Distributed Information Retrieval

Fabio Crestani and Ilya Markov

Faculty of Informatics
University of Lugano, Switzerland

Topics covered in this course

Outline

1. Introduction
2. Architectures
3. Broker-Based DIR
4. Evaluation
5. Applications

Topics Covered

1. Introduction
   - What is DIR?
   - Deep Web
   - Federated Search
   - Metasearch
   - Aggregated Search

What is DIR?

- A DIR system is an IR system that is designed to search for information that is distributed across different resources.
- Each *resource* is composed of a search engine and one or more collection of documents. Each resource is assumed to handle the search process on its own collection in an independent way.
- Other names for DIR are: federated search and federated information retrieval.
- Example of DIR systems are: *PubMed*, *FedStats*, *WestLaw*, *Cheshire* etc.

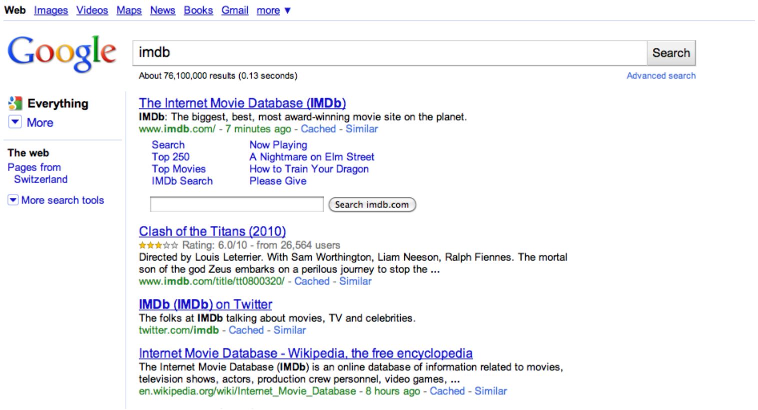
Why do we need DIR?

- There are limits to what a search engines can find on the Web.
  1. Not everything that is on the Web can be crawled or harvested.
  2. The "one size fits all" approach of web search engines has many limitations.
  3. Often there is more than one type of answer to the same query.
- Thus: Deep Web, Federated Search, MetaSearch, Aggregated Search.

Deep Web

- There is a lot of information on the Web that cannot be accessed by search engines (deep or hidden web).
- There are many different reasons why this information is not accessible to crawlers.
- This is often very valuable information!
- All current search engines are able to identify deep web resources.
- Web search engines can only be used to identify a resource (if possible), then a user has to deal directly with it.

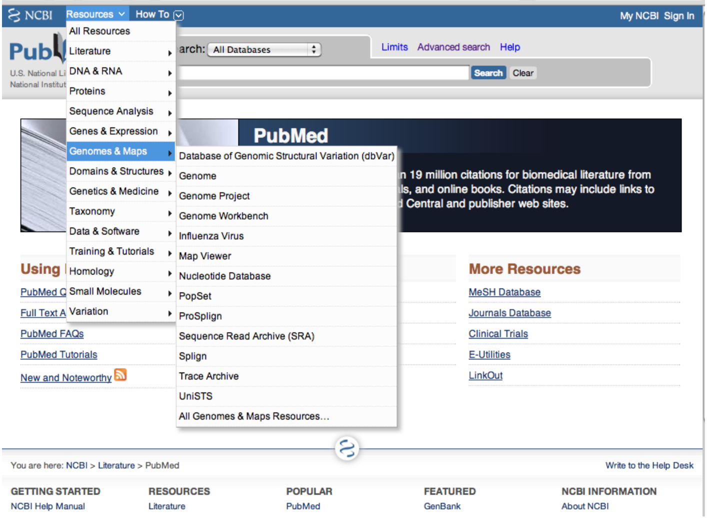
Deep Web: Example



Federated Search

- Federated Search is another name for DIR.
- Federated search systems do not crawl a resource, but pass a user query to the search facilities of a resource itself.
- Why would this be better?
  - Preserves the property rights of the resource owner.
  - Search facilities are optimised to a specific resource.
  - Index is always up-to-date.
  - Resources are curated and of high quality.
- Examples of federated search systems: *PubMed*, *FedStats*, *WestLaw*, *Cheshire* etc.

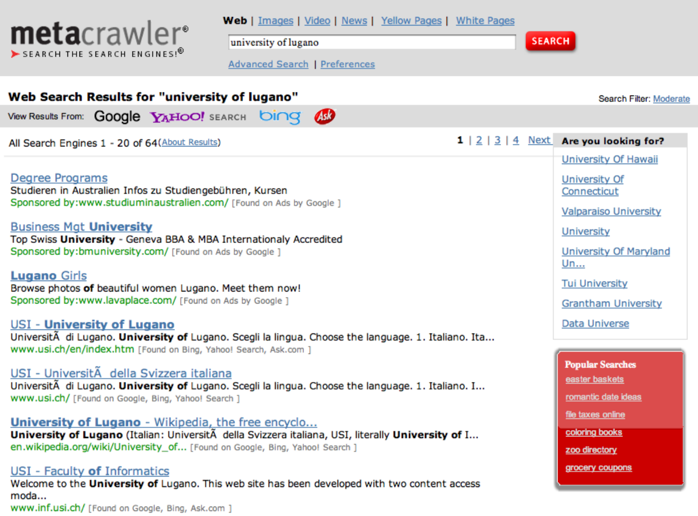
Federated Search: Example



Metasearch

- Even the largest search engine cannot crawl effectively the entire Web.
- Different search engines crawl different portions of the Web.
- Different search engines use different ranking functions.
- Metasearch engines do not crawl the Web, but pass a user query to a number of search engines and then present the fused result list.
- Examples of metasearch systems: *Dogpile*, *MataCrawler*, *AllInOneNews*, and *SavvySearch*.

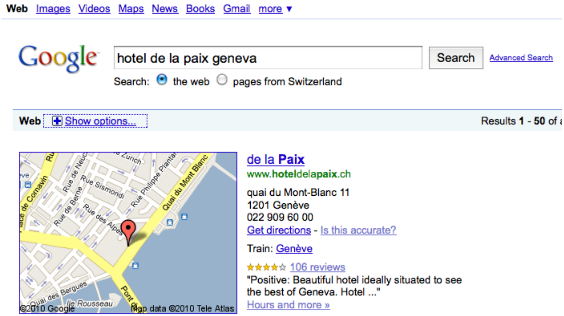
Metasearch: Example



Aggregated Search

- Often there is more than one type of information relevant to a query (e.g. web page, images, map, reviews, etc.).
- These types of information are indexed and ranked by separate sub-systems.
- Presenting this information in an aggregated way is more useful to a user.

Aggregated Search: Example



Questions?

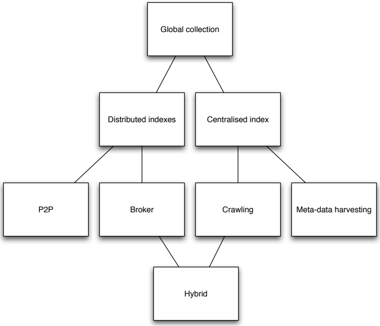
Outline

Topics Covered

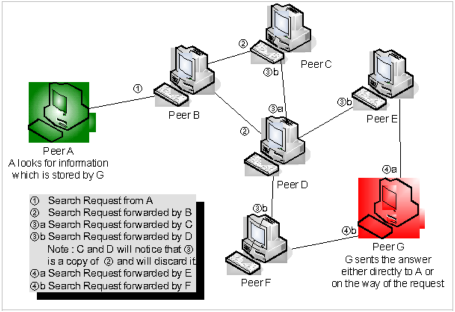
2. Architectures
   - Peer-to-Peer Network
   - Broker-Based Architecture
   - Crawling
   - Metadata Harvesting
   - Hybrid

A Taxonomy of DIR Systems

- A taxonomy of DIR architectures can be build considering where the indices are kept.
- This suggests 4 different types of architectures: broker-based, peer-to-peer, crawling, and meta-data harvesting.

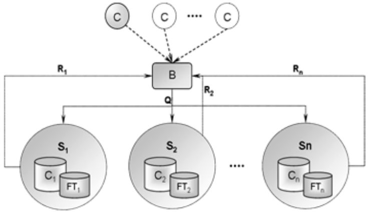


Peer-to-Peer Networks



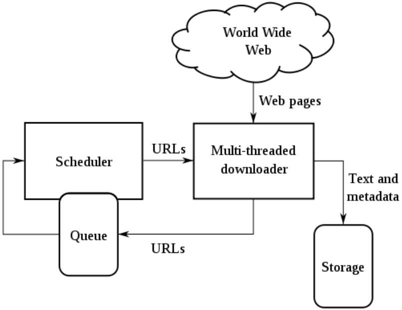
- Indices are located with resources.
- Some parts of indices are distributed to other resources.
- Queries are distributed across resources and results are merged by the peer that originated the query.

Broker-Based Architecture



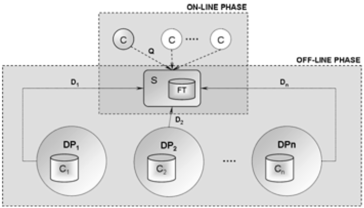
- Indices are located with resources.
- Queries are forwarded to resources and results are merged by a *broker*.

Crawling



- Resources are crawled and documents are harvested.
- Indices are centralized.
- Queries are evaluated out in a centralized way and documents are fetched from resources or from a storage.

Metadata Harvesting

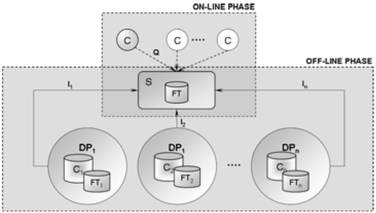


- Indices are located with resources, but metadata is harvested according to some protocol (off-line phase).
- Queries are evaluated at a broker level (on-line phase) to identify relevant documents based on the metadata. The documents are then requested from resources.

The Open Archive Initiative

- The Open Archives Initiative (OAI) develops and promotes interoperability standards that aim to facilitate the efficient dissemination of content.
- The OAI developed a Protocol for Metadata Harvesting (OAI-PMH).
- Only Dublin Core type metadata (or some extension of that set) is exchanged via HTTP in a XML like format.
- OAI has its origin in library world and is very popular in federated digital libraries.

Indexing Harvesting

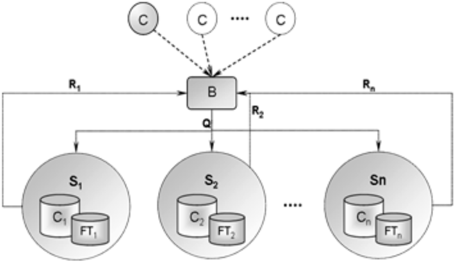


- It is possible to crawl indices, instead of metadata according to some protocol (off-line phase).
- Queries are evaluated out at a broker level (on-line phase) to identify relevant documents based on the documents' full content. The documents are then requested from resources.

Questions?

Outline

Architecture of a Broker-based DIR System



- Indices are located with resources.
- Queries are forwarded to resources and results are merged by a broker.

