Topics in Information Retrieval

hussein suleman
uct cs honours 2016

Topics

- IR Evaluation
- Probabilistic Model
- Language Modeling
- Scalable and Distributed IR
- CLIR
- Recommendations
- Classification

Testbeds

- Used to evaluate IR algorithms.
- Maintained by NIST, TREC, CLEF, etc.
- Contains:
  - document collection (domain-specific, crawled, etc.)
  - set of queries
  - set of relevance judgements
  - optional toolsets
  - optional baseline software

Recall and Precision

- Recall is number of relevant documents that are returned.
- Precision is number of returned documents that are relevant.
- A better IR system may have:
  - better recall
  - better precision
  - better recall and precision

- … but this does not consider rank!

Recall/Precision @N

- Calculate recall/precision at discrete cut-off points in the result set (instead of the entire result set).

Document	Relevance	Precision @N	Recall @N
1	Y	1	1/10
2	Y		
3			
4	Y		
5		3/5	3/10
6			
7	Y		
8	Y		
9			
10		5/10	5/10
… 100	5 more	10/100	10/10

MAP

- Average precision is the average of precision values for recall set to each position in the ranked list.

Document	Relevance	Precision @N
1	Y	1
2	Y	1
3		2/3
4	Y	3/4
5		3/5
Average Precision		0.803

- Mean Average Precision (MAP) is the mean of average precision values for a set of test queries.

Normalized Discounted Cumulative Gain

- Cumulative Gain is the sum of relevance values of each result up to result p.

$$CG_p = \sum_{i=1}^{p} rel_i$$

- Discounted Cumulative Gain gives lower weights to items lower in the list.

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

Normalized Discounted Cumulative Gain

- Normalised Discounted Cumulative Gain (NDCG) normalizes by best possible DCG for comparability of result lists for different queries.

$$NDCG_p = \frac{DCG_p}{IDCG_p}$$

- where IDCG is the DCG for the ideal ranking of results

NDCG Example

- Suppose there are 5 results:
  - D1, D2, D3, D4, D5
- User-assigned relevance scores are :
  - 5, 2, 3, 0, 1
- CG@5 = 5 + 2 + 3 + 0 + 1 = 11
- DCG@5 = 5/log2+2/log3+3/log4+1/log6 = 8.149
- Ideal results would be:
  - D1[5], D3[3], D2[2], D5[1], D4[0]
- IDCG@5 = 5/log2+3/log3+2/log4+1/log4 = 8.323
- NDCG@5 = 8.149/8.323 = 0.979

Language Models

- Statistical/rule-based model of language
- Used in:
  - core IR algorithms like stemming
  - query-likelihood model of retrieval
  - non-independent term model
  - spelling corrections
  - machine translation, tts, speech recognition
  - etc.

Language Models

D1: apple apple apple banana; D2: pear banana; D3: apple

Term	Frequency
apple	4
pear	1
banana	2
...	

word frequency

Term	Frequency
apple apple	2
pear banana	1
apple banana	1
...	

word 2-gram (n-gram,where n=2)

Term	Frequency
ana	4
app	4
ar<space>	1
...	

