Hypertext databases

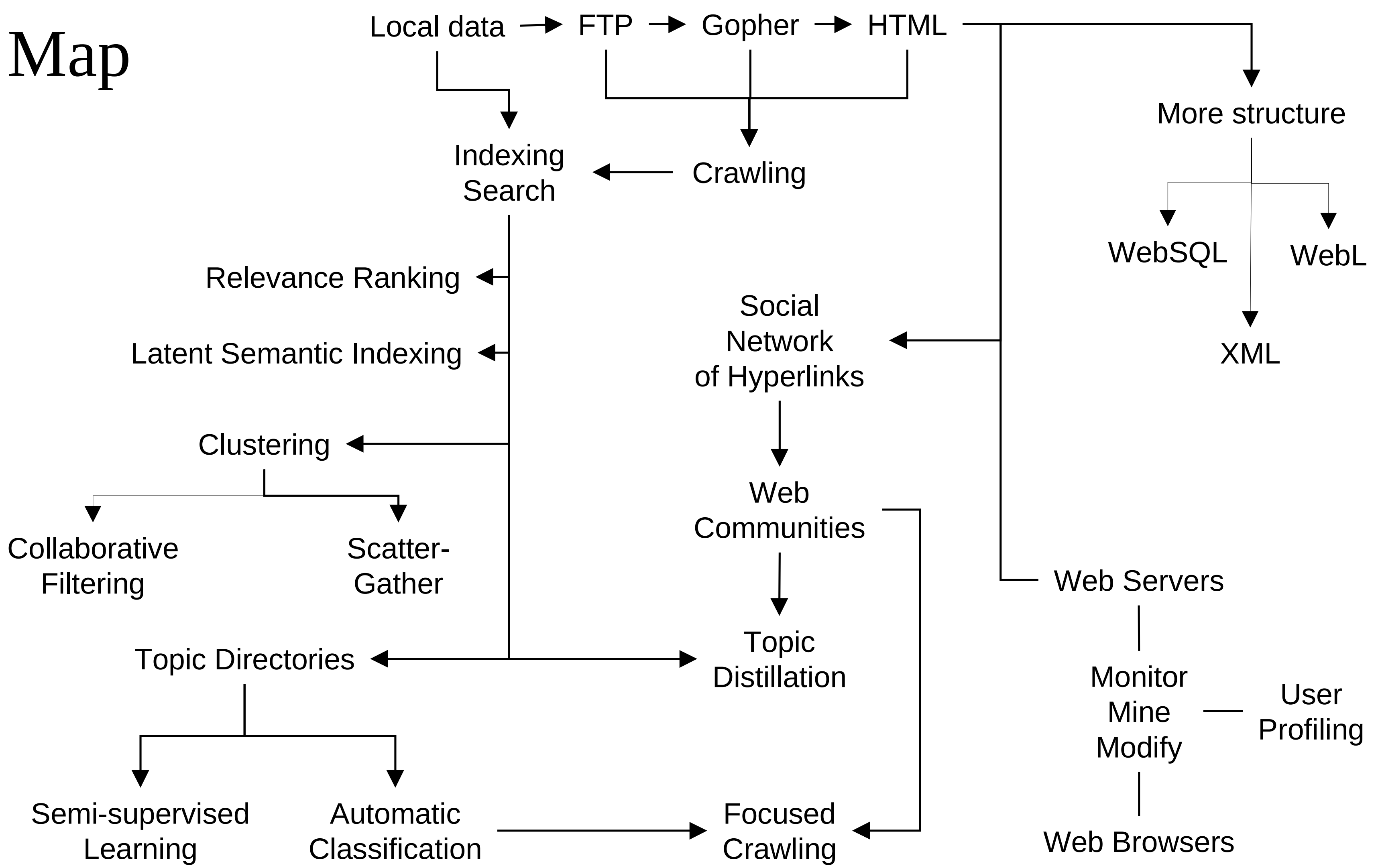
- Academia
  - Digital library, web publication
- Consumer
  - Newsgroups, communities, product reviews
- Industry and organizations
  - Health care, customer service
  - Office documents, email

The Web is bigger than the sum of its parts

Search products and services

- Verity
- Fulcrum
- PLS
- Oracle text extender
- DB2 text extender
- Infoseek Intranet
- SMART (academic)
- Glimpse (academic)
- Inktomi (HotBot)
- Alta Vista
- Google!
- Yahoo!
- Infoseek Internet
- Lycos
- Excite

