumptions to make modeling $P(R|D,Q)$ feasible
- The "binary" in BIM: documents and queries as binary term incidence vectors

$$D \; as \; \vec{x} = (x_1, x_2, ..., x_M), \; where \; x_i = \{0,1\}$$

many documents with the same representation

- The "independence" in BIM: terms are modeled as occuring independently in documents
- Terms not appearing in the query do not affect the ranking

Binary independence model (BIM)

- $P(R|D,Q)$ modeled with incidence vectors

$$P(R=1|\vec{x},\vec{q}) = \frac{P(\vec{x}|R=1,\vec{q})P(R=1|\vec{q})}{P(\vec{x}|\vec{q})}$$

$$P(R=0|\vec{x},\vec{q}) = \frac{P(\vec{x}|R=0,\vec{q})P(R=0|\vec{q})}{P(\vec{x}|\vec{q})}$$

Bayes rule

How to compute?

probability that if a relevant/non-relevant document is retrieved, its document representation is $\vec{x}$
(from the space of all possible documents)

Binary independence model (BIM)

- $P(R|D,Q)$ modeled with incidence vectors

$$P(R=1 \mid \vec{x},\vec{q}) = \frac{P(\vec{x} \mid R=1,\vec{q})P(R=1 \mid \vec{q})}{P(\vec{x} \mid \vec{q})}$$

$$P(R=0 \mid \vec{x},\vec{q}) = \frac{P(\vec{x} \mid R=0,\vec{q})P(R=0 \mid \vec{q})}{P(\vec{x} \mid \vec{q})}$$

Easy to compute if we knew the total number of relevant documents in the corpus

Prior probability of retrieving a relevant/ non-relevant document given a query

$$P(R=1 \mid \vec{x},\vec{q}) + P(R=0 \mid \vec{x},\vec{q}) = 1$$

Binary independence model (BIM)

- We are interested in a ranking of documents ➔ $P(R=1|D,Q)$ is difficult to determine, use easier to compute quantities which result in the **same ordering**

- Rank documents by the odds of relevance

$$O(R \mid \vec{x},\vec{q}) = \frac{P(R=1 \mid \vec{x},\vec{q})}{P(R=0 \mid \vec{x},\vec{q})} = \frac{\frac{P(R=1 \mid \vec{q})P(\vec{x} \mid R=1,\vec{q})}{\cancel{P(\vec{x} \mid \vec{q})}}}{\frac{P(R=0 \mid \vec{q})P(\vec{x} \mid R=0,\vec{q})}{\cancel{P(\vec{x} \mid \vec{q})}}}$$

constant given Q

$$= \cancel{\frac{P(R=1 \mid \vec{q})}{P(R=0 \mid \vec{q})}} \times \frac{P(\vec{x} \mid R=1,\vec{q})}{P(\vec{x} \mid R=0,\vec{q})}$$

Binary independence model (BIM)

- We are left with:

term independence assumption

$$\frac{P(\vec{x} \mid R=1,\vec{q})}{P(\vec{x} \mid R=0,\vec{q})} = \prod_{t=1}^{M} \frac{P(x_t \mid R=1,\vec{q})}{P(x_t \mid R=0,\vec{q})}$$

- In odds notation:

$$O(R \mid \vec{x},\vec{q}) = O(R \mid \vec{q}) \times \prod_{t=1}^{M} \frac{P(x_t \mid R=1,\vec{q})}{P(x_t \mid R=0,\vec{q})}$$

separate terms occurring and not occurring in the document

$$= O(R \mid \vec{q}) \times \prod_{t:x_t=1} \frac{P(x_t \mid R=1,\vec{q})}{P(x_t \mid R=0,\vec{q})} \times \prod_{t:x_t=0} \frac{P(x_t \mid R=1,\vec{q})}{P(x_t \mid R=0,\vec{q})}$$

Binary independence model (BIM)