letter 3-gram

Term	Frequency
apple apple apple	1
apple apple banana	1

word 3-gram

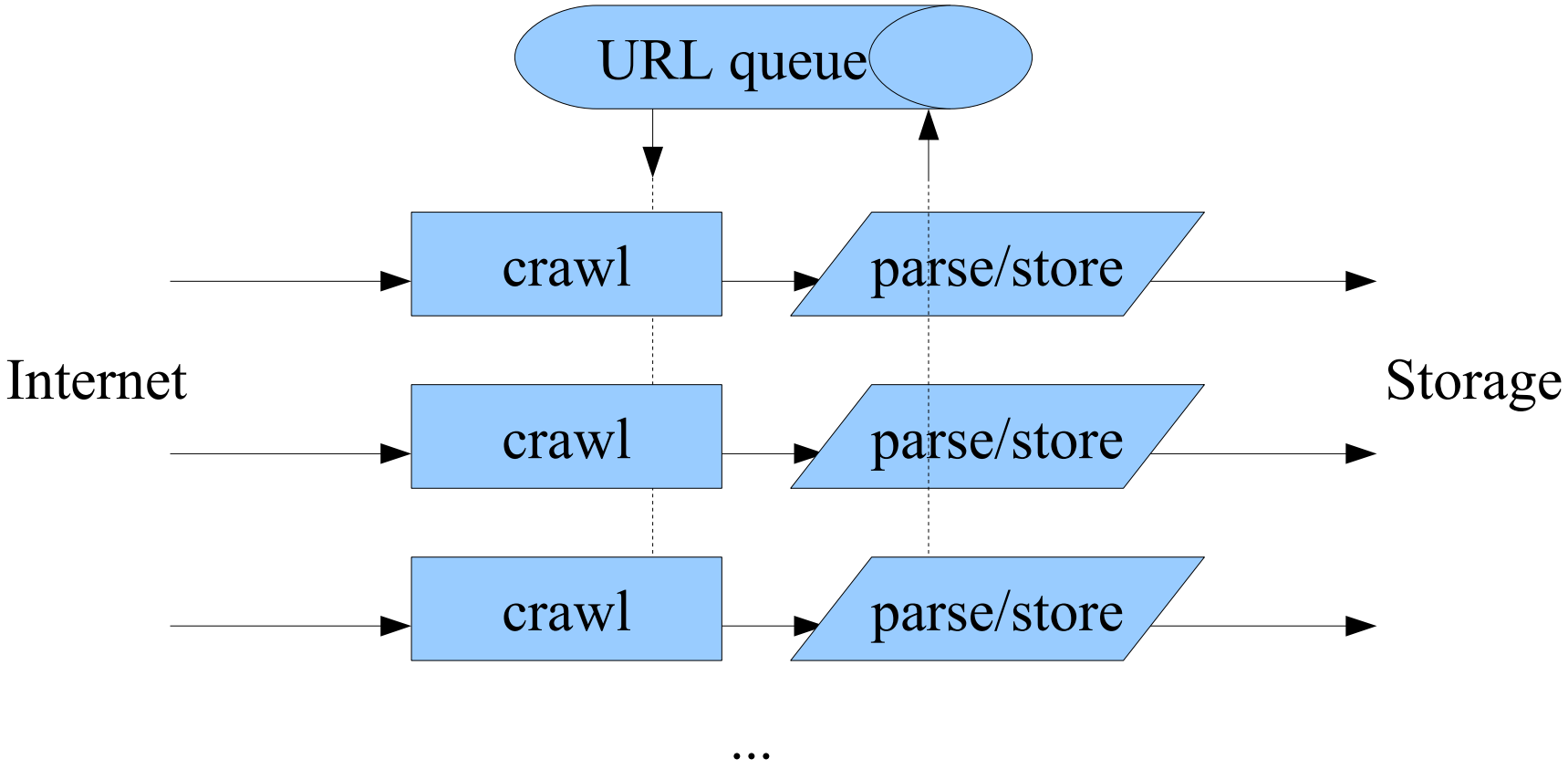
Scalable Search Engines

- Using multiple discrete machines in a cluster/grid/supercomputer configuration.

- 3 Basic problems:
  - Crawling in parallel
  - Indexing in parallel
  - Querying in parallel

Parallel Crawling

- Need a queue of URLs to crawl.
- Processing and networking can be parallelized.



Parallel Indexing

- Term-based or Document-based
- Term: every document indexed by every node.
  - Problem: distribution of documents
- Document: each document indexed on a single node.
  - Problem: merging of term lists
- Term+Document:
  - “sharding”
  - MapReduce generalized algorithm

Parallel Querying

- Many queries means multiple query engine instances.
- How many index copies?
  - Is the index central or distributed?
- How to make querying faster?
- How to update indices?
- Can we generalize architecture for parallel crawling/indexing/querying?

Federated Search

- Instead of one large search engine, how about multiple search engines queried real-time?
- Issues:
  - API/protocol for communication
    - SRU/SRW, Z39.50, etc.
  - Source selection
  - Query languages
  - Result merging
  - Robustness

Cross-Lingual IR

- Searching in one language and obtaining documents in another language.
  - Alternatively, searching in multiple languages and obtaining documents in multiple languages.
- Usually includes some machine translation.
- Why would an English speaker want documents in Mandarin?