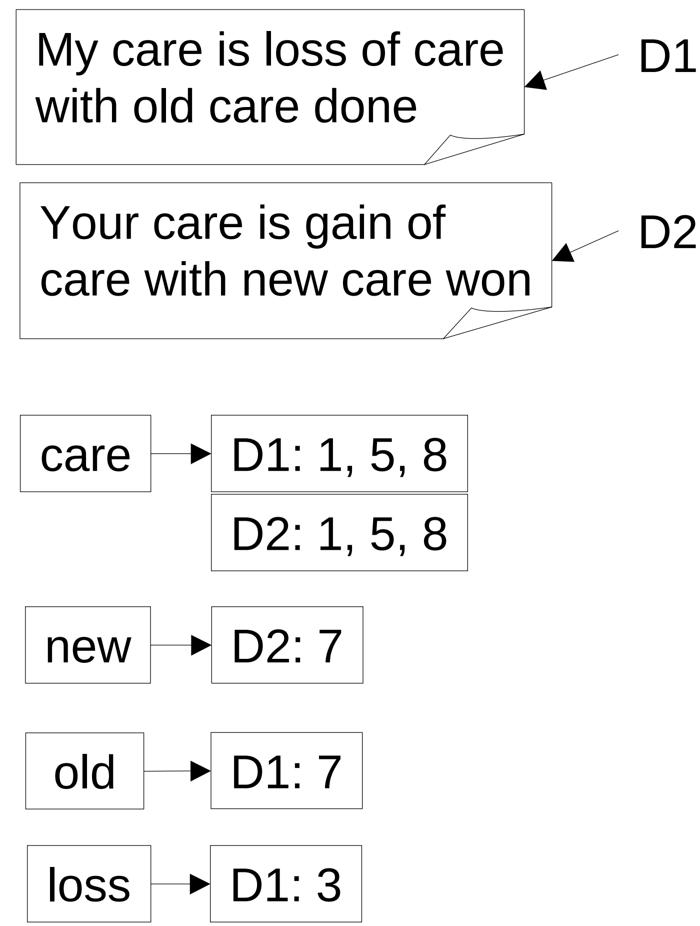
Map



Local data
FTP
Gopher
HTML
More structure
Indexing Search
Crawling
WebSQL
WebL
XML
Relevance Ranking
Latent Semantic Indexing
Social Network of Hyperlinks
Clustering
Web Communities
Collaborative Filtering
Scatter-Gather
Web Servers
Topic Directories
Topic Distillation
Monitor Mine Modify
User Profiling
Semi-supervised Learning
Automatic Classification
Focused Crawling
Web Browsers

Keyword indexing

- Boolean search
  - care AND NOT old
- Stemming
  - gain*
- Phrases and proximity
  - “new care”
  - loss NEAR/5 care
  - <SENTENCE>



Tables and queries 1

POSTING

tid	did	pos
care	d1	1
care	d1	5
care	d1	8
care	d2	1
care	d2	5
care	d2	8
new	d2	7
old	d1	7
loss	d1	3
…	…	…

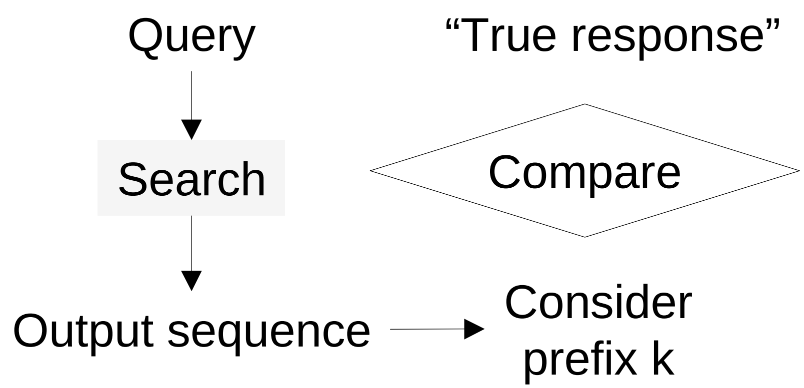
select distinct did from POSTING where tid = ‘care’ except
select distinct did from POSTING where tid like ‘gain%’

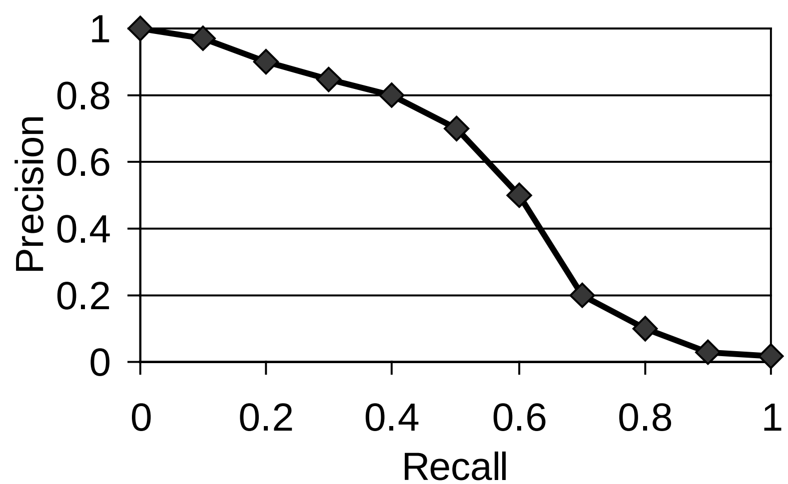
with
TPOS1(did, pos) as
        (select did, pos from POSTING where tid = ‘new’),
TPOS2(did, pos) as
        (select did, pos from POSTING where tid = ‘care’)
select distinct did from TPOS1, TPOS2
        where TPOS1.did = TPOS2.did
        and **proximity**(TPOS1.pos, TPOS2.pos)