- Let:

$$p_t = P(x_t = 1 \mid R = 1, \vec{q})$$
$$u_t = P(x_t = 1 \mid R = 0, \vec{q})$$

probability of a term occurring in a $R=\{0,1\}$ document

- Add another assumption: terms not occurring in the query are equally likely in both classes

$$if\ q_t = 0\ then\ p_t = u_t$$

- Simplifies the odds equation further

query terms found in D

query terms not found in D

$$O(R \mid \vec{x}, \vec{q}) = O(R \mid \vec{q}) \times \prod_{t:x_t=q_t=1} \frac{p_t}{u_t} \times \prod_{t:x_t=0,q_t=1} \frac{1-p_t}{1-u_t}$$

Binary independence model (BIM)

constant for a given query

- Another reformulation (right side now over all query terms)

$$O(R \mid \vec{x}, \vec{q}) = O(R \mid \vec{q}) \times \prod_{t: x_t = q_t = 1} \frac{p_t (1 - u_t)}{u_t (1 - p_t)} \times \prod_{t: q_t = 1} \frac{1 - p_t}{1 - u_t}$$

left to estimate to rank the documents

- Rank documents according to the **retrieval status value** (RSV)

$$RSV_D = \log \prod_{t: x_t = q_t = 1} \frac{p_t (1 - u_t)}{u_t (1 - p_t)} = \sum_{t: x_t = q_t = 1} \log \frac{p_t (1 - u_t)}{u_t (1 - p_t)}$$

Log is monotonic!

Binary independence model (BIM)

- Reformulate!

$$c_t = \log \frac{p_t(1-u_t)}{u_t(1-p_t)} = \log \frac{p_t}{(1-p_t)} + \log \frac{1-u_t}{u_t}$$

odds of the term appearing if the document is relevant

odds of the term appearing if the document is non-relevant

- $c_t=0$ if term t is equally likely to appear in relevant and non-relevant documents
- $c_t>0$ if t is more likely to appear in relevant documents
- $c_t<0$ if t is more likely to appear in non-relevant documents

Binary independence model (BIM)

- Estimate c_t for a given corpus and query
- Assume we know the number of relevant documents (S)

	documents	**relevant**	**non-relevant**	**total**
term present	$x_t=1$	s	df_t-s	df_t
term absent	$x_t=0$	S-s	$(N$-$df_t)$-$(S$-$s)$	N-df_t
	total	S	N-S	N

$$p_t = s / S \text{ and } u_t = (df_t - s) / (N - S)$$

$$c_t = \log \frac{s/(S-s)}{(df_t - s)/((N - df_t) - (S - s))}$$

+0.5 +0.5 +0.5 +0.5

smoothing

Binary independence model (BIM)

In practice: probabilities for the non-relevant components

- Very few relevant documents usually exist (e.g. 3 documents out of 7 million documents in our Wikipedia corpus)
- Estimate the probabilities across all documents in the corpus

$$u_t = \log \frac{1-u_t}{u_t} = \log \frac{N-df_t}{df_t} \approx \log \frac{N}{df_t}$$

theoretical justification for IDF

Binary independence model (BIM)

In practice: probabilities for the non-relevant components

- Many variations, usually difficult to estimate accurately
- Croft & Harper (1979)
  - Assume $p_t=0.5$ and let it be constant for all query terms
  - Equally likely to appear in relevant/non-relevant documents
  - In effect, the documents are ranked by the query terms occurring in the documents scaled by their IDF weighting
  - Weak estimate, but can be useful
    - Short documents (titles, paper abstracts)

Binary independence model (BIM)

In practice: probabilities for the non-relevant components

- Many variations, usually difficult to estimate accurately
- Greiff (1998)
  - Empirical observation: p_t rises with df_t (just think about stopwords)
  - Proposal:

$$p_t = \frac{1}{3} + \frac{2}{3} \times \frac{df_t}{N}$$

Binary independence model (BIM)

In practice: probabilities for the relevant components

- Many variations, usually difficult to estimate accurately
- If a few relevant documents are known, the probabilities can be estimated across those
  - Relevance feedback
  - Effectiveness dependent on the number of relevant documents and the document content (typical for the class of relevant documents?)