CLIR: Translation

- Term translation: documents are translated at indexing time.
- Query translation: query terms are translated at query time.

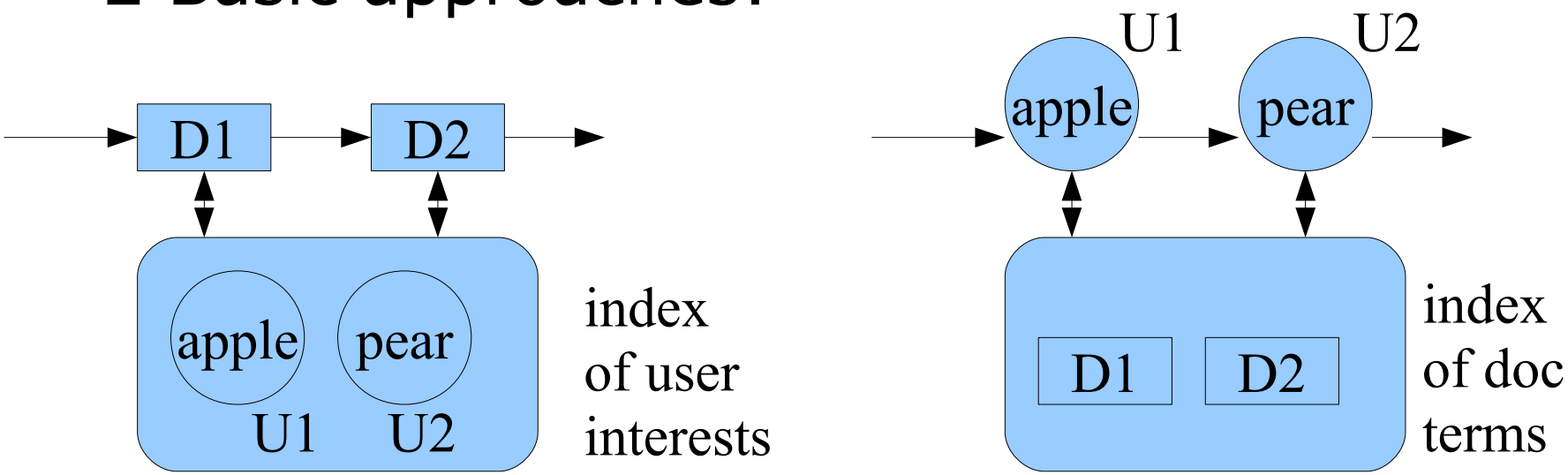
- Issues:
  - Language identification? How?
  - MT of documents in result set?
  - Use pivot languages? Use transliteration/soundex?
  - Language models and algorithms are different!

Multilingual IR

- Basic idea:
  - Search for “apple banana”.
  - Translate query into “яблуко банан apple banana”
  - Get results from English corpus, Ukrainian corpus, language-independent corpus.
  - Do result fusion/merging.
  - Do machine translation.
- Still much exploration in this area ...