Phases of the DIR Process

The DIR process is divided in the following phases:

1. Resource discovery
2. Resource description
3. Resource selection
4. Results fusion
5. Results presentation

Topics Covered

3. Broker-Based DIR
   - Resource Discovery
   - Resource Description
   - Resource Selection
   - Results Merging
   - Results Presentation

Resource Discovery

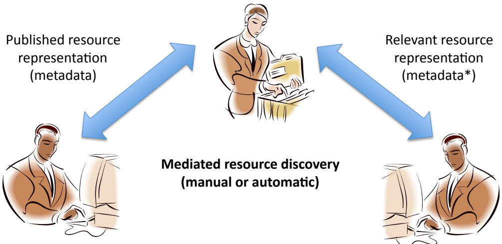
Objectives of the Resource Discovery Phase

The resource discovery phase is concerned with identifying and locating existing resources. These resources might be located because they have a particular property or might be generally relevant to users' interests served by a DIR system.

Obviously, this phase is essential for all other subsequent phases. If we cannot find resources, there is no point in a DIR process.

Mediated Resource Discovery

- Despite its importance not much research has been carried out in DIR on resource discovery.
- Resources are assumed to be already known.
- But no other requirement is expected from known resources (contrary to federated DBMS).
- More generally, we could assume to have a mediated resource discovery.



Automatic Resource Discovery

- Machine-based resource discovery relies on crawling, clustering, and classifying resources discovered on the web automatically.
- Resources are organised with respect to metadata that characterise, for example, their:
  - content (for data sources);
  - semantics (in terms of ontological classes and relationships);
  - characteristics (syntactical properties);
  - performance (with metrics and benchmarks);
  - quality (curation, reliability, trust, ...).
- Resource discovery systems allow the expression of queries to identify and locate resources that are relevant to specific information need.

Resource Discovery

Since this topic has not been studied much in the area of DIR, it will not be presented here.

We will assume that we already know the resources we want to integrate in our DIR system.

However, we will assume that we know little or nothing about them.

Essential Resource Discovery References

B. Yuwono, S.L. Lam, J.H. Ying, D.L. Lee.
A World Wide Web Resource Discovery System.
In *The Fourth International WWW Conference*, Boston, USA, December 11–14, 1995.

M. J. Carman, and C.A. Knoblock.
Learning se- mantic definitions of online information sources
In *Journal of Artificial Intelligence Research*, 30:1–50. 2007

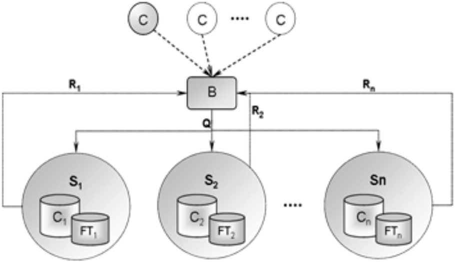
Questions?

Resource Description

Objectives of the Resource Description Phase

The resource description phase is concerned with building a description of each and every resource a broker has to handle.

This phase is required for all other subsequent phases.



Resource Description Outline

1. Resource Description
   - Cooperative Environments: STARTS
   - Un-Cooperative Environments: Query-based Sampling
2. Resource Description Evaluation
3. Resource Description in Un-Cooperative Environments
   - Query Selection
   - Stopping Criteria
4. Other Problems
   - Estimating Collection Size
   - Updating Resource Description

Resource Description Outline

1. Resource Description
    - Cooperative Environments: STARTS
    - Un-Cooperative Environments: Query-based Sampling
2. Resource Description Evaluation
3. Resource Description in Un-Cooperative Environments
    - Query Selection
    - Stopping Criteria
4. Other Problems
    - Estimating Collection Size
    - Updating Resource Description

DIR Cooperation

There are two kinds of environments that determine the way resource description is carried out:

- *Cooperative* environments: a resource provides full access to documents and indices and responds to queries.
- *Uncooperative* environments: a resource does not provide any access to documents and indices; it only respond to queries.

Resource Description in Cooperative Environments

Resource Description in cooperative environments can be very simple as a broker has full access to collection(s) held at a resource.

- A broker could crawl or harvest full collection(s) and deal with queries locally, but this might not be a good idea.
- A resource could provide a broker with information (a description) useful for retrieval.

Stanford Protocol Proposal for Internet and Retrieval Search (STARTS)

STARTS is similar to OAI. For each resource it stores some resource metadata and content summary:

- Query language
- Statistics (term frequency, document frequency, number of documents)
- Score range
- Stopwords list
- Others (sample results, supported fields, etc)

Stanford Protocol Proposal for Internet and Retrieval Search (STARTS)

- STARTS provides a query language with:
  - Filter expressions
  - Ranking expressions
- Retrieved documents are provided by each resource with:
  - Unnormalised score
  - Source indication
- Using the source metadata and content summary a broker can produce a normalised score for each document

Resource Description in Un-Cooperative Environments

- Resource Description in uncooperative environments is far more difficult as a broker does not have access to full collections or metadata and content summary.
- A broker needs to acquire this information without any help from a resource.
  - Important information to acquire for the resource description includes: collection size, term statistics, document scores.
  - **The required information can only be estimated and will contain estimation errors!**

Query-based Sampling

The idea

Query	$\longrightarrow$	Resource
Documents (2-10)	$\longleftarrow$	

Analyse documents
Build description

Resource Description Outline

1. Resource Description
    - Cooperative Environments: STARTS
    - Un-Cooperative Environments: Query-based Sampling
2. Resource Description Evaluation
3. Resource Description in Un-Cooperative Environments
    - Query Selection
    - Stopping Criteria
4. Other Problems
    - Estimating Collection Size
    - Updating Resource Description

Resource Description Evaluation

Vocabulary correspondence - CTF ratio

$$CTF = \frac{\sum_{t \in S_C} ctf_t}{\sum_{t \in C} ctf_t}$$

- CTF Ratio is a proportion of the total terms in a collection that are covered by the terms in its sampled documents.
- Common terms having high *ctf* contribute more than content-bearing terms with low *ctf*.

Resource Description Evaluation

Spearman Rank Correlation Coefficient

$$\rho = 1 - \frac{6\sum(rank_{t,C} - rank_{t,S_C})^2}{n(n^2-1)}, \; n = V_C \cap V_{S_C}$$

- $rank_t$ - the rank of a term t according to its tf.
- The formula used in practice is more complex.

- SRCC measures the correlation between term rankings in a collection C and its description S_C.
- Actual term frequencies are not considered.
- SRCC measures only the intersection in vocabulary between a collection and a description.

Resource Description Evaluation

Kullback-Leibler Divergence

$$KL(\theta_{S_C} || \theta_C) = \sum_{t \in C} P(t|\theta_{S_C}) \log \frac{P(t|\theta_{S_C})}{P(t|\theta_C)}$$

- KL-Divergence measures the distance between the language model of a description θ_{S_C} and the language model of a collection θ_C.
- KL-Divergence has been shown to be more stable and precise.

Resource Description Outline

1. Resource Description
   - Cooperative Environments: STARTS
   - Un-Cooperative Environments: Query-based Sampling
2. Resource Description Evaluation
3. Resource Description in Un-Cooperative Environments
   - Query Selection
   - Stopping Criteria
4. Other Problems
   - Estimating Collection Size
   - Updating Resource Description

Selecting Sampling Queries

Queries can be selected from:

- Other Resource Description (ORD): selects terms from a reference dictionary.
- Learned Resource Description (LRD): selects terms from the retrieved documents based on term statistics.

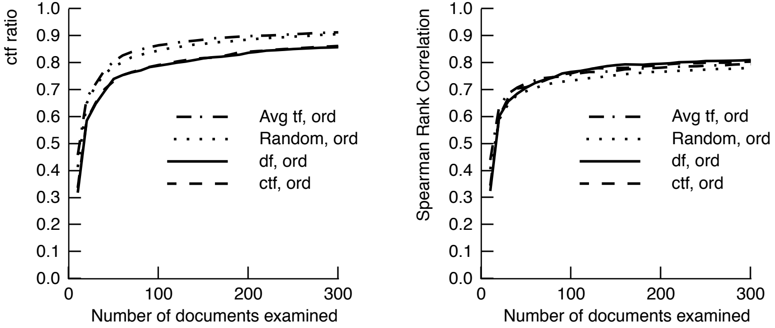
ORD produces more representative samples, but is sensitive to out of vocabulary terms (OOV) that do not return any document.

Selecting Sampling Queries

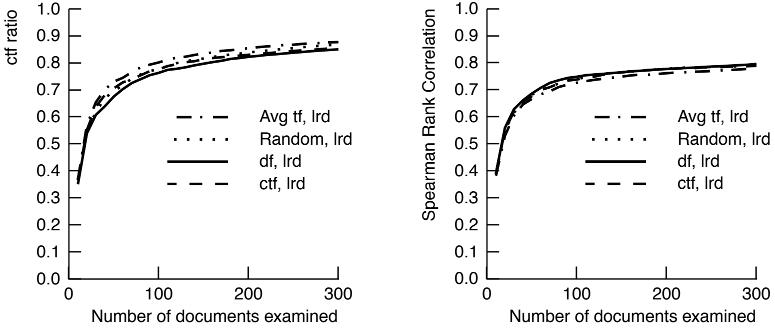
Queries can be selected by:

- Random selection
- Document Frequency (df)
- Collection Frequency (ctf)
- Average Term Frequency (ctf/df)

Selecting Sampling Queries



Selecting Sampling Queries



Stopping Criteria

- Not a well studied problem, mostly approached in a heuristic way.
- Experimental studies suggest to stop after downloading 300-500 unique documents.
  - But this depends on the collection size.
  - Different regions of the resource document space could be unequally sampled.

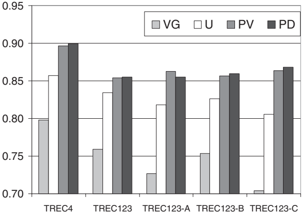
Stopping Criteria

Ideally we would need an adaptive stopping criterium, related to:

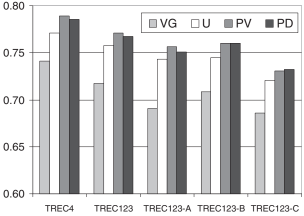
- The proportion of documents sampled in relation of the size of a collection (PD)
- The proportion of terms sampled in relation to the size of a vocabulary (PV)
- Vocabulary growth (VG)

All this needs to be estimated in uncooperative environments!

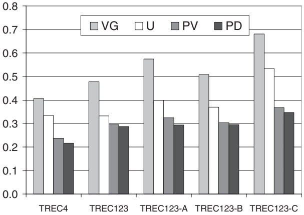
Stopping Criteria



(a) Weighted Common Terms



(b) Term Rankings (Spearman)



(c) JS-Divergence [Lower is Better]

Stopping Criteria

- Q - set of training queries
- θ_k - language model of a sample at k-th iteration

$$p(Q|\theta_k) = \prod_{i=1}^{|Q|} \prod_{j=1}^{|q_i|} p(t = q_{ij}|\theta_k)$$

$$\ell(\theta_k, Q) = \log(p(Q|\theta_k))$$

- Sampling should be stopped if a new sampling iteration does not increase the likelihood substantially

$$\phi_k = \ell(\theta_k, Q) - \ell(\theta_{k-1}, Q) = \log(\frac{p(Q|\theta_k)}{p(Q|\theta_{k-1})}) < \epsilon$$

Stopping Criteria

Aquaint: By-source testbed			
Parameters	$\hat{R}$@10%	$\hat{R}$@20%	Avg. (Total) docs.
QBS-PL	0.212	0.332	501 (56066)
QBS-T $n = 300$	0.179	0.308	300 (36960)
QBS-T $n = 500$	0.191	0.310	500 (56000)
QBS-T $n = 1000$	0.207	0.353	1000 (112000)
Complete	0.249	0.390	11744 (1033461)

Aquaint: By-topic testbed			
Parameters	$\hat{R}$@10%	$\hat{R}$@20%	Avg. (Total) docs.
QBS-PL	0.755	0.856	456 (39685)
QBS-T $n = 300$	0.227	0.495	300 (26400)
QBS-T $n = 500$	0.692	0.808	500 (44000)
QBS-T $n = 1000$	0.733	0.842	1000 (88000)
Complete	0.746	0.854	2262 (1033461)

Figure: Resource selection recall for top 10% and top 20% of all resources.