Binary independence model (BIM)

In practice: iterative (pseudo)-relevance feedback

1. Guess p_t and u_t
2. Retrieve a set of candidate relevant documents (based on our initial estimates)
3. The user judges **a few** documents as relevant (VR) and non-relevant (VNR)
4. Revise the model from the judgments

$$VR = \{D \in V, R_{D,Q} = 1\} \subset R, VNR = \{D \in V, R_{D,Q} = 0\}$$

5. Re-estimate p_t and u_t via Bayesian updating, e.g.

$$p_t^{(k+1)} = \frac{|VR_t| + \kappa p_t^{(k)}}{|VR| + \kappa}$$

6. Repeat from step 2

simpler:

$$p_t = \frac{|VR_t| + \frac{1}{2}}{|VR| + 1}$$

Okapi BM25

- In the early years, PRP &BIM
  - Offered a good theoretical justification
  - Required partial relevance judgments
  - Without such judgments, degrades to adhoc term weighting models (e.g. IDF)
- This changed with the development of BM25
  - High retrieval effectiveness
  - Today still used as a baseline in research
- BIM neither includes term frequencies nor document length
  - Okapi BM25 does!

Okapi BM25

Development of the scoring function

$$\frac{N - df_t + 0.5}{df_t + 0.5}$$ can be negative; floor to zero!

include tf and document length

$$RSV_D = \sum_{t \in Q} \log \frac{N - df_t + 0.5}{df_t + 0.5}$$

$$RSV_D = \sum_{t \in Q} \log \frac{N - df_t + 0.5}{df_t + 0.5} \times \frac{(k_1 + 1) tf_{t,D}}{k_1((1-b) + b \times (L_D / L_{av})) + tf_{t,D}}$$

positive tuning value
$k_1=0$: binary model
$k_1>>0$: raw tf values

scaling by document length
$b=0$: no normalization
$b=1$: full normalization
$b \in [0,1]$

document length

av. document length in corpus

Okapi BM25

Long queries: query term weighting

length normalization unnecessary

$$RSV_D = \sum_{t \in Q} \log \frac{N}{df_t} \times \frac{(k_1+1)tf_{t,D}}{k_1((1-b)+b \times (L_D / L_{av}))+tf_{t,D}} \times \frac{(k_3+1)tf_{t,Q}}{k_3+tf_{t,Q}}$$

positive tuning value

TREC 2001 Web adhoc topic
Narrative: A document that merely states that the Vikings visited or lived in Scotland is not relevant. A relevant document must mention the source of the information, such as relics, sagas, runes or other records from those times.

Parameter settings

Ideally: use separate train/test collections

Often: $k_1, k_3 \in [1.2, 2], b = 0.75$

Empirical comparison

TF.IDF, BM25 and LM with Dirichlet smoothing

- 3 TREC corpora, reported is Mean Average Precision

0.5M docs
Av. length: 266
News articles

1.7M docs
Av. length: 378
Web

25.2M docs
Av. length: 665
Web

	queries	TF.IDF	Okapi	LM (μ=1000)
TREC Vol. 4+5	301-350			
	351-400			
	401-450			
WT10g	451-500			
	501-550			
GOV2	701-750			
	751-800			
	801-850			

Empirical comparison

TF.IDF, BM25 and LM with Dirichlet smoothing

- 3 TREC corpora, reported is Mean Average Precision

0.5M docs
Av. length: 266
News articles

1.7M docs
Av. length: 378
Web

25.2M docs
Av. length: 665
Web

	queries	TF.IDF	Okapi	LM (μ=1000)
TREC Vol. 4+5	301-350	0.109		
	351-400	0.073		
	401-450	0.088		
WT10g	451-500	0.055		
	501-550	0.061		
GOV2	701-750	0.029		
	751-800	0.036		
	801-850	0.023		