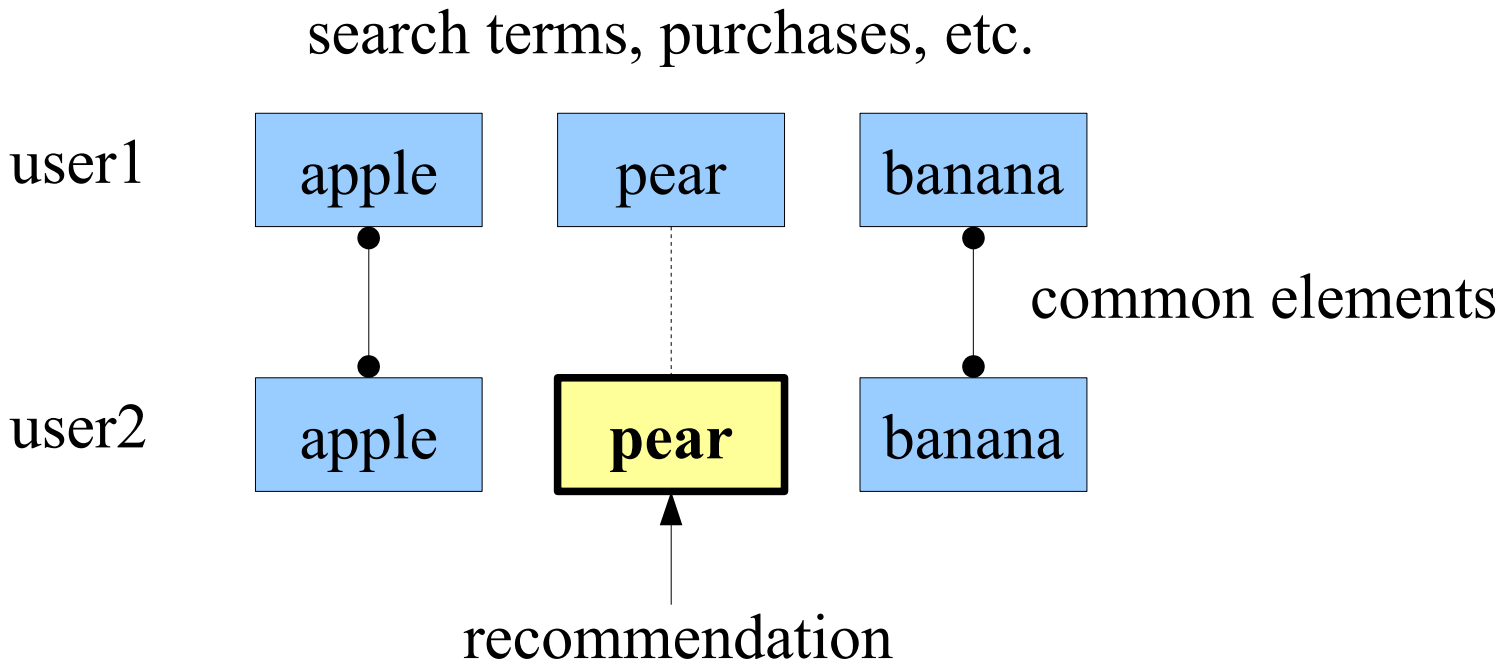
Recommender Systems: SDI

- Selective Dissemination of Information (SDI) is the idea of pushing information to users.
- Every user has a profile of interests (terms).
- 2 Basic approaches:



Recommender Systems: CF

- Collaborative Filtering is about suggesting items based on other common items.



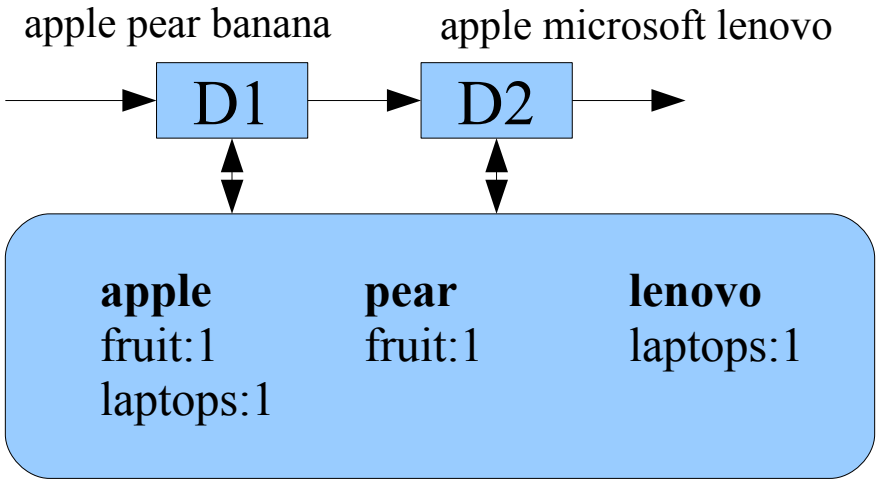
Classification

- Given a set of categories, assign a category automatically to each item.
  - For example: automatically assigning a subject to an electronic thesis.
- Use topic descriptors or sample documents.

topics:

fruit: apple pear

laptops: apple lenovo



index/inverted files of topics

Probabilistic Models

- Considered to be better theoretical basis for IR than ad-hoc models.
- Not as easy to implement, and effect is similar to statistical models.
- Requires some training data!
- Lots of assumptions (like binary independence).