Resource Description Outline

1. Resource Description
   - Cooperative Environments: STARTS
   - Un-Cooperative Environments: Query-based Sampling
2. Resource Description Evaluation
3. Resource Description in Un-Cooperative Environments
   - Query Selection
   - Stopping Criteria
4. Other Problems
   - Estimating Collection Size
   - Updating Resource Description

Estimating Collection Size

- The size of a collection is an important element of a resource description.
- It is useful for a better stopping criterium of query-based sampling.
- It is also a crucial parameter of the resource selection phase.

Capture-Recapture

Idea

- X - event that a randomly sampled document is already in a sample
- Y - number of X in n trials
- Two samples S_1 and S_2

$$\mathbb{E}[X] = \frac{|S|}{|C|}, \mathbb{E}[Y] = n \cdot \mathbb{E}[X] = n \cdot \frac{|S|}{|C|}$$

$$|S_1 \cap S_2| \approx \frac{|S_1||S_2|}{|C|} \implies |\hat{C}| = \frac{|S_1||S_2|}{|S_1 \cap S_2|}$$

Capture-Recapture

- Take two samples
- Count the number of common documents
- Estimate collection size $|\hat{C}|$

Not very clear how random samples should be generated.

Sample-Resample

Idea

- Randomly pick a term t from a sample
- A - event that some sampled document contains t
- B - event that some documents from the resource contains t

$$P(A) = \frac{df_{t,S}}{|S|},\ P(B) = \frac{df_{t,C}}{|C|}$$

$$P(A) \approx P(B) \implies |\hat{C}| = df_{t,C} \cdot \frac{|S|}{df_{t,S}}$$

Sample-Resample

- Randomly choose a term t from a resource description.
- Send a query t to a resource to estimate $df_{t,C}$
- Repeat several times and estimate collection size $|\hat{C}|$ as an average value of estimates

Assumption that $P(A) \approx P(B)$ is very strong and requires a random sample of a good quality.
Also the method relies on a resource giving the correct document frequency of query terms.

Updating Resource Description

- For many resources the content changes over time.
- Their resource descriptions become outdated.
- It was shown that retrieval accuracy degrades when using an outdated resource description.
- There is a need to keep resource description up-to-date.

Updating Resource Description - General Idea

Idea

- Old resource description
- Use query-based sampling
- Download n documents from a resource
- Add documents to a resource description
- Current resource description

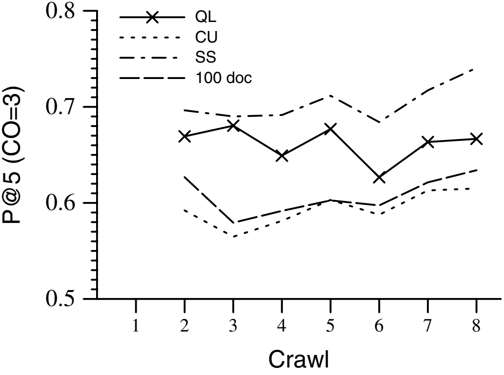
Updating Resource Description - Constraints

- Many resources - M
- Limited bandwidth - N documents can be downloaded at a time jointly from all resources
- n_i - number of documents downloaded form a resource i
- $\sum_{i=1}^{M} n_i = N$

Updating Resource Description - Policies

- $\sum_{i=1}^{M} n_i = N$
- Uniform: $n_i = N \cdot \frac{1}{M}$
- Popularity: $n_i = N \cdot \frac{\rho_i}{\sum_{i=1}^{M} \rho_i}$
- Size: $n_i = N \cdot \frac{S_i}{\sum_{i=1}^{M} S_i}$

Updating Resource Description - Results

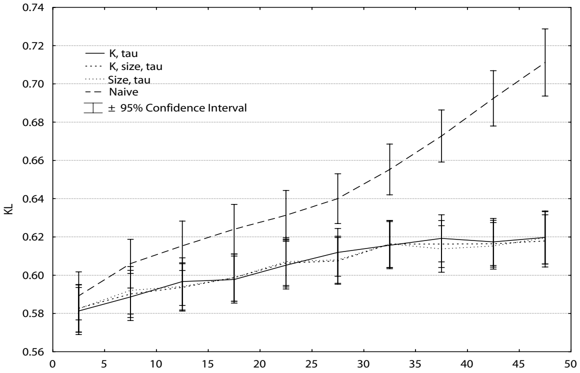


- Precision is stable
- Size-based is the best
- Uniform is the worst

Updating Resource Description - Modeling Content Changes

1. Model
   - Content changes when $KL(R_O||R_C) > \tau$
   - Survival function $S(t) = Pr[T > t]$
   - $S(t)$ depends on linear combination of τ, $\log Size$ and $\Delta KL = \frac{\sum_{t=1}^{t_{train}} KL(S_{t-1}||S_t)}{t_{train}}$
2. Optimality problem
   - $max \sum_{i=1}^{M} S_i(t)$, with the constraint $\sum_{i=1}^{M} n_i = N$
3. Optimal solution with Lagrange-multiplier method

Updating Resource Description - Results



- $KL(S_O||R_C)$ increases
- $KL(S_C||R_C)$ is stable

Essential Resource Description References

STARTS

Luis Gravano, Chen-Chuan K. Chang, Héctor García-Molina, and Andreas Paepcke.
Starts: Stanford proposal for internet meta-searching.
SIGMOD Rec., 26(2):207–218, 1997.

Query-based Sampling

Jamie Callan and Margaret Connell.
Query-based sampling of text databases.
ACM Trans. Inf. Syst., 19(2):97–130, 2001.

Milad Shokouhi, Justin Zobel, Seyed M. M. Tahaghoghi, and Falk Scholer.
Using query logs to establish vocabularies in distributed information retrieval.
Inf. Process. Manage., 43(1):169–180, 2007.

James Caverlee, Ling Liu, and Joonsoo Bae.
Distributed query sampling: a quality-conscious approach.
In *Proceedings of the ACM SIGIR*, pages 340–347, ACM, 2006.

Mark Baillie, Leif Azzopardi, and Fabio Crestani.
An adaptive stopping criteria for query-based sampling of distributed collections.
In *String Processing and Information Retrieval (SPIRE)*, pages 316–328, 2006.

Essential Resource Description References

Estimating collection size

K.-L. Liu, C. Yu, and W. Meng.
Discovering the representative of a search engine.
In *Proceedings of the ACM CIKM*, pages 652–654. ACM, 2002.

Luo Si and Jamie Callan.
Relevant document distribution estimation method for resource selection.
In *Proceedings of the ACM SIGIR*, pages 298–305. ACM, 2003.

M. Shokouhi, J. Zobel, F. Scholer, and S. M. M. Tahaghoghi.
Capturing collection size for distributed non-cooperative retrieval.
In *Proceedings of the ACM SIGIR*, pages 316–323. ACM, 2006.

Updating resource description

M. Shokouhi, M. Baillie, and L. Azzopardi.
Updating collection representations for federated search.
In *Proceedings of the ACM SIGIR*, pages 511–518. ACM, 2007.

P. G. Ipeirotis, A. Ntoulas, J. Cho, and L. Gravano.
Modeling and managing content changes in text databases.
In *ICDE*, pages 606–617, 2005.

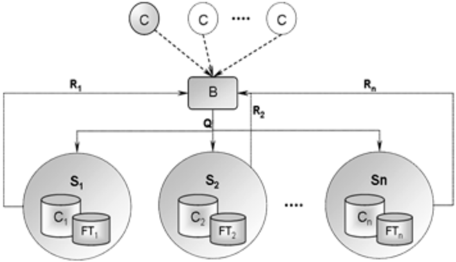
Questions?

Resource Selection

Objectives of the Resource Selection Phase

The resource selection phase is concerned with a broker, given a query, selecting only those resources that are likely to retrieve relevant documents.

Resource selection uses descriptions built on the resource description phase.



Resource Selection Outline

- Main Approaches
  - Theoretical
  - First Generation: cooperative or large document model
  - Second Generation: small document models
- Other Approaches
  - Third Generation: classification-based
  - Classification-aware
- Other Problems
  - Resource Selection Evaluation
  - Resource Selection for Overlapping Collections

Resource Selection Outline

- Main Approaches
  - Theoretical
  - First Generation: cooperative or large document model
  - Second Generation: small document models
- Other Approaches
  - Third Generation: classification-based
  - Classification-aware
- Other Problems
  - Resource Selection Evaluation
  - Resource Selection for Overlapping Collections

Decision-Theoretic Framework (DTF)

Decision-Theoretic Framework works as follows.

1. Models a cost function for each resource.
2. States optimality problem in terms of minimizing the overall cost.
3. Solves it by using Lagrange multipliers method.
4. Selects resources in order to approximate the optimum solution.

Decision-Theoretic Framework - Cost Function

$EC_i(s)$ - the expected cost of retrieving s documents from i-th resource.

$EC_i(s)$ consist of

- $C_i(s)$ – the "physical" cost including connection time, computation costs, charges for delivery, etc.
- rC^+ – the cost of retrieving r relevant documents.
- $(s-r)C^-$ – the cost of retrieving remaining $s-r$ nonrelevant documents.

Expected Cost

$$EC_i(s) = C_i(s) + rC^+ + (s-r)C^-$$

Decision-Theoretic Framework - Cost Function

Number of relevant documents r is not known.
Expected precision $EP_i(s)$ is used instead: $r = sEP_i(s)$.

Expected Cost

$$EC_i(s) = C_i(s) + sEP_i(s)C^+ + s(1 - EP_i(s))C^-$$

Remarks:

- We do not discuss precision estimation methods.
- C^+ and C^- are assumed to be the same for all resources.
- $C^+ \leq 0 \leq C^-$.

Decision-Theoretic Framework - Optimality Problem

Notation:

- l - the number of resources.
- n - the number of documents to be retrieved.
- s_i - the number of documents retrieved from i-th resource.

Optimality Problem

$$EM(n) = \min_{\{s_i\}} \sum_{i=1}^{l} EC_i(s_i)$$

$$\sum_{i=1}^{l} s_i = n$$

Decision-Theoretic Framework - Optimum Solution

Lagrange multipliers method for solving the optimality problem.

$$f(s) = \sum_{\substack{i=1 \\ s_i > 0}}^{l} EC_i(s_i) + \lambda(n - \sum_{\substack{i=1 \\ s_i > 0}}^{l} s_i)$$

$$\frac{\partial f}{\partial s_i} = \frac{\partial EC_i(s_i)}{\partial s_i} - \lambda \stackrel{!}{=} 0$$

For optimum solution $\frac{\partial EC_i(s_i)}{\partial s_i} = \lambda$, i.e. all differentials are equal.

Decision-Theoretic Framework - Resource Selection

Formal solution has to be adjusted for Resource Selection task.