Empirical comparison

TF.IDF, BM25 and LM with Dirichlet smoothing

- 3 TREC corpora, reported is Mean Average Precision

0.5M docs
Av. length: 266
News articles

1.7M docs
Av. length: 378
Web

25.2M docs
Av. length: 665
Web

	queries	TF.IDF	Okapi	LM (μ=1000)
TREC Vol. 4+5	301-350	0.109	0.218	
	351-400	0.073	0.176	
	401-450	0.088	0.223	
WT10g	451-500	0.055	0.183	
	501-550	0.061	0.163	
GOV2	701-750	0.029	0.230	
	751-800	0.036	0.296	
	801-850	0.023	0.250	

Empirical comparison

TF.IDF, BM25 and LM with Dirichlet smoothing

- 3 TREC corpora, reported is Mean Average Precision

0.5M docs
Av. length: 266
News articles

1.7M docs
Av. length: 378
Web

25.2M docs
Av. length: 665
Web

	queries	TF.IDF	Okapi	LM (μ=1000)
TREC Vol. 4+5	301-350	0.109	0.218	0.226
	351-400	0.073	0.176	0.187
	401-450	0.088	0.223	0.245
WT10g	451-500	0.055	0.183	0.207
	501-550	0.061	0.163	0.180
GOV2	701-750	0.029	0.230	0.269
	751-800	0.036	0.296	0.324
	801-850	0.023	0.250	0.297

BM25F

Extending BM25 to document fields [8]

```
<page>
        <title>Anarchism</title>
        <id>12</id>
        <contributor>
                <username>Skomorokh</username>
                <id>1749684</id>
        </contributor>
        <comment>
                /* External links */ partial reversion - we don't
                link to forums per [[WP:EL]]
        </comment>
        <text xml:space="preserve">
                '''Anarchism''' is a [[political philosophy]] encompassing
                theories and attitudes which support the elimination of all ...
        </text>
</page>
```

important

unimportant

potentially useful

BM25F

Extending BM25 to document fields [8]

- Textual data often found in some sort of structural form
- Retrieval effectiveness can be improved by taking the structure into account