- Probability Ranking Principle:
  - If documents are ordered in decreasing probability of relevance based on available data, then this is the optimal ranking.

Probabilistic Model: Overview

$$P(\mathrm{rel}|d,q) \propto_q \frac{P(\mathrm{rel}|d,q)}{P(\overline{\mathrm{rel}}|d,q)} \tag{2.1}$$

$$= \frac{P(d|\mathrm{rel},q)}{P(d|\overline{\mathrm{rel}},q)} \frac{P(\mathrm{rel}|q)}{P(\overline{\mathrm{rel}}|q)} \tag{2.2}$$

$$\propto_q \frac{P(d|\mathrm{rel},q)}{P(d|\overline{\mathrm{rel}},q)} \tag{2.3}$$

$$\approx \prod_{i \in \mathbf{V}} \frac{P(TF_i = tf_i|\mathrm{rel},q)}{P(TF_i = tf_i|\overline{\mathrm{rel}},q)} \tag{2.4}$$

$$\approx \prod_{i \in \mathbf{q}} \frac{P(TF_i = tf_i|\mathrm{rel})}{P(TF_i = tf_i|\overline{\mathrm{rel}})} \tag{2.5}$$

$$\propto_q \sum_{\mathbf{q}} \log \frac{P(TF_i = tf_i|\mathrm{rel})}{P(TF_i = tf_i|\overline{\mathrm{rel}})} \tag{2.6}$$

from: Robertson and Zaragoza

Probabilistic Model: Overview

$$= \sum_{\mathbf{q}} U_i(tf_i) \tag{2.7}$$

$$= \sum_{\mathbf{q}, tf_i>0} U_i(tf_i) + \sum_{\mathbf{q}, tf_i=0} U_i(0)$$

$$- \sum_{\mathbf{q}, tf_i>0} U_i(0) + \sum_{\mathbf{q}, tf_i>0} U_i(0) \tag{2.8}$$

$$= \sum_{\mathbf{q}, tf_i>0} (U_i(tf_i) - U_i(0)) + \sum_{\mathbf{q}} U_i(0) \tag{2.9}$$

$$\propto_q \sum_{\mathbf{q}, tf_i>0} (U_i(tf_i) - U_i(0)) \tag{2.10}$$

$$= \sum_{\mathbf{q}, tf_i>0} w_i \tag{2.11}$$

from: Robertson and Zaragoza

Probabilistic Model:

$$P(rel|d,q)=\sum_{i=1}^{t}\frac{P(w_i|rel)}{P(w_i|nonrel)}=\sum_{q,tf_i>0}w_i=\sum_{q,tf_i>0}\log\left(\frac{\frac{r+0.5}{(R-r)+0.5}}{\frac{(n-r)+0.5}{(N-n)-(R-r)+0.5}}\right)$$

- where
  - N = number of documents
  - R = number of relevant documents for q
  - n = number of documents with term t
  - r = number of relevant documents with term t

BM25 Ranking

$$Similarity(D,Q)=\sum_{i=1}^{n}\log\left(\frac{N-f_i+0.5}{f_i+0.5}\right)\cdot\left(\frac{tf_i\cdot(k_1+1)}{tf_i+k_1\cdot\left(1-b+b\cdot\frac{|D|}{avgdl}\right)}\right)$$

- where:
  - tf(i) is term frequency of term i in document D
  - f(i) is document frequency of term i
  - k(1) is in [1.2, 2.0]
  - b=0.75
  - |D| is document length
  - avgdl is average document length

References

- Grossman, David, and Ophir Frieder (2004) Information Retrieval: Algorithms and Heuristics, Springer.
- Melucci, Massimo, and Ricardo Baeza-Yates (2011) Advanced Topics in Information Retrieval, Springer.
- Robertson, Stephen, and Hugo Zaragoza (2009) The Probabilistic Relevance Framework: BM25 and Beyond, Foundations and Trends in Information Retrieval, Vol 3, No 4, p.333-389. DOI: 10.1561/1500000019
- Wikipedia (2014) Discounted cumulative gain. http://en.wikipedia.org/wiki/Discounted_cumulative_gain
- Wikipedia (2014) Probabilistic relevance model: BM25. http://en.wikipedia.org/wiki/Probabilistic_relevance_model_%28BM25%29