1. $\frac{\partial EC_i(s_i)}{\partial s_i}$ is approximated by $\Delta_i(k) = EC_i(k) - EC_i(k-1)$.
2. Required equality of $\frac{\partial EC_i(s_i)}{\partial s_i}$ is relaxed to approximate equality of $\Delta_i(k)$.
3. Optimum resource selection rule is defined by *uniform vector*

Uniform Vector

$$\forall i \ \Delta_i(s_i) = \begin{bmatrix} \Delta_{max} = \max_i \Delta_i(s_i) \\ \Delta_i(s_i+1) \geq \Delta_{max} \end{bmatrix}$$

Decision-Theoretic Framework - Resource Selection

Uniform Vector

$$\forall i \;\; \Delta_i(s_i) = \begin{bmatrix} \Delta_{max} = \max\limits_i \Delta_i(s_i) \\ \Delta_i(s_i + 1) \geq \Delta_{max} \end{bmatrix}$$

From an i-th resource s_i documents should be selected so that the difference in cost $\Delta_i(s_i)$ is

- either the maximum: $\Delta_i(s_i) = \Delta_{max}$
- or retrieving one more document will make this difference $\Delta_i(s_i + 1)$ larger than the maximum: $\Delta_i(s_i + 1) \geq \Delta_{max}$

Decision-Theoretic Framework - Resource Selection

Theorem 1

For any optimum solution $\{s_i\}$ there exists a uniform vector.

The reverse is not true: not every uniform vector is an optimum solution.

Theorem 2

For cost-monotonic resources any uniform vector is an optimum solution.

Cost-monotonic resource: $\forall s > 0 \;\; \Delta(s) \leq \Delta(s+1)$.

Decision-Theoretic Framework - Resource Selection

For cost-monotonic resources it is enough to find any uniform vector $\{s_i\}$. This will lead to an optimal resource selection in terms of overall cost.

Uniform Vector

$$\forall i \ \Delta_i(s_i) = \left[\begin{array}{l} \Delta_{max} = \max\limits_i \Delta_i(s_i) \\ \Delta_i(s_i + 1) \geq \Delta_{max} \end{array} \right.$$

The actual algorithm for calculating a uniform vector is not discussed here and can be found in the literature.

Decision-Theoretic Framework - Summary

+ DTF shows a way to obtain formally proven optimum solution for the resource selection problem.
+ It incorporates all types of costs in a unified framework.
+ Varying number of documents retrieved per resource.

– It is not obvious how to estimate costs in practice.
– Simple only for cost-monotonic resources.

DTF is just a model. It needs to be implemented.

Decision-Theoretic Framework - Exercise

k	$EC_1(k)$	$\Delta_1(k)$	$EC_2(k)$	$\Delta_2(k)$	Uniform Vector	Opt. Sol.	$EM(k)$
1	6	6	7	7	(1,0)	(1,0)	6
2	10	4	9	2	(0,2), (2,0)	(0,2)	9
3	16	6	14	5	(0,3), (3,0)	(0,3)	14
4	22	6	20	6	(0,4), (1,3), (2,2), (4,0)	(2,2)	19
5	28	6	26	6	(0,5), (1,4), (2,3), (5,0)	(2,3)	24

First Generation Approaches

- Cooperative Environments
  - Collections are ranked according to statistics provided by them.
  - Glossary-of-Servers Server (GlOSS).
- Large Document Model
  - Collections are treated as large single documents.
  - These large documents are ranked with ad-hoc techniques.
  - Collection Retrieval Inference Network (CORI).

Glossary-of-Servers Server (GlOSS)

1. For each resource documents with high similarity to a query are obtained.

$$Rank(q, l, C) = \{d \in C | sim(q, d) > l\}$$

2. Resource's score is calculated based on these documents.

$$Goodness(q, l, C) = \sum_{d \in Rank(q,l,C)} sim(q, d)$$

We assume cooperation and availability of document and term statistics.

Glossary-of-Servers Server (GlOSS) - Exercise

Resource	Similarity between documents and a query q					
C_1	4	13	2	10	7	3
C_2	23	11	6	2	15	8
C_3	4	7	18	21	9	1

$$l = 10$$

	C_1	C_2	C_3
$Goodness(q, l, C)$	23	49	39

Collection Retrieval Inference Network (CORI)

- Resource $\implies$ Large document
- Bayesian inference network on large documents
- Adapted Okapi BM25

$$p(t|C_i) = b + (1-b) \cdot T \cdot I$$

$$T = \frac{df_{t,i}}{df_{t,i} + 50 + 150 \cdot cw_i/avg\_cw}$$

$$I = \frac{\log(\frac{N_c + 0.5}{cf_t})}{\log(N_c + 1.0)}$$

- Resouces are ranked according to $p(Q|C_i)$

Collection Retrieval Inference Network (CORI) - Exercise

$Q =$ "RuSSIR Voronezh", $score_{C_i}(Q) = \sum_{q \in Q} tf_{q,C_i} \cdot idf_q$

Resource	*tf* of "RuSSIR"			*tf* of "Voronezh"		
C_1	2	0	6	1	1	1
C_2	1	4	1	3	2	4
C_3	0	0	0	4	6	2

Term q	tf_{q,C_1}	tf_{q,C_2}	tf_{q,C_3}	idf_q
RuSSIR	8	6	0	1/2
Voronezh	3	9	12	1/3
	C_1	C_2	C_3	
$score_{C_i}(Q)$	5	6	4	

Second Generation Approaches

- Resources are selected based on the number of relevant documents they contain.
- As opposed to the Large Document approach Small Document model retains document boundaries.
- The best known methods are ReDDE, CRCS and SUSHI.

Second Generation Approaches - Main Idea

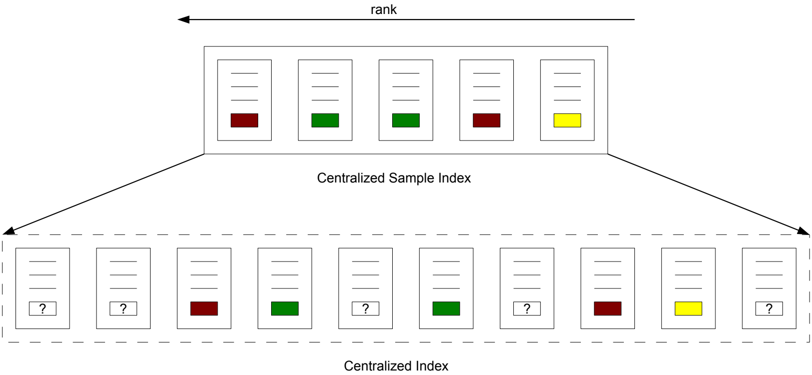
The perfect document ranking can be obtained if all federated resources are merged into one centralized index. This is impossible as we do not have access to their full content.

However we have sampled documents. These sampled documents can be merged into a centralized sample index and ranked according to a user query.

This ranking can be used to estimate the ranking in a hypothetical full centralized index and the number of documents relevant to a query in each resource.

Finally, resources can be ranked according to the estimated number of relevant documents they contain.

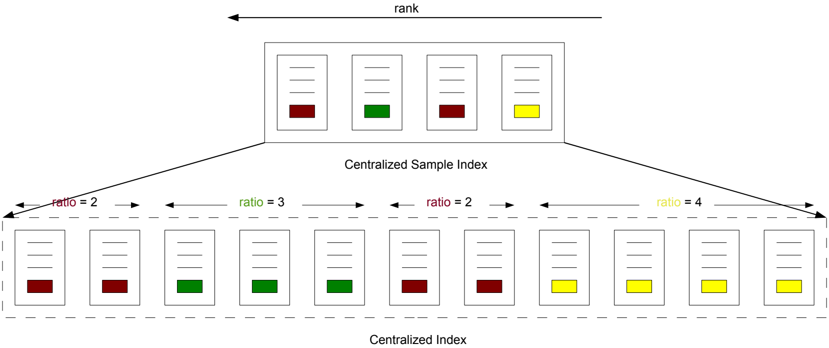
Second Generation Approaches - Main Idea



How to estimate the ranking in a Centralized Index based on the ranking in a Centralized Sample Index?

Relevant Document Distribution Estimation (ReDDE)

If we assume that our sample of documents is random, then for every relevant document in a sample there are $\frac{|C|}{|S_C|}$ similar relevant documents in a resource.



Relevant Document Distribution Estimation (ReDDE)

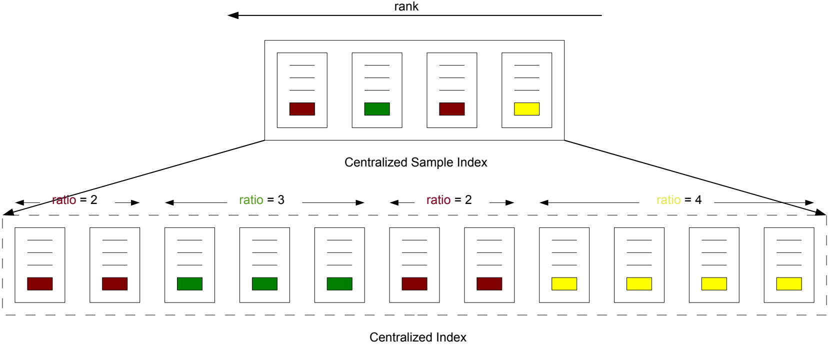
- One sampled document is relevant to a query with a probability of relevance $P(\mathcal{R}|d) \iff \frac{|C|}{|S_C|}$ similar documents in a resource are relevant to a query with the same probability.
- Resource score is estimated as follows.

$$\mathcal{R}(C, Q) \approx \sum_{d \in S_C} P(\mathcal{R}|d) \frac{|C|}{|S_C|}$$

- Probability of relevance $P(\mathcal{R}|d)$ needs to be calculated.

Relevant Document Distribution Estimation (ReDDE)

If a document d_{red} appears before a document d_{green} in a sample ranking, then $\frac{|C_{red}|}{|S_{C_{red}}|} = 2$ documents appear before d_{green} in a centralised ranking.



Relevant Document Distribution Estimation (ReDDE)

- A document d_j appears before a document d_i in a sample ranking $\iff \frac{|C_j|}{|S_{C_j}|}$ documents appear before d_i in a centralised ranking.
- Centralized rank of a document is estimated as follows.

$$Rank_{centralized}(d_i) = \sum_{d_j : Rank_{sample}(d_j) < Rank_{sample}(d_i)} \frac{|C_j|}{|S_{C_j}|}$$

- The probability of relevance $P(\mathcal{R}|d)$ is estimated as follows.

$$P(\mathcal{R}|d) = \begin{cases} \alpha & \text{if } Rank_{centralized}(d) < \beta \cdot \sum_i |C_i| \\ 0 & \text{otherwise.} \end{cases}$$

Relevant Document Distribution Estimation (ReDDE)

$$\mathcal{R}(C, Q) \approx \sum_{d \in S_C} P(\mathcal{R}|d) \frac{|C|}{|S_C|}$$

$$P(\mathcal{R}|d) = \begin{cases} \alpha & \text{if } Rank_{centralized}(d) < \beta \cdot \sum_i |C_i| \\ 0 & \text{otherwise.} \end{cases}$$

$$Rank_{centralized}(d_i) = \sum_{d_j : Rank_{sample}(d_j) < Rank_{sample}(d_i)} \frac{|C_j|}{|S_{C_j}|}$$

α is a constant positive probability of relevance and β is a percentage threshold separating relevant from non-relevant documents.