- Simple solution: calculate score for each field and combine them linearly

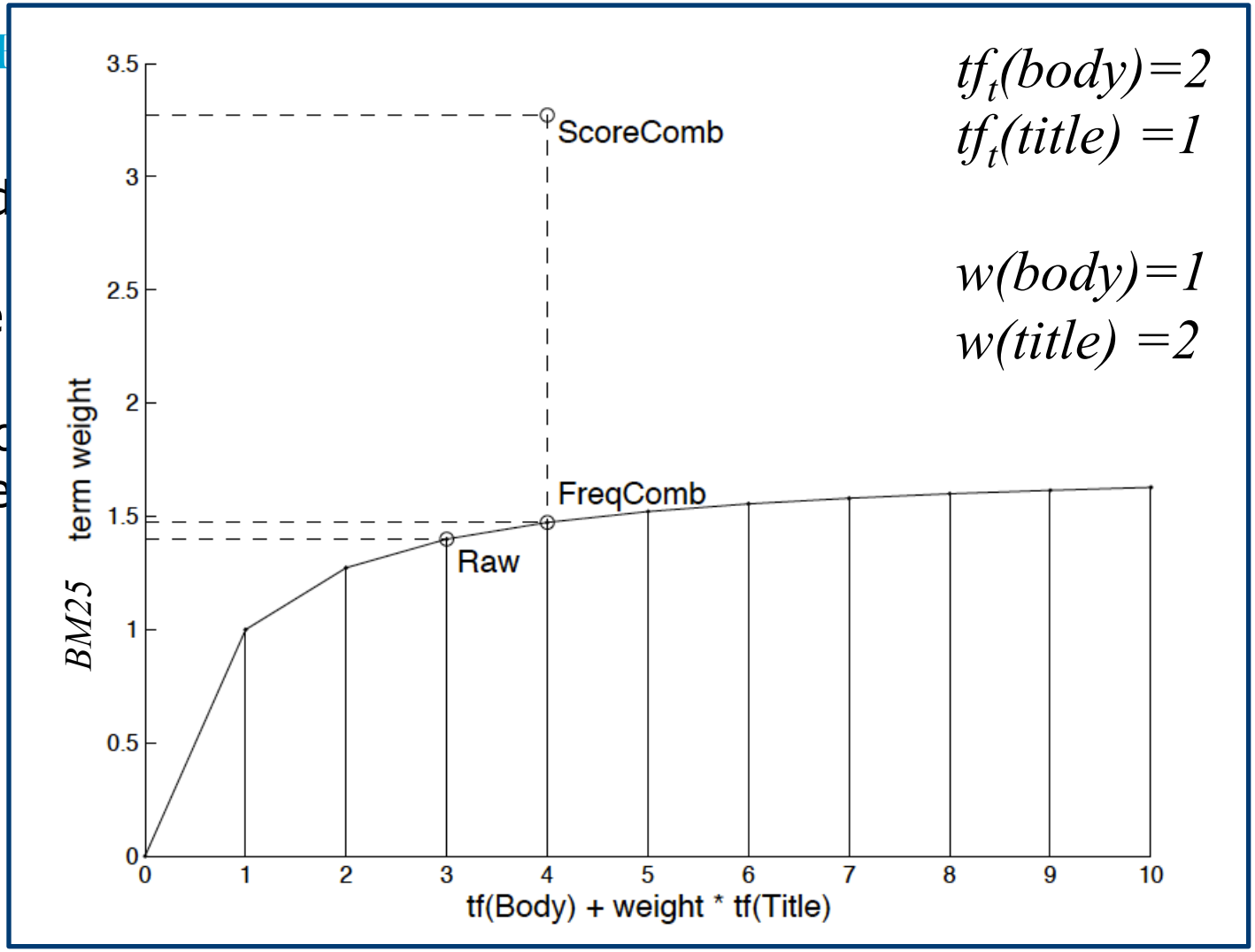
$$RSV_D = \sum_{k=1}^{K} v_f \times RSV_{D_f}$$

Source: [8]

BM25F

Extending I

- Textual d
- Retrieval
  structure
- Simple sc
  them line



Source: [8]

BM25F

Extending BM25 to document fields [8]

- Textual data often found in some sort of structural form
- Retrieval effectiveness can be improved by taking the structure into account

- Simple solution: calculate score for each field and combine them in a linear fashion:

$$RSV_D = \sum_{k=1}^{K} v_f \times RSV_{D_f}$$

- TF usually non-linear: information gained by observing a term for the first time is greater than observing subsequent occurrences
  - Linear combination of scores breaks this relation

BM25F

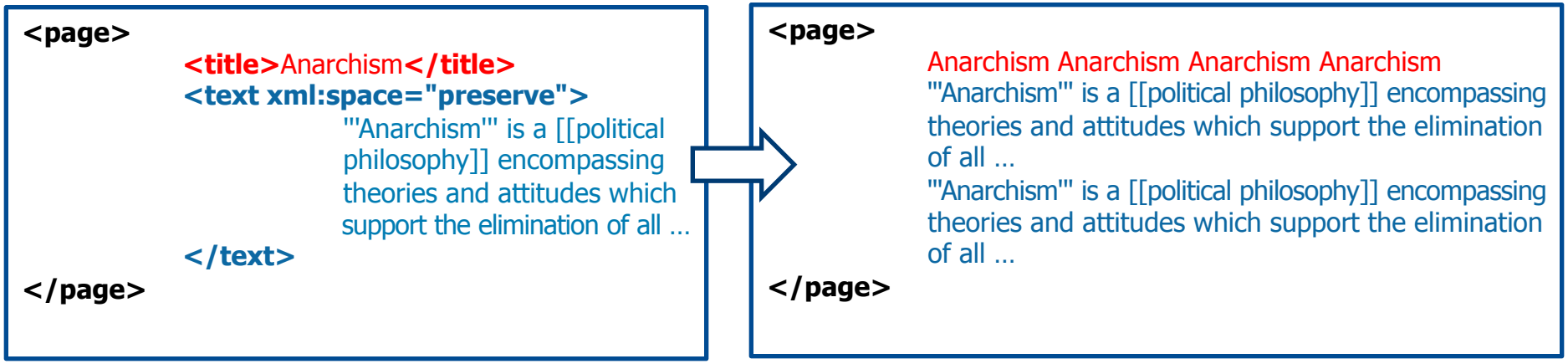
Extending BM25 to document fields [8]