proximity(a, b) ::=
        a + 1 = b
        abs(a - b) < 5

Relevance ranking

- Recall = coverage
  - What fraction of relevant documents were reported
- Precision = accuracy
  - What fraction of reported documents were relevant
- Trade-off





Vector space model and TFIDF

- Some words are more important than others
- W.r.t. a document collection D
  - d_+ have a term, d_- do not
  - "Inverse document frequency"

$$1 + \log \frac{d_+ + d_-}{d+}$$

- "Term frequency" (TF)
  - Many variants:

$$\frac{n(d,t)}{\sum_t n(d,t)}, \frac{n(d,t)}{\max_t n(d,t)}$$

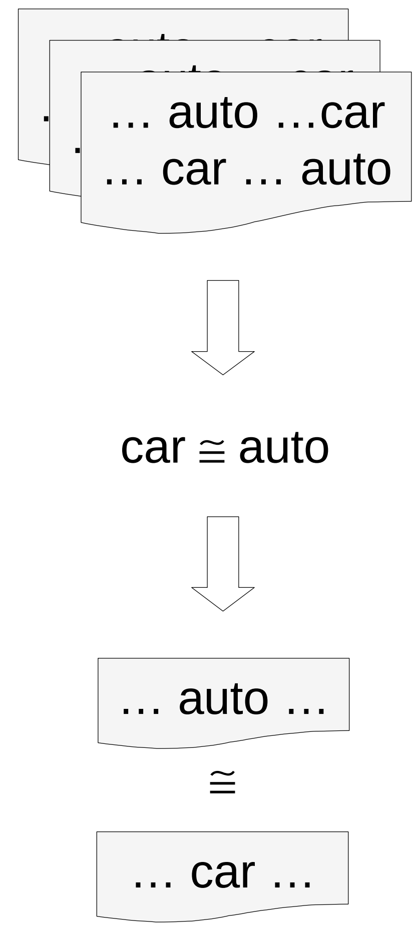
- Probabilistic models

Tables and queries 2

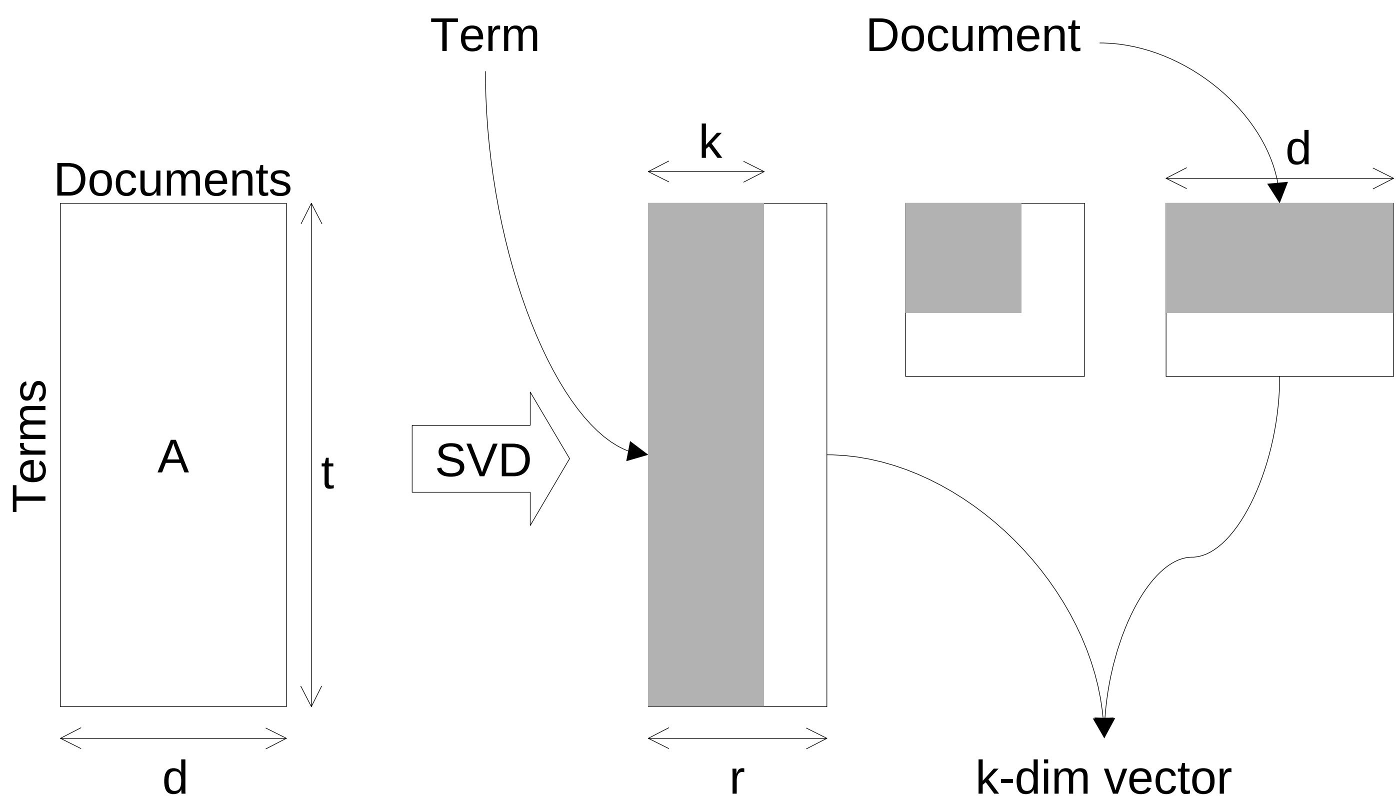
```
VECTOR(did, tid, elem) ::=
With
TEXT(did, tid, freq) as
        (select did, tid, count(distinct pos) from POSTING
        group by did, tid),
LENGTH(did, len) as
        (select did, sum(freq) from TEXT group by did),
DOCFREQ(tid, df) as
        (select tid, count(distinct did) from POSTING
        group by tid)
select did, tid,
(freq / len) * (1 + log((select count(distinct did from POSTING))/df))
from TEXT, LENGTH, DOCFREQ
where TEXT.did = LENGTH.did
and TEXT.tid = DOCFREQ.tid
```

Similarity

- Direct similarity
  - Cosine, normalized distance
- Indirect similarity
  - auto and car co-occur often
  - They must be related
  - Documents having related words are related
- Useful for search and clustering



Latent semantic indexing