ReDDE - Exercise

	C_1	C_2	C_3
$\lvert C_i \rvert$	9000	25000	15000
$\lvert S_{C_i} \rvert$	300	500	300
$\lvert C_i \rvert / \lvert S_{C_i} \rvert$	30	50	50

$$\beta = \frac{1}{500},\ \alpha = 1 \implies Rank_{centralized(d)} < 98$$

Rank	1	2	3	4	5	6
Sample rank	d_{C_2}	d_{C_1}	d_{C_1}	d_{C_3}	d_{C_2}	d_{C_3}
Centr. rank	0	50	80	110	160	210
$P(\mathcal{R}\|d)$	1	1	1	0	0	0

	C_1	C_2	C_3
$\mathcal{R}(C, Q)$	60	50	0

Centralised-Rank Collection Selection (CRCS)

ReDDE assigns a constant probability of relevance $P(\mathcal{R}|d) = \alpha$ to all the top ranked documents.

CRCS instead defines the probability of relevance proportionally to the document rank with the following strategies.

- Linear

$$P(\mathcal{R}|d) = \frac{\gamma - Rank_{sample}(d)}{|C_{max}|}$$

- Exponential

$$P(\mathcal{R}|d) = \frac{\alpha \exp(-\beta Rank_{sample}(d))}{|C_{max}|}$$

Also centralized rank of a document is not used.

Centralised-Rank Collection Selection (CRCS) - Exercise

	C_1	C_2	C_3
$\|C_i\|$	9000	25000	15000
$\|S_{C_i}\|$	300	500	300
$\|C_i\|/\|S_{C_i}\|$	30	50	50

Linear CRCS, $\gamma = 5$

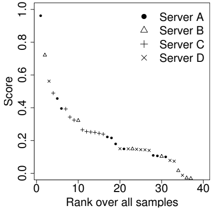
Rank	1	2	3	4	5	6
Sample ranking	d_{C_2}	d_{C_1}	d_{C_1}	d_{C_3}	d_{C_2}	d_{C_3}
$P(\mathcal{R}\|d)$	$4/25000$	$3/25000$	$2/25000$	$1/25000$	0	0

	C_1	C_2	C_3
$\mathcal{R}(C, Q)$	$3/500$	$4/500$	$1/500$

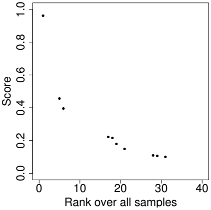
Scoring Scaled Samples for Server Selection (SUSHI)

1. Similarly to ReDDE and CRCS, SUSHI scores each sampled document with regard to a query.
2. The sampled documents for each resource are extracted from the ranking.
3. The document scores are adjusted: $(score + 0.5)\frac{|C|}{|S_C|}$.
4. A curve is fitted to the reranked sampled documents.
5. The rank of an unseen document is estimated by the fitted curve.
6. Top document scores across all resources are calculated by sorting the estimated scores.

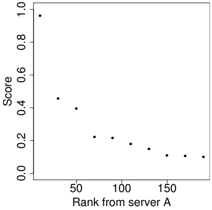
Scoring Scaled Samples for Server Selection (SUSHI)



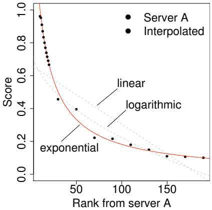
(a) All sampled documents scored



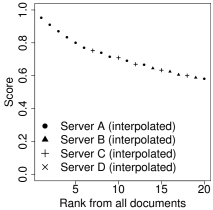
(b) One server extracted



(c) Re-ranked and scaled



(d) Curves fitted, best fit chosen, and scores interpolated



(e) Interpolated scores for all servers re-merged

Scoring Scaled Samples for Server Selection (SUSHI)

SUSHI achieves the performance comparable to ReDDE and CRCS while selecting less resources.

Resource Selection Outline

- Main Approaches
  - Theoretical
  - First Generation: cooperative or large document model
  - Second Generation: small document models
- Other Approaches
  - Third Generation: classification-based
  - Classification-aware
- Other Problems
  - Resource Selection Evaluation
  - Resource Selection for Overlapping Collections

Third Generation Approaches

Classification-based approaches.

Come from Vertical Selection.

Will be discussed in Applications section.

Classification-aware Resource Selection

Completely different approach by Panagiotis G. Ipeirotis and Luis Gravano.

Classifies resources into hierarchical structure of topics similar to the Web Directories.

Uses Focused Query-Probing instead of Query-based Sampling.

Considers topical similarity when selecting resources.

Will not be discussed here. More details can be found in the literature.

Resource Selection Outline

- Main Approaches
  - Theoretical
  - First Generation: cooperative or large document model
  - Second Generation: small document models
- Other Approaches
  - Third Generation: classification-based
  - Classification-aware
- Other Problems
  - Resource Selection Evaluation
  - Resource Selection for Overlapping Collections

Resource Selection Evaluation

Recall

$$Recall = R_k = \frac{\sum_{i=1}^{k} \Omega_i}{\sum_{i=1}^{k} O_i}$$

- k - the number of resources selected.
- $\sum_{i=1}^{k} \Omega_i$ - the total number of relevant documents in selected resources.
- $\sum_{i=1}^{k} O_i$ - the total number of relevant documents if the selection is optimal.

Resource Selection Evaluation

Mean Square Error

$$MSE = \frac{1}{N_C} \sum_{i \in C} (O_i - \Omega_i)^2$$

- N_C - the number of resources.
- MSE measures the mean squared error between the optimal resource ranking $\{O_i\}$ and the ranking obtained by resource selection $\{\Omega_i\}$.

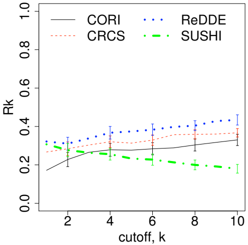
Resource Selection Evaluation

Spearman Rank Correlation Coefficient

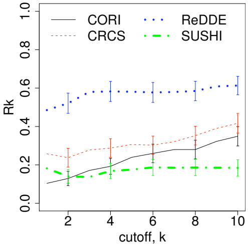
$$\rho = 1 - \frac{6 \sum_{i=1}^{N_C} (O_i - \Omega_i)^2}{N_C (N_C^2 - 1)}$$

The same idea as MSE: SRCC measures the difference between optimal and selected rankings.

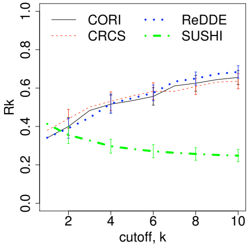
Comparison of Resource Selection Techniques



(d) $\mathcal{R}_k$, uniform testbed; the nonrelevant testbed is fairly similar

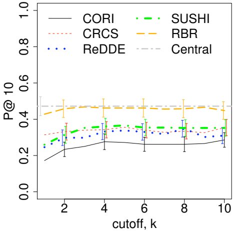


(e) $\mathcal{R}_k$, relevant testbed; the representative testbed is fairly similar

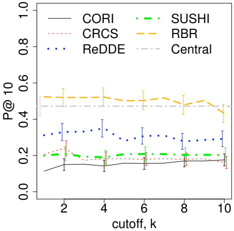


(f) $\mathcal{R}_k$, k-means testbed

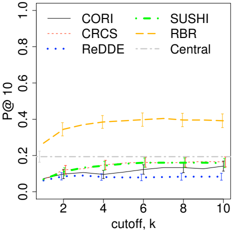
Comparison of Resource Selection Techniques



(a) P@10, uniform testbed; the nonrelevant, representative, and k-means testbeds are similar



(b) P@10, relevant testbed



(c) P@10, web testbed

- "RBR" ranks servers according to the number of relevant documents they hold.
- "Central" uses centralized index.

Overlapping Collections - Overlap Estimate

Given

- Collections C_1 and C_2
- K overlap documents between them
- Samples S_1 and S_2
- D duplicate documents within them

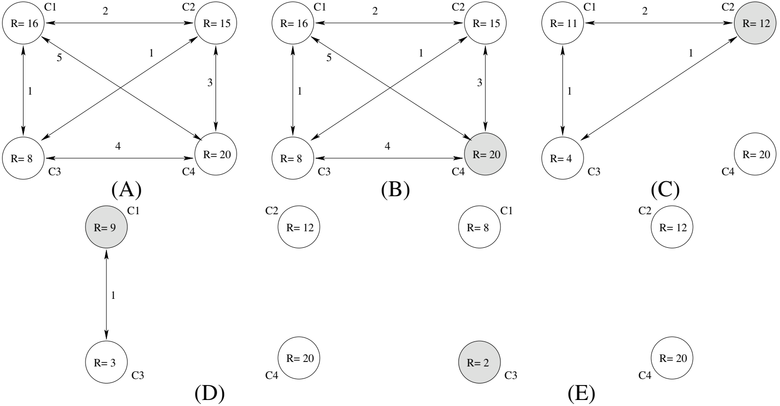
Estimated number of common documents

$$\hat{K} = \frac{|C_1||C_2| \cdot D}{|S_1||S_2|}$$

Overlapping Collections - Relax Algorithm

1. The number of overlapped relevant documents between each pair of resources are estimated.
2. The federated environment is represented by a graph, where each vertex is a resource and the weight of each edge is computed using the number of common relevant documents between the connected pairs.
3. The resource with the highest estimated number of relevant documents is selected.
4. The graph is updated by relaxing all resources and removing unnecessary edges.
5. Repeat until there are no more edges or enough resources are chosen.

Overlapping Collections - Relax Algorithm



Essential Resource Selection References

Theoretic approaches

N. Fuhr.
A decision-theoretic approach to database selection in networked ir.
ACM Trans. Inf. Syst., 17(3):229–249, 1999.

First generation approaches

James P. Callan, Zhihong Lu, and W. Bruce Croft.
Searching distributed collections with inference networks.
In *Proceedings of the ACM SIGIR*, pages 21–28. ACM, 1995.

Luis Gravano, Héctor García-Molina, and Anthony Tomasic.
GlOSS: text-source discovery over the internet.
ACM Trans. Database Syst., 24(2):229–264, 1999.

Essential Resource Selection References

Second generation approaches

Luo Si and Jamie Callan.
Relevant document distribution estimation method for resource selection.
In *Proceedings of the ACM SIGIR*, pages 298–305. ACM, 2003.

Milad Shokouhi.
Central-rank-based collection selection in uncooperative distributed information retrieval.
In *ECIR*, pages 160–172, 2007.

P. Thomas and M. Shokouhi.
Sushi: scoring scaled samples for server selection.
In *Proceedings of the ACM SIGIR*, pages 419–426. ACM, 2009.

Resource Selection for Overlapping Collections

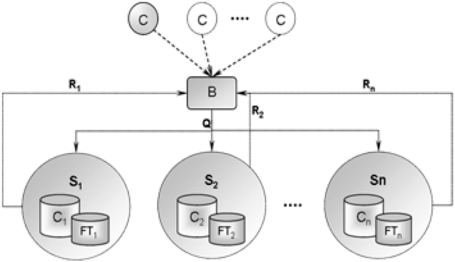
Milad Shokouhi and Justin Zobel.
Federated text retrieval from uncooperative overlapped collections.
In *Proceedings of the ACM SIGIR*, pages 495–502. ACM, 2007.

Questions?

Results Merging

Objectives of the Results Merging Phase

The results merging phase is concerned with merging the list of top-ranked documents retrieved from selected resources and returning a fused list to a user.



Not to be confused with *data fusion*, where results come from a single resource and are then ranked by multiple retrieval models.