- Undesirable effects of linear combination
  - A document matching a single query term over several fields can score much higher than a document matching several query terms
  - Term weights need to be kept small to preserve term dependence (e.g. a weight of 0.1 for *title* would bring *raw* and *ScoreComb* closer together)
- What about the IDF component?
  - If corpus statistics are computed per field, IDF can vary highly in different fields (e.g. stopwords scoring highly in the title field)
- Extensive parameter tuning necessary (per field)

BM25F

Extending BM25 to document fields [8]

- Solution: term frequency combination
  - Map the structured collection into unstructured space with modified term frequencies: combine original term frequencies in the different fields in a weighted manner



$$w(title)=4,\ w(body)=2$$

  - Rank in the usual manner

Evaluation

Task-dependent evaluation

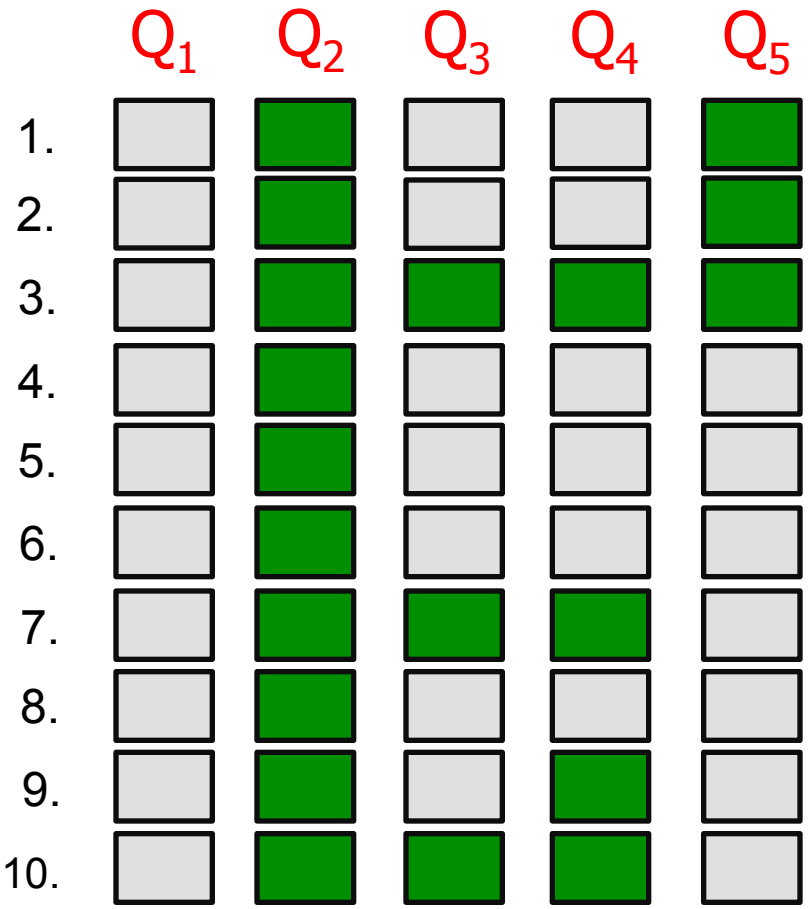
Evaluation is an ongoing research topic.