Supervised learning (classification)

- Many forms
  - Content: automatically organize the web per Yahoo!
  - Type: faculty, student, staff
  - Intent: education, discussion, comparison, advertisement
- Applications
  - Relevance feedback for re-scoring query responses
  - Filtering news, email, etc.
  - Narrowing searches and selective data acquisition

Difficulties

- Dimensionality
  – Decision tree classifiers: dozens of columns
  – Vector space model: 50,000 ‘columns’
- Context-dependent noise
  – ‘Can’ (v.) considered a ‘stopword’
  – ‘Can’ (n.) may not be a stopword in /Yahoo/SocietyCulture/Environment/Recycling
- Need for scalability
  – High dimension needs more data to learn

Techniques

- Nearest neighbor
  - \+ Standard keyword index also supports classification
  - – How to define similarity? (TFIDF may not work)
  - – Wastes space by storing individual document info
- Rule-based, decision-tree based
  - – Very slow to train (but quick to test)
  - \+ Good accuracy (but brittle rules)
- Model-based
  - \+ Fast training and testing with small footprint

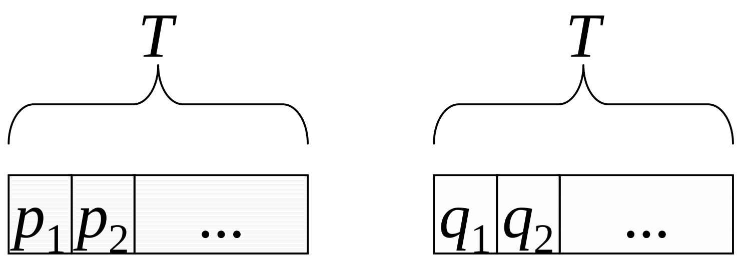
The "bag-of-words" document model

- Decide topic; topic c is picked with prior probability $\pi(c)$; $\sum_c \pi(c) = 1$
- Each topic c has parameters $\theta(c,t)$ for terms t
- Coin with face probabilities $\sum_t \theta(c,t) = 1$
- Fix document length and keep tossing coin
- Given c, probability of document is