Results Merging Outline

- Approaches
  - Merging with CORI collection scores (CORI)
  - Semi-supervised learning (SSL)
  - Sample-agglomerate fitting estimate (SAFE)
- Other Problems
  - Results Merging Evaluation
  - Results Merging for Overlapping Collections

Results Merging Outline

- Approaches
  - Merging with CORI collection scores (CORI)
  - Semi-supervised learning (SSL)
  - Sample-agglomerate fitting estimate (SAFE)
- Other Problems
  - Results Merging Evaluation
  - Results Merging for Overlapping Collections

Results Merging Issues

- In uncooperative environments resources might provide scores
  - But a broker does not have any information on how these score are computed.
  - Score normalisation requires some way of comparing scores.
- Alternatively resources might provide only rank positions
  - But a broker does not have any information on the relevance of each document in rank lists.
  - Merging the ranks requires some way of comparing rank positions.
- The main idea of results merging algorithms is to derive a centralized score of a retrieved document based on its resource specific score or rank.

Collection Retrieval Inference Network (CORI)

Idea

Linear combination of the score of a resource and the score of a document.

Normalised scores

- Normalized collection score: $C_i' = \frac{(C_i - C_{min})}{(C_{max} - C_{min})}$
- Normalized document score: $D_j' = \frac{(D_j - D_{min})}{(D_{max} - D_{min})}$
- Heuristic linear combination: $D_j'' = \frac{D_j' + 0.4 \times D_j' \times C_i'}{1.4}$

Collection Retrieval Inference Network (CORI)

+ Simple to implement.

– Implicitly assumes that resources use identical retrieval models and resource specific document scores are relatively similar.

Semi-Supervised Learning (SSL)

Idea

Learn to map resource specific document scores into centralized scores.

Semi-Supervised Learning (SSL) - Basic Algorithm

1. Rank documents in a centralized sample index. Let a document d_j from a j-th resource has a centralized score $D'_{i,j}$.
2. Retrieve documents from selected resources. Let a document d_j from a j-th resource has a resource specific score $D_{i,j}$.
3. Find documents that appear **both in a centralized sample index and in the retrieved results**. Thus we have pairs of corresponding document scores $D'_{i,j}$ and $D_{i,j}$.
4. Use these known pairs to train a regression model.
5. Use this trained model to estimate centralized scores of other retrieved documents.

Semi-Supervised Learning (SSL) - Cases

1. Resources use identical retrieval models.
2. Resources use different retrieval models.

SSL with Identical Retrieval Models

Idea

Resources use identical retrieval models $\Longrightarrow$ resource specific document scores are relatively similar $\Longrightarrow$ CORI-like approach can be used.

Model

$$D'_{i,j} = a \times D_{i,j} + b \times D_{i,j} \times C_i$$

Training

$$\begin{bmatrix} D_{1,1} & C_1 D_{1,1} \\ D_{1,2} & C_1 D_{1,2} \\ \dots & \dots \\ D_{n,m} & C_n D_{n,m} \end{bmatrix} \times \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} D'_{1,1} \\ D'_{1,2} \\ \dots \\ D'_{n,m} \end{bmatrix}$$

SSL with Different Retrieval Models

Idea

Different regression models are trained for different resource.

Model

$$D'_{i,j} = a_i \times D_{i,j} + b_i$$

Training

$$\begin{bmatrix} D_{1,1} & 1 \\ D_{1,2} & 1 \\ \dots & \dots \\ D_{n,m} & 1 \end{bmatrix} \times \begin{bmatrix} a_i \\ b_i \end{bmatrix} = \begin{bmatrix} D'_{1,1} \\ D'_{1,2} \\ \dots \\ D'_{n,m} \end{bmatrix}$$

Semi-Supervised Learning (SSL)

+ Allows different retrieval models by different resources.
+ Trains parameters instead of choosing them empirically.

– Assumes that resources return document **scores**.

Sample-Agglomerate Fitting Estimate (SAFE)

Idea

The same idea as SSL but uses estimated document rank instead of the score.

Model

$$D'_{i,j} = a_i \times f(\widehat{Rank}_{D_{i,j}}) + b_i$$

where f is a transformation function.

Sample-Agglomerate Fitting Estimate (SAFE)

- Centralized rank of a document $\widehat{Rank}_{D_{i,j}}$ is estimated in a ReDDE manner.
- A document d_k appears before a document d_l in a sample ranking $\iff \frac{|C_k|}{|S_{C_k}|}$ documents appear before d_l in a centralised ranking.
- Centralized rank of a document is estimated as follows.

Estimated Centralized Rank

$$\widehat{Rank}_{centralized}(d_l) = \sum_{d_k : Rank_{sample}(d_k) < Rank_{sample}(d_l)} \frac{|C_k|}{|S_{C_k}|}$$

Sample-Agglomerate Fitting Estimate (SAFE)

+ Uses document ranks instead of scores.

– Still needs training data - documents that appear both in a centralized sample index and in the retrieved results.

Results Merging Outline

- Approaches
  - Merging with CORI collection scores (CORI)
  - Semi-supervised learning (SSL)
  - Sample-agglomerate fitting estimate (SAFE)
- Other Problems
  - Results Merging Evaluation
  - Results Merging for Overlapping Collections

Results Merging Evaluation

P@N

Comparison of Results Merging Techniques

Document Rank	Trec123 Testbed		Trec4_kmeans Testbed	
	CORI Merge	SSL Merge 700 sampled documents	CORI Merge	SSL Merge 700 sampled documents
5	0.3280	0.3880 (+18.3%)	0.2600	0.3800 (+46.2%)
10	0.3400	0.3640 (+7.1%)	0.2160	0.3320 (+53.7%)
15	0.3360	0.3520 (+4.8%)	0.1947	0.3107 (+59.6%)
20	0.3260	0.3420 (+4.9%)	0.1850	0.2880 (+55.7%)
30	0.3100	0.3133 (+1.1%)	0.1700	0.2587 (+52.5%)

Note: Ten databases were selected to search for each query. Results are averaged over 50 queries.

Comparison of Results Merging Techniques

Rank	Uniform testbed			Relevant testbed		
	SSL (single)	SSL (multi)	SAFE	SSL (single)	SSL (multi)	SAFE
5	0.33	0.33	0.35	0.31	0.32	0.26
10	0.34	0.33	0.34	0.28	0.29	0.23

Table: P@N for Uniform and Relevant testbeds when selecting 5 resources.

Results Merging for Overlapping Collections

There are two ways of dealing with duplicate documents on results merging phase.

1. Remove duplicates from the final result list.
2. Give higher score to a document appeared more than in one result list.

Results Merging for Overlapping Collections

To remove duplicates from the final result list any near-duplicate detection technique can be used.

- document similarity measures
- shingles
- grainy hash vectors
- etc.

Results Merging for Overlapping Collections

If a document d appears in m collections with scores $\{s_i\}$, this information can be leveraged to calculate the final document score with the following methods.

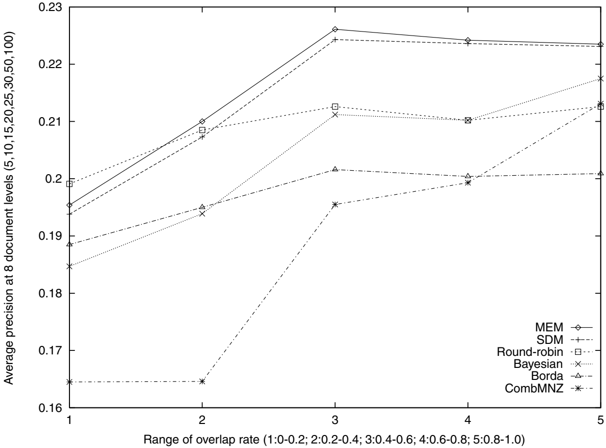
- Shadow Document: assumes that d also appears in $n - m$ collections with a score $\frac{\sum_{i=1}^{m} s_i}{m}$.

$$score(d) = \sum_{i=1}^{m} s_i + k(n - m)\frac{\sum_{i=1}^{m} s_i}{m}$$

- Multi-Evidence

$$score(d) = f(m)\frac{\sum_{i=1}^{m} s_i}{m},\ f(x) \text{ is a nondecreasing function}$$

Results Merging for Overlapping Collections



Essential Results Merging References

Results Merging Approaches

James P. Callan, Zhihong Lu, and W. Bruce Croft.
Searching distributed collections with inference networks.
In *Proceedings of the ACM SIGIR*, pages 21–28. ACM, 1995.

L. Si and J. Callan
A semisupervised learning method to merge search engine results
In *ACM Transactions on Information Systems*, 21: 457–491, 2003.

M. Shokouhi and J. Zobel.
Robust result merging using sample-based score estimates.
In *ACM Transactions on Information Systems*, 27(3): 1–29, 2009.

Results Merging for Overlapping Resources

Yaniv Bernstein, Milad Shokouhi, and Justin Zobel.
Compact features for detection of near-duplicates in distributed retrieval.
In *SPIRE*, pages 110–121, 2006.

Shengli Wu and Sally McClean.
Result merging methods in distributed information retrieval with overlapping databases.
Inf. Retr., 10(3):297–319, 2007.

S. Wu and F. Crestani.
Shadow document methods of results merging.
In *Proceedings of the ACM SAC*, pages 1067–1072, 2004

Questions?

Results Presentation

Objectives of the Results Presentation Phase

The main objective of this phase of to presents the results of the DIR results merging phase.

It is the final phase of the DIR, but the one that might impact the most of user satisfaction with the DIR system. Thus, results needs to be presented in the most appropriate way.

Despite a clear recognition of its importance, there is not a lot of research on this topic.

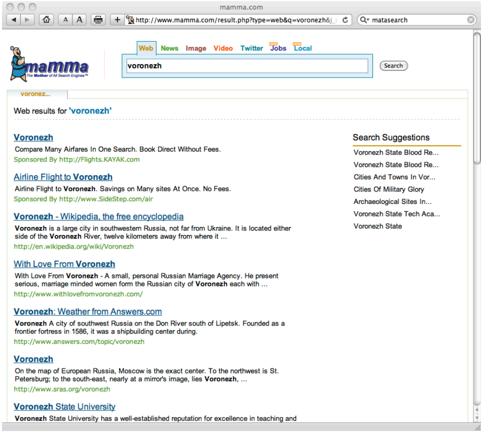
The Main Question

The main questions is: how should we present the results?

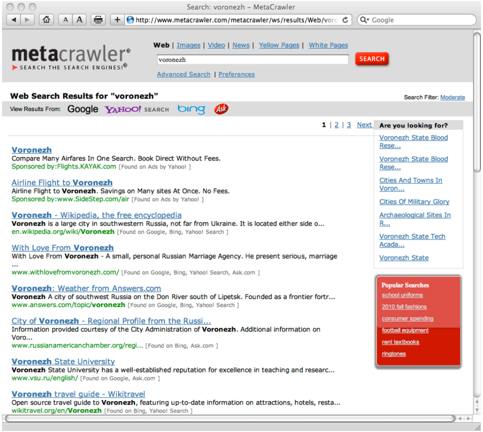
In fact, contrary to popular belief, there are several options ...

For example ...