- Query: homepage TU Delft
  - Navigational
  - ~1 relevant entry page
- Query: TU Delft world-wide university ranking
  - Informational query
  - N relevant Web pages, retrieving some is good enough
  - It would also be beneficial to retrieve diverse results *
- Query: TU Delft patents nano-technology
  - Informational
  - N relevant patents, retrieving all is important
- Query: successful treatment of Newcastle disease
  - Informational
  - N relevant Web pages, retrieving all is important

Broder's query classification:
- Navigational
- Informational
- Transactional (buy house")

Mean Average Precision

One system, five queries.



	Q_1	Q_2	Q_3	Q_4	Q_5
AvP	0.0	1.0	0.09	0.13	0.3

Given a set of queries, the average effectiveness is the mean over AvP.

$$MAP = \frac{1}{|\mathbf{Q}|} \sum_{Q \in \mathbf{Q}} \frac{\sum_{k=1}^{s} P@k \times rel(k)}{R}$$

MAP=0.364

GMAP

Geometric mean average precision

- A measure designed to highlight improvements for low-performing topics
- Geometric mean of per-topic average precision values

$$GMAP = \sqrt[n]{\prod_n AP_n}\ , n\ is\ \#topics$$

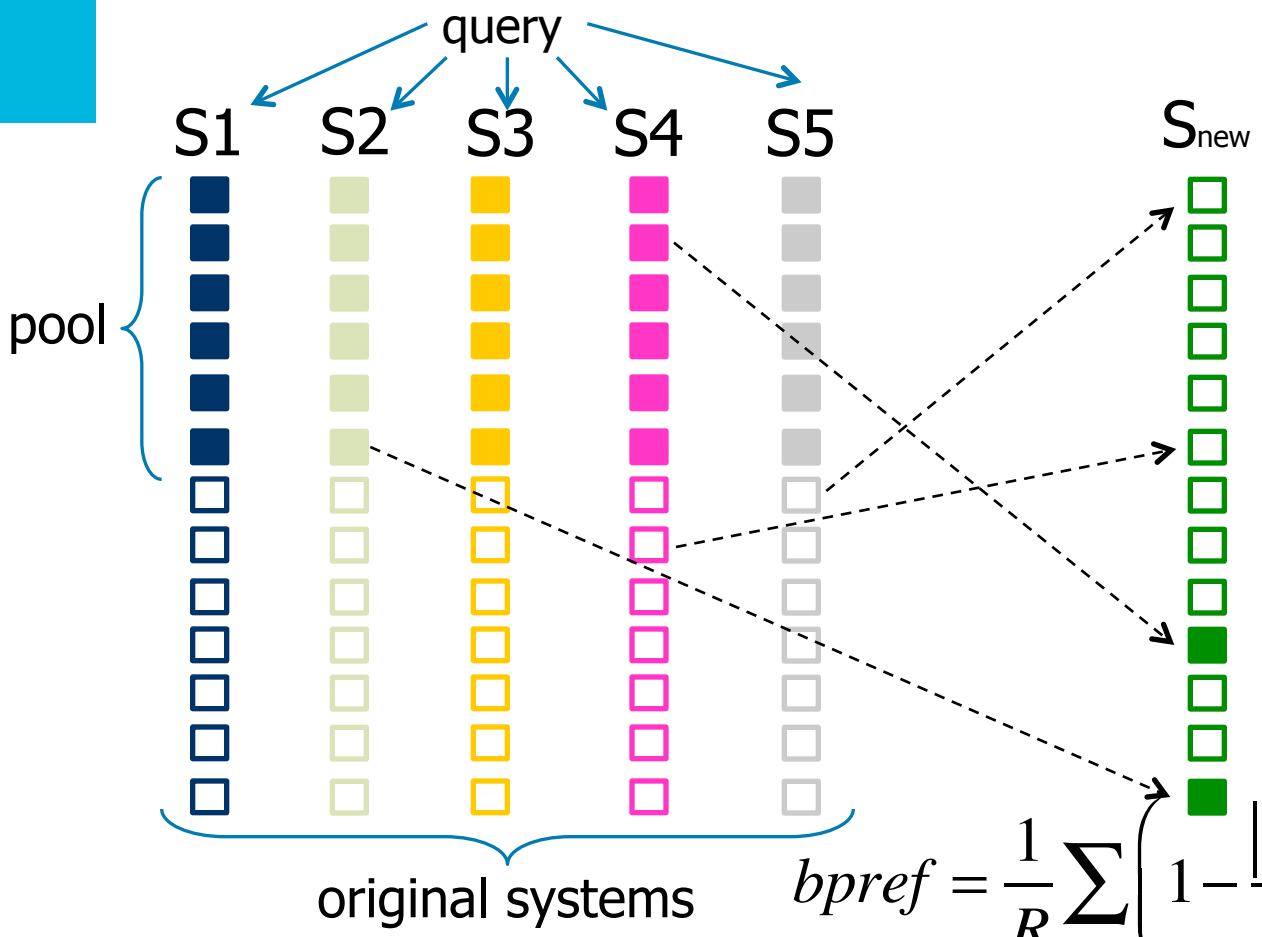
$$= \exp \frac{1}{n} \sum_n \log AP_n$$

Two systems, five queries.
AP values shown.

	S_1	S_2
1.	0.60	0.58
2.	0.20	0.18
3.	0.01	0.03
4.	0.04	0.06
5.	0.90	0.90
	MAP =0.350	MAP =0.350
	GMAP=0.134	GMAP=0.176

bpref

Buckley & Voorhees, 2004