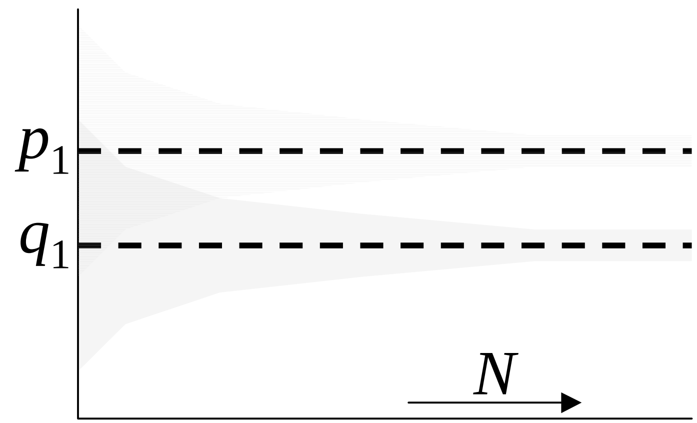
$$\Pr[d \mid c] = \binom{n(d)}{\{n(d,t)\}} \prod_{t \in d} \theta(c,t)^{n(d,t)}$$

Feature selection

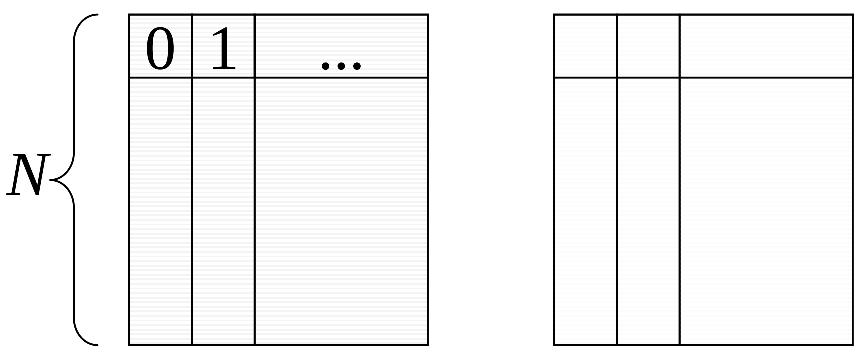
Model with unknown parameters



Confidence intervals



Observed data

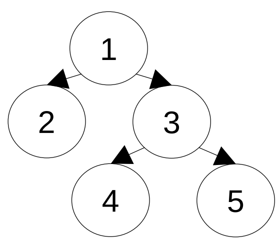


Pick $F \subseteq T$ such that models built over F have high separation confidence

Tables and queries 3

TAXONOMY

pcid	kcid	kcname
	1	
1	2	Arts
1	3	Science
3	4	Math
3	5	Physics



EGMAP

did	kcid

TEXT

did	tid	freq

```
EGMAPR(did, kcid) ::=
        ((select did, kcid from EGMAP) union all
        (select e.did, t.pcid from
        EGMAPR as e, TAXONOMY as t
        where e.kcid = t.kcid))

STAT(pcid, tid, kcid, kdoc, knum) ::=
        (select pcid, tid, TAXONOMY.kcid,
        count(distinct TEXT.did), sum(freq)
        from EGMAPR, TAXONOMY, TEXT
        where TAXONOMY.kcid = EGMAPR.kcid
        and EGMAPR.did = TEXT.did
        group by pcid, tid, TAXONOMY.kcid)
```

Clustering

- Standard notion from structured data analysis
- Techniques
  - Agglomerative, *k*-means
  - Mixture models and Expectation Maximization
- How to reduce distance computation time?
  - Sample points or directions at random
  - Pre-cluster (eliminate redundancy)
  - Project remaining points on to subspace