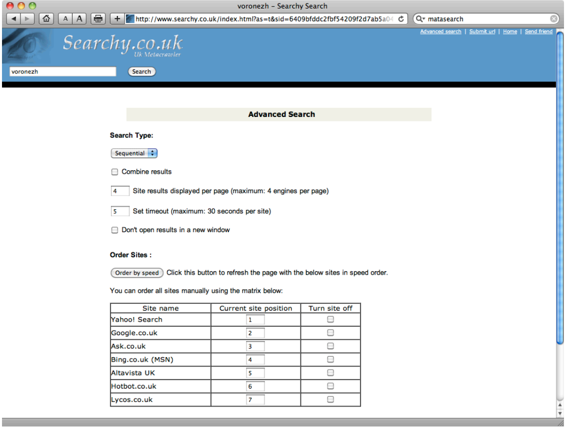
The Merged List Interface



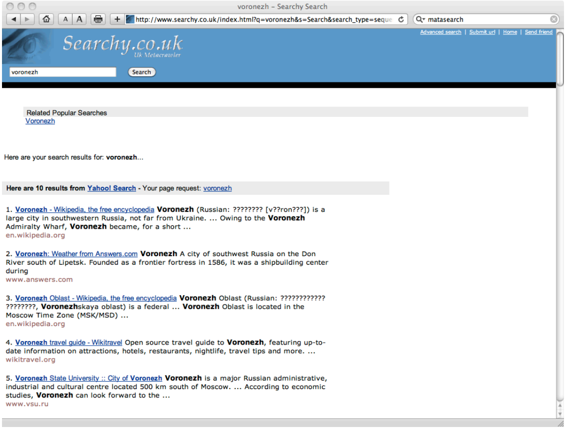
The Tabbed Interface



The Separated-Results Interface



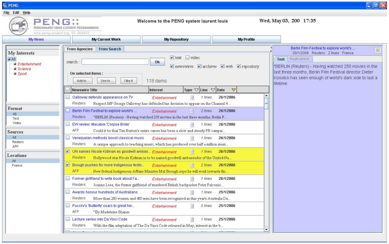
The Separated-Results Interface (cont.)



Which Interface Should We Choose?

Which one would you choose!

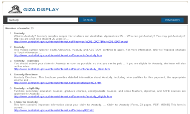
The Project PENG Experience

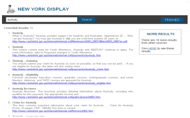


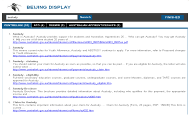
- PENGS stands for Personalise News Content Programming, and EU project in FP6.
- Professionals do not want to repeat search on different sources: they love DIR.
- Professionals want freedom to choose source, media type, format, location, and would like to express preferences in relation to timeliness, trust, etc.

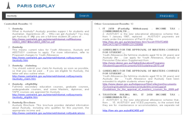
An Experimental Evaluation

- The most interesting experimental study on results presentation was carried out by Paul Thomas.
- Paul Thomas compared 4 interfaces for results presentation in DIR in a well designed user study.
- The 4 interfaces were: merged, more results, tabbed, and side-by-side.



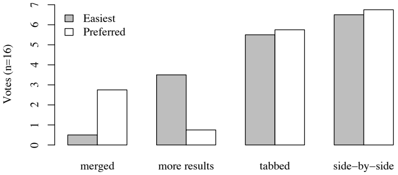






An Experimental Evaluation (cont.)

- Paul Thomas's evaluation showed that there was no significant difference in the effectiveness the task were carried out: the interface did not have a significant effect on such tasks and with such users.
- However, when asked which interface users preferred the results were almost surprising:

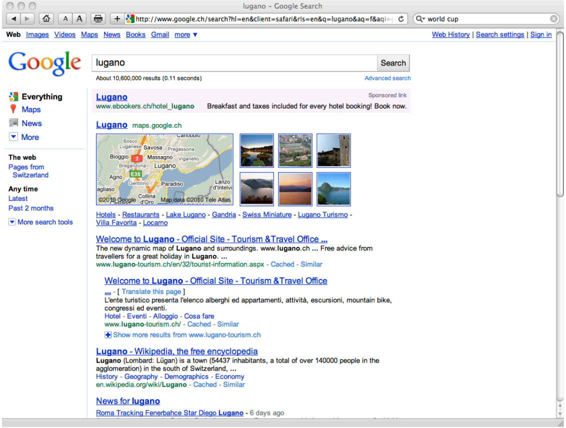


General Guidelines for Results Presentation in DIR

- The results of the PENG project and Thomas's evaluation were in agreement in suggestion the following guidelines:
  - The interface should enable the widest freedom for the user to choose sources and other document characteristics.
  - The interface should immediately expose enough information to let the user decide where to look next, although a gradual disclosure seems a better option.
  - The interface should require the fewer the number of actions by the user.

Results Presentation for Aggregated Search

Results presentation in aggregated search is another matter. There are far fewer option on how to present results.



Essential Results Presentation References

P. Thomas, K. Noack, and C. Paris.
Evaluating interfaces for government metasearch
In *Proceedings of IIiX 2010*, pages XX–XX, 2010.

M. Baillie, G. Bordogna, F. Crestani, M. Landoni, and G. Pasi.
The PENG System: Integrating Push and Pull for Information Access.
In *Proceedings of ICADL 2007*, pages : 351–360. 2007.

K.L. Liu, W. Meng, J Qui, et al.
AllInOneNews: development and evaluation of a large-scale news metasearch engine.
In *Proceedings of the ACM SIGMOD 2007*, pages 1017-1028. ACM, 2007.

Questions?

Outline

Objectives of DIR Evaluation

- Evaluation is very important, as in all subareas of IR.
- The relative effectiveness of federated search methods tends to vary between different testbeds (i.e., set of test collections).
- Important to have different testbeds.
- Two main categories:
  - Testbeds with disjoint collections.
  - Testbeds with overlapping collections.
- There are several testbeds, here I report only some examples.

Datasets available

Table 6.1 *Testbed statistics.*

Testbed	Size (GB)	# docs (×1000) Min	# docs (×1000) Avg	# docs (×1000) Max	Size (MB) Min	Size (MB) Avg	Size (MB) Max
trec123-100col-bysource	3.2	0.7	10.8	39.7	28	32	42
trec4-kmeans	2.0	0.3	5.7	82.7	4	20	249
trec-gov2-100col	110.0	32.6	155.0	717.3	105	1 126	3 891

Datasets available

Table 6.2 *The domain names for the largest fifty crawled servers in the TREC GOV2 dataset. The 'www' prefix of the domain names is omitted for brevity.*

Collection	# docs	Collection	# docs
ghr.nlm.nih.gov	717 321	leg.wa.gov	189 850
nih.library.nih.gov	709 105	library.doi.gov	185 040
wcca.wicourts.gov	694 505	dese.mo.gov	173 737
cdaw.gsfc.nasa.gov	656 229	science.ksc.nasa.gov	170 971
catalog.kpl.gov	637 313	nysed.gov	170 254
edc.usgs.gov	551 123	spike.nci.nih.gov	145 546
catalog.tempe.gov	549 623	flowmon.boulder.noaa.gov	136 583
fs.usda.gov	492 416	house.gov	134 608
gis.ca.gov	459 329	cdc.gov	132 466
csm.ornl.gov	441 201	fda.gov	111 950
fgdc.gov	403 648	forums.census.gov	105 638
archives.gov	367 371	atlassw1.phy.bnl.gov	98 227
oss.fnal.gov	363 942	ida.wr.usgs.gov	90 625
census.gov	342 746	ornl.gov	88 418
ssa.gov	340 608	ncicb.nci.nih.gov	83 902
cfpub2.epa.gov	337 017	ftp2.census.gov	82 547
cfpub.epa.gov	315 116	walrus.wr.usgs.gov	81 758
contractsdirectory.gov	311 625	nps.gov	79 870
lawlibrary.courts.wa.gov	306 410	in.gov	77 346
uspto.gov	286 606	nist.time.gov	77 188
nis.www.lanl.gov	280 106	elections.miamidade.gov	73 863
d0.fnal.gov	262 476	hud.gov	70 787
epa.gov	257 993	ncbi.nlm.nih.gov	68 127
xxx.bnl.gov	238 259	nal.usda.gov	66 756
plankton.gsfc.nasa.gov	205 584	michigan.gov	66 255

Evaluation measures

- DIR evaluation uses the same evaluation measures of IR.
- The benchmark is a centralised IR system, that is DIR is compared with IR over the crawled set of all resources.
- Currently DIR performs almost as well as IR, and in some cases even better.

Essential DIR Evaluation References

James Callan
Distributed Information Retrieval.
In Croft, B., Editor, *Advances in Information Retrieval*, chapter 5, pages 127-150. Kluwer Academic Publishers, 2000.

James Callan, Fabio Crestani, and Mark Sanderson
Distributed Multimedia Information Retrieval.
Lecture Notes in Computer Science Vol. 2924, Springer-Verlag, 2004.

Questions?

Outline

Topics Covered

5. Applications
   - Vertical Selection
   - Blog Distillation
   - Other Applications

Objectives of Vertical Selection

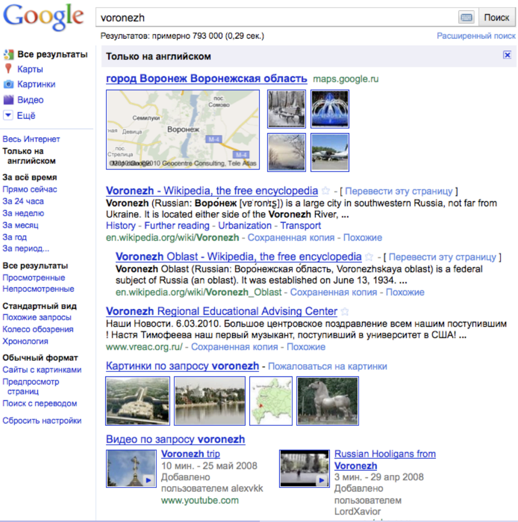
Vertical

Specialized subcollection focused on a *specific domain* (e.g., news, travel, and local search) or a *specific media type* (e.g., images and video).

Vertical Selection

The task of selecting the relevant verticals, if any, in response to a user query.

Vertical Selection Example



Vertical Selection

Recently emerged and currently very hot topic.

Two out of four published papers by Fernando Diaz et al. won best paper awards (WSDM'2009, SIGIR'2009).

Vertical Selection is a special case of DIR.

Vertical Selection vs. Resource Selection

Vertical Selection

- Verticals specialize on identifiable domains and types of media – users can express interest in vertical content explicitly by using keywords like "news", "pictures" and so on.
- Verticals are usually run by search engines that have access to their query-logs.
- If users do not seek for vertical content, no vertical should be selected.

Resource Selection

- Usually there is no way for users to specify what resource they prefer to search.
- Resources are run separately by their owners and do not provide access to their query-logs.
- Some resource should always be selected.

Approaches to Vertical Selection

Currently there are two approaches to Vertical Selection:

- **Classification-based**: each vertical is decided to be displayed or not by a binary classifier.
- **Probabilistic**: each vertical is assigned a probability to be displayed.