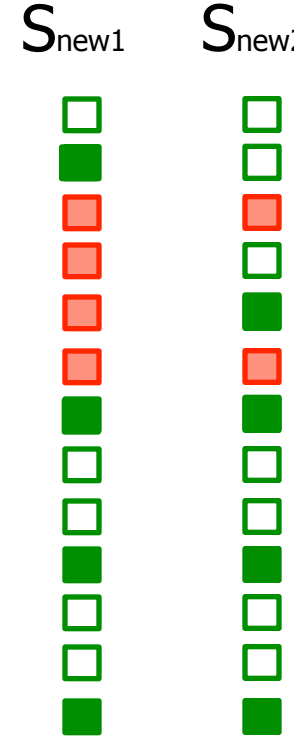
Traditional measures:
not judged == non-relevant
(biased against new systems)

bpref: considers only *judged* documents (rel./non-rel.)

R #judged relevant
N #judged non-relevant
r relevant retrieved document
n member of the first R judged non-relevant documents as retrieved by the system

$$bpref = \frac{1}{R}\sum_{r}\left(1 - \frac{|n \text{ ranked higher than } r|}{R}\right)$$

bpref



$$bpref = \frac{1}{R}\sum_{r}\left(1 - \frac{|n \text{ ranked higher than } r|}{R}\right)$$

$$assume\ R = 20$$

0.170 0.1825

Evaluation: points to remember

- Evaluation is not straight-forward
- Still researched today
- **The task is paramount to the correct choice of evaluation measure**

Sources

① Introduction to Information Retrieval. Manning et al. 2008
② Information retrieval. Keith van Rijsbergen, 1979
③ Managing gigabytes, Witten et al.
④ The probability ranking principle in IR, S.E. Robertson, 1977
⑤ A study of smoothing methods for language models applied to ad hoc information retrieval. Zhai & Lafferty. 2001.
⑥ The importance of prior probabilities for entry page search. Kraaij et al. 2002.
⑦ Cluster-based retrieval using language models. Liu & Croft, 2004.
⑧ Simple BM25 extension to multiple weighted fields. Robertson et al. 2004.
⑨ Cumulated gain-based evaluation of IR techniques. Järvelin & Kekäläinen. 2002