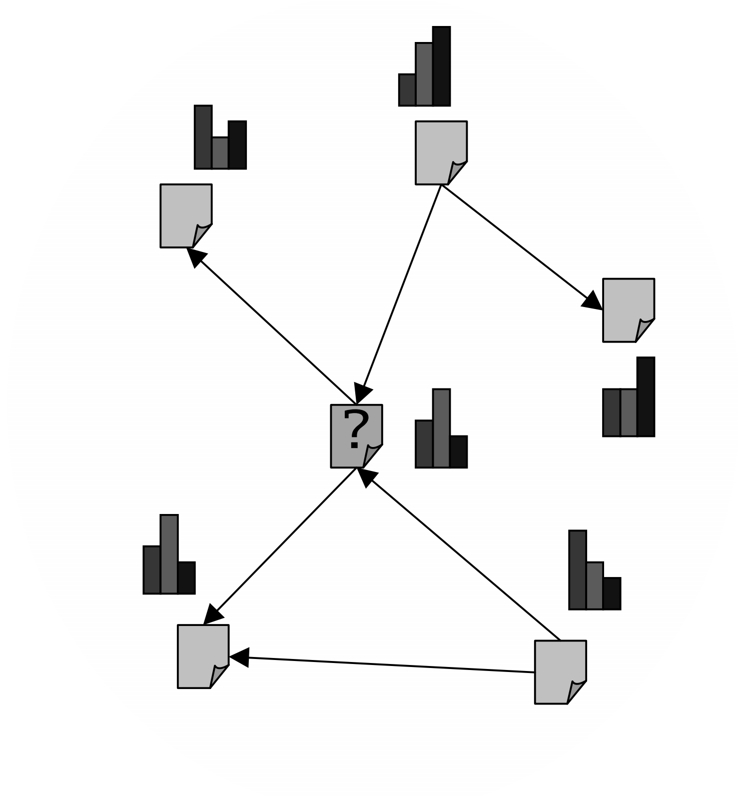
Collaborative filtering

- People=record, movies=features, cluster people
- Both people and features can be clustered
- For hypertext access, time of access is a feature
- Need advanced models

	Batman	Rambo	Andre	Hiver	Whispers	StarWars
Lyle						
Ellen						
Jason						
Fred						
Dean						
Karen						

Hypertext models

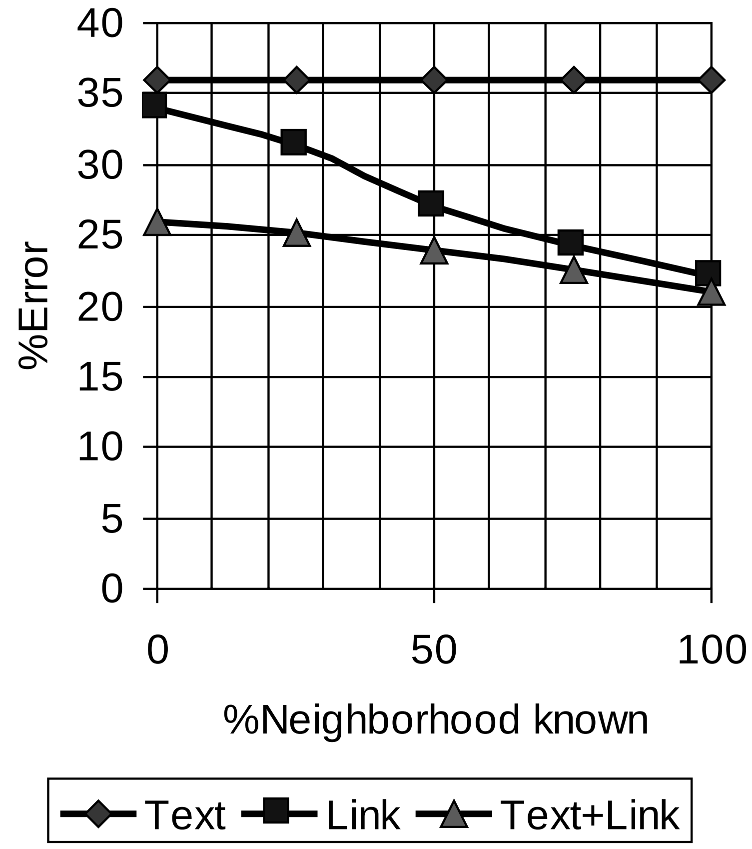
- *c*=class, *t*=text, *N*=neighbors
- Text-only model: Pr[*t*|*c*]
- Using neighbors' text to judge my topic: Pr[*t*, *t*(*N*) | *c*]
- Better model: Pr[*t*, *c*(*N*) | *c*]
- Non-linear relaxation



Result of exploiting link features

- Pretend to know only a % of neighborhood topics
- Bootstrap using text-only classifier
- Use non-linear relaxation to update topic assignment iteratively

☛Link information reduces error significantly



Hyperlink graph analysis

- Hypermedia is a **social network**
  - Telephoned, advised, co-authored, paid, cited
- Social network theory (cf. Wasserman & Faust)
  - Extensive research applying graph notions
  - **Centrality**
  - **Prestige**
  - **Reflected prestige**
- Can be applied directly to Web search
  - HIT, Google, CLEVER, topic distillation

Google and HITS

- In-degree ≈ prestige
- Not all votes worth the same
- Prestige of a page is the sum of prestige of citing pages: $\boldsymbol{p} = \boldsymbol{E}\boldsymbol{p}$
- Pre-compute query independent prestige score
- High prestige ⇔ good authority
- High reflected prestige ⇔ good hub
- Bipartite iteration
  - $\boldsymbol{a} = \boldsymbol{E}\boldsymbol{h}$
  - $\boldsymbol{h} = \boldsymbol{E}^T\boldsymbol{a}$
  - $\boldsymbol{h} = \boldsymbol{E}^T\boldsymbol{E}\boldsymbol{h}$

Tables and queries 4

HUBS

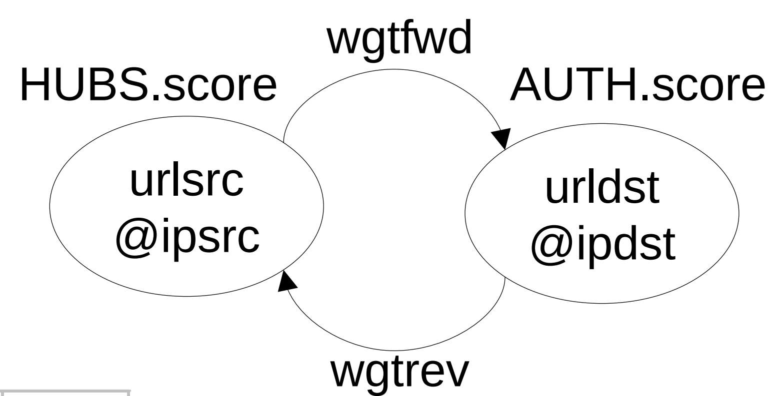
url	score

AUTH

url	score

```
delete from HUBS;
insert into HUBS(url, score)
        (select urlsrc, sum(score * wtrev) from AUTH, LINK
        where authwt is not null and type = non-local
        and ipdst <> ipsrc and url = urldst
        group by urlsrc);
update HUBS set (score) = score /
        (select sum(score) from HUBS);
```

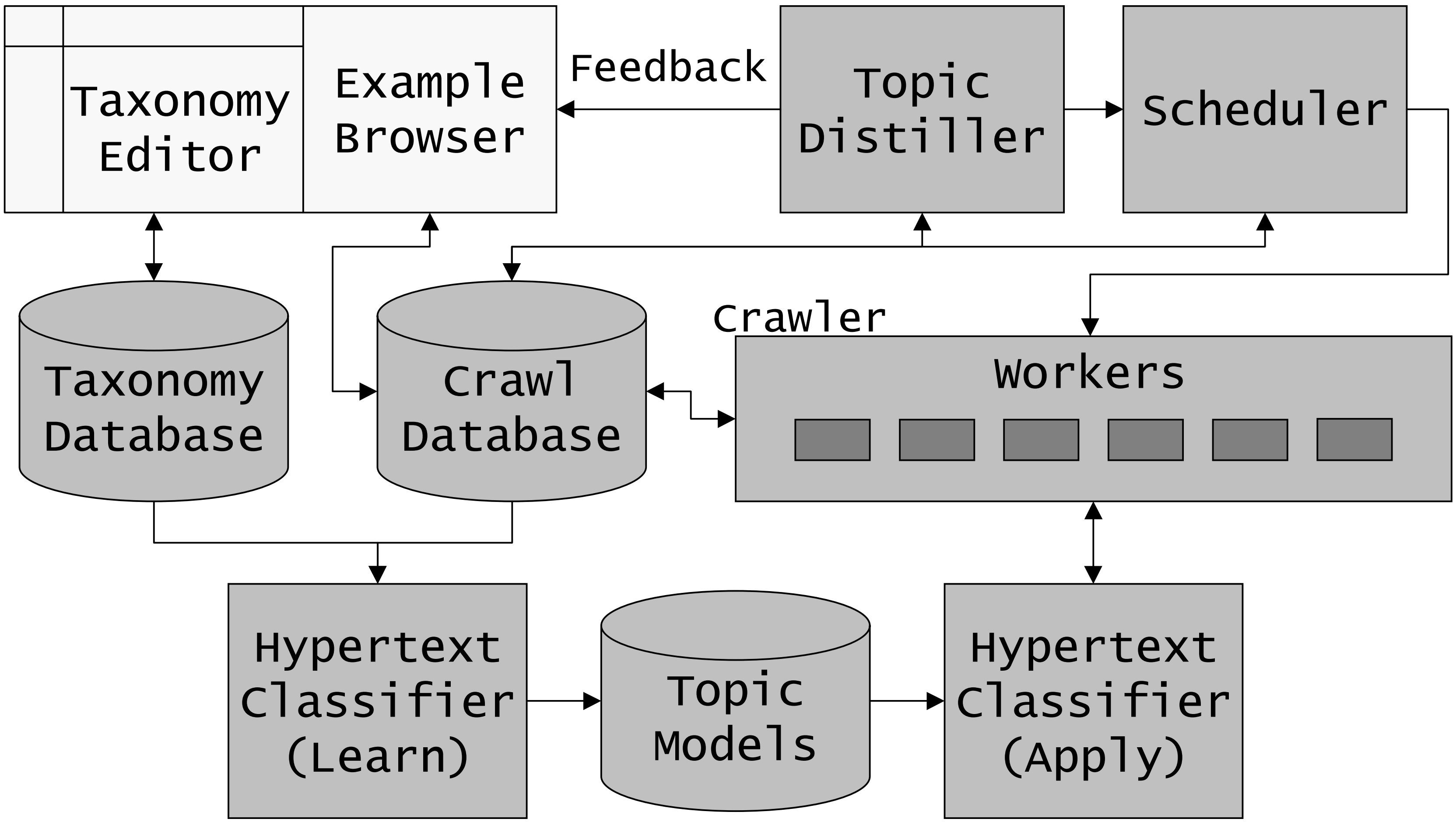
```
update LINK as X set (wtfwd) = 1. /
        (select count(ipsrc) from LINK
        where ipsrc = X.ipsrc
        and urldst = X.urldst)
        where type = non-local;
```



LINK

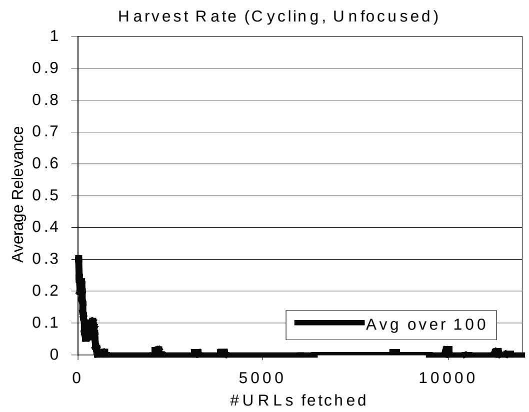
urlsrc	urldst	ipsrc	ipdst	wgtfwd	wtrev	type

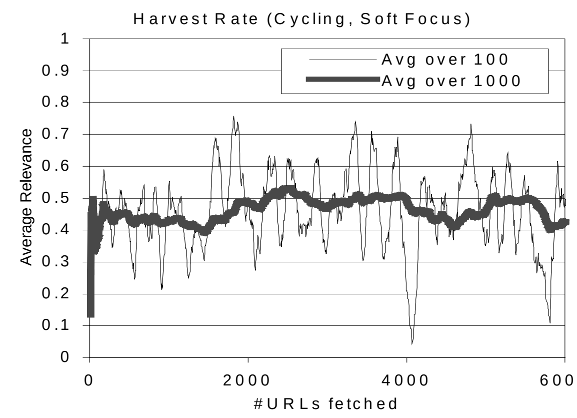
Resource discovery

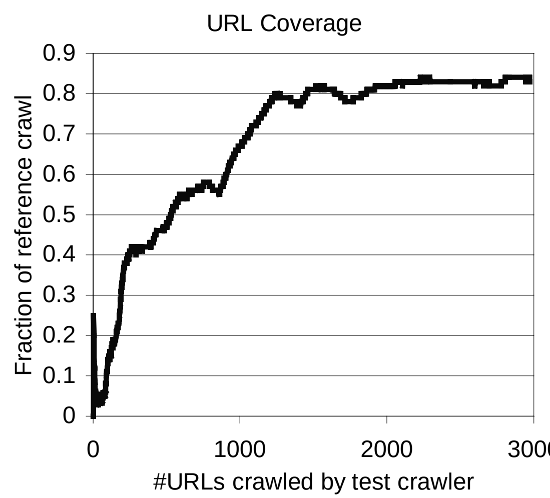


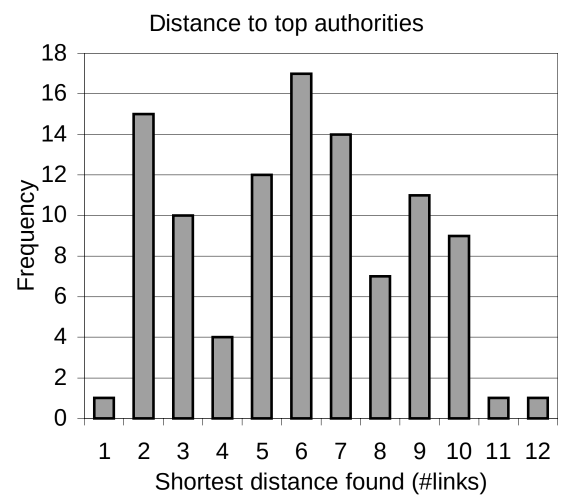
Resource discovery results

- High rate of "harvesting" relevant pages
- Robust to perturbations of starting URLs
- Great resources found 10 links from start set









Database issues

- Useful features
  - + Concurrency and recovery (for crawling)
  - + I/O-efficient representation of mining algorithms
  - + Enables ad-hoc semi-structured queries
- Would help to have
  - – Unlogged tablespaces, flexible choice of recovery
  - – Index (-ed scans) over temporary table expressions
  - – Efficient string storage and operations
  - – Answering multiple queries approximately