Classification-based Vertical Selection - Outline

1. Features
2. Classification
3. Results
4. Discussion

Classification-based Vertical Selection - Features

- **Query string features**
  - Rule-based vertical triggers:
    movies → *movies*
    sports, *sports player* → *sports*
    events, *weather* → several verticals
  - Geographic features: *airport, country, historical town, land feature, zip code* ...
- **Query-log feature**
  - Similarity between a *query* and a vertical's *query-log*
- **Corpus features**
  - Similarity between a *query* and a vertical's *content* (clarity)
  - A score assigned to a vertical by ReDDE resource selection algorithm

Classification-based Vertical Selection - Classification

- Each query is manually assigned to a number of relevant verticals (between zero and six) for training and evaluation purposes.
- Single-feature runs: vertical with the highest feature score is selected.
- Multiple-feature run:
  - multiple logistic regression model is trained for each vertical
  - vertical with the highest combined score (obtained from a trained model) is selected.
- Precision-based quality measure:
  $\mathcal{P} = \frac{1}{\mathcal{Q}}(\sum_{q \in \mathcal{Q} | \nu_q \neq 0} \mathcal{I}(\tilde{\nu_q} = \nu_q) + \sum_{q \in \mathcal{Q} | \nu_q = \emptyset} \mathcal{I}(\tilde{\nu_q} = \emptyset))$

Classification-based Vertical Selection - Results

Feature	$\mathcal{P}$
Clarity	0.254
No vertical	0.263†
ReDDE	0.336†
Query-log	0.368†
Multiple	0.583†

Table: Precision of single- and multiple-feature predictors

Classification-based Vertical Selection - Discussion

- Query-log feature is the best single evidence predictor.
- Query-logs are not accessible in uncooperative DIR environment.
- ReDDE feature is very close to the query-log one.
- Multiple-feature prediction has 58% improvement over the best single-feature predictor.

Classification-based Vertical Selection - Results

Features	$\mathcal{P}$	diff %
all	0.583	
No query-logs	0.583	0.03%
No triggers	0.583	-0.03%
No clarity	0.582	-0.10%
No geo-inf	0.577†	-1.01%
No ReDDE	0.568†	-2.60%

Table: Multiple-feature predictors with one feature out, showing the contribution of that feature

Classification-based Vertical Selection - Discussion

- Features may be correlated $\Longrightarrow$ performance drop does not necessarily mean that the feature captures no useful information.
- Query-log feature, the best single evidence predictor, does not contribute significantly because it might be highly correlated with other features.
- ReDDE feature contributes significantly.

Probabilistic Vertical Selection - Outline

1. Estimation of the probability of a vertical to be displayed
    1. Estimation based on offline training
    2. Adaptation in the presence of feedback
    3. Using information from similar queries
2. Results
3. Discussion

Probabilistic Vertical Selection - Training Data & Features

- **Training data**: each query is manually assigned to a number of relevant verticals (between zero and six).
- **Features**: discussed in the classification-based Vertical Selection.
  - Query string features
  - Query-log features
  - Corpus features

Probabilistic Vertical Selection - Training & Prediction

The idea

Selection of a vertical ν for a query q is considered as a Bernoulli experiment with unknown probability of success π_q^ν.
In other words, π_q^ν is a probability of a vertical ν to be displayed for a query q.

Training phase

π_q^ν is modeled as a function of the *features* by using logistic regression on the *training data*.

Prediction phase

For a user query q and for each vertical ν, π_q^ν is predicted by the *trained model*.

Probabilistic Vertical Selection - Feedback

- User clicks/skips of the displayed verticals are considered as a feedback:
  - $\mathcal{R}_q^\nu$ - the number of *clicks* for a query q and a vertical ν
  - $\bar{\mathcal{R}}_q^\nu$ - the number of *skips* for a query q and a vertical ν
- Offline estimated probability of a vertical ν to be relevant to a query q, π_q^ν**, needs to be adjusted in the presence of feedback**.
- p_q^ν - probability adapted in the presence of feedback.

Probabilistic Vertical Selection - Adaptation

- π_q^ν is a probability of success in a Bernoulli experiment.
- Beta distribution is a conjugate prior to a Bernoulli one.
- Adapted probability p_q^ν is modeled as a Beta distribution.

$$p_q^\nu \sim Beta(a_q^\nu, b_q^\nu) = \frac{p^{a-1}(1-p)^{b-1}}{\int_0^1 p^{a-1}(1-p)^{b-1}\,du}$$

$a_q^\nu = \mu\pi_q^\nu$, $b_q^\nu = \mu(1-\pi_q^\nu)$, μ is a hyperparameter.

Probabilistic Vertical Selection - Adaptation

In the presence of feedback ($\mathcal{R}_q^\nu$ and $\bar{\mathcal{R}}_q^\nu$) p_q^ν is adapted.

By the property of a conjugate prior $p_q^\nu | \mathcal{R}_q^\nu, \bar{\mathcal{R}}_q^\nu$ is also distributed according to a Beta distribution but with new parameters.

$$p_q^\nu | \mathcal{R}_q^\nu, \bar{\mathcal{R}}_q^\nu \sim Beta(a_q^\nu + \mathcal{R}_q^\nu, b_q^\nu + \bar{\mathcal{R}}_q^\nu)$$

$$\tilde{p}_q^\nu = \frac{a_q^\nu + \mathcal{R}_q^\nu}{(a_q^\nu + \mathcal{R}_q^\nu) + (b_q^\nu + \bar{\mathcal{R}}_q^\nu)} = \frac{\mathcal{R}_q^\nu + \mu \pi_q^\nu}{\mathcal{V}_q^\nu + \mu}, \ \mathcal{V}_q^\nu = \mathcal{R}_q^\nu + \bar{\mathcal{R}}_q^\nu$$

$$\tilde{p}_q^\nu = \frac{\mathcal{R}_q^\nu + \mu \pi_q^\nu}{\mathcal{V}_q^\nu + \mu}$$

Probabilistic Vertical Selection - Similar Queries

A prior probability π_q^ν of a vertical ν to be relevant to a query q is likely to be related to the feedback on *topically similar queries*.

1. *Topically similar queries* are identified by calculating the distance between queries with any metrics – $Dist(q, q')$.
2. The impact $\hat{\pi}_q^\nu$ of similar queries is calculated.
3. Prior π_q^ν is adjusted according to this impact.

$$\hat{\pi}_q^\nu = \frac{1}{\mathcal{Z}_q} \sum_{q'} Dist(q, q') \tilde{p}_{q'}^\nu$$

$$\tilde{\pi}_q^\nu = (1 - \lambda_q)\pi_q^\nu + \lambda_q \hat{\pi}_q^\nu$$

Probabilistic Vertical Selection - Results

Precision-based quality measure:
$\mathcal{P} = \frac{1}{\mathcal{Q}}(\sum_{q \in \mathcal{Q}|\nu_q \neq 0} \mathcal{I}(\tilde{\nu_q} = \nu_q) + \sum_{q \in \mathcal{Q}|\nu_q = \emptyset} \mathcal{I}(\tilde{\nu_q} = \emptyset))$

Run	$\mathcal{P}$
only prior π	0.618
p with $\mathcal{U}$ prior[1]	0.745
p with π prior	0.878
p with π prior & sim. queries	0.885

[1] $\mathcal{U}$ - uniform prior with $\pi = \frac{1}{2}$.

Probabilistic Vertical Selection - Discussion

- Adjusted probability p outperforms prior probability π.
- Adjusted probability p with offline trained prior π outperforms the one with uniform prior $\mathcal{U}$.
- Similar queries do not improve the precision substantially.

Vertical Selection References

Fernando Diaz.
Integration of news content into web results.
In *Proceedings of the ACM WSDM*, pages 182–191. ACM, 2009. **Best paper award.**

Fernando Diaz and Jaime Arguello.
Adaptation of offline vertical selection predictions in the presence of user feedback.
In *Proceedings of the ACM SIGIR*, pages 323–330. ACM, 2009.

Jaime Arguello, Fernando Diaz, Jamie Callan, and Jean-Francois Crespo.
Sources of evidence for vertical selection.
In *Proceedings of the ACM SIGIR*, pages 315–322. ACM, 2009. **Best paper award.**

Jaime Arguello, Fernando Diaz, and Jean-Francois Crespo.
Vertical selection in the presence of unlabeled verticals.
In *Proceedings of the ACM SIGIR*, pages 691–698. ACM, 2010.

Classification-based Resource Selection

Almost the same as classification-based Vertical Selection.

Classification-based Resource Selection - Outline

1. Features
2. Classification
3. Training Data
4. Results
5. Discussion

Classification-based Resource Selection - Features

- Corpus features
  - CORI
  - Geometric Average

$$GAVG_q(C_i) = (\prod_{d \in S_{C_i}} P(q|d))^{\frac{1}{|S_{C_i}|}}$$

  - ReDDE.top: $P(q|d)$ instead of $P(\mathcal{R}|d)$

$$ReDDE.top_q(C_i) = \frac{|C_i|}{|S_{C_i}|} \sum_{d \in \mathcal{R}_N^{sampled}} \mathcal{I}(d \in C_i) P(q|d)$$

- Query category features
- Click-through features (if available)

Classification-based Resource Selection - Classification

The same as for Vertical Selection.

- Logistic regression model is trained for each resource.
- Given a query q, each model makes a binary prediction with respect to its resource.
- Resources are ranked by $P_i(C_i = 1|q)$ - the confidence of a positive prediction from the i-th resource model.

Classification-based Resource Selection - Classification

Classification can be further adapted for DIR by considering also a relationship between resources.

Relationship can be defined by similarity between resources.

Any similarity measure can be used.

Classification-based Resource Selection - Training Data

Differs from Vertical Selection.

- Full centralized index is created.
- A training query q is issued to this index and top T documents are retrieved.
- Resource C_i is a positive instance for q if more than τ documents from C_i are in the top T.
- In the current studies $T = 30$, $\tau = 3$.

Usually full centralized index is not available in DIR!

Classification-based Resource Selection - Training Data

- A training query q is issued to a resource C_i and top T documents are retrieved.
- Resource C_i is a positive instance for q if there are more than τ **relevant** documents.
- In the current studies $T = 100$, $\tau = \{1, 3\}$.

Relevance judgements are needed!

Classification-based Resource Selection - Results

gov2.1000.1000

P@5

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.569	0.224	**0.405**	0.360	0.166	0.192	0.183	0.392 (-3.31%)
2	0.569	0.315	0.446	**0.447**	0.275	0.256	0.239	0.436 (-2.40%)
3	0.569	0.372	0.479	**0.489**	0.336	0.302	0.277	0.482 (-1.37%)
4	0.569	0.405	0.483	**0.506**	0.380	0.321	0.322	**0.506 (0.00%)**
5	0.569	0.417	0.495	**0.529**	0.395	0.336	0.337	0.510 (-3.55%)

P@10

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.534	0.188	0.331	0.321	0.150	0.152	0.147	**0.355 (7.30%)**
2	0.534	0.264	0.390	0.394	0.248	0.215	0.194	**0.399 (1.19%)**
3	0.534	0.323	0.423	0.436	0.302	0.261	0.228	**0.446 (2.47%)**
4	0.534	0.359	0.438	0.457	0.344	0.285	0.270	**0.458 (0.15%)**
5	0.534	0.380	0.442	**0.484**	0.364	0.302	0.281	0.468 (-3.33%)

P@30

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.452	0.113	0.201	0.206	0.102	0.095	0.091	**0.224 (8.68%)**
2	0.452	0.167	0.266	0.268	0.168	0.139	0.124	**0.281 (4.59%)**
3	0.452	0.217	0.305	0.312	0.206	0.170	0.152	**0.319 (2.51%)**
4	0.452	0.247	0.319	0.337	0.248	0.194	0.185	**0.339 (0.53%)**
5	0.452	0.266	0.325	**0.362**	0.275	0.205	0.195	0.352 (-2.60%)

Figure: Unsupervised vs. Classification-based Resource Selection.

Classification-based Resource Selection - Results

gov2.1000.1000

P@10

k	all.features	no.cori	no.gavg	no.redde.top	no.cats	no.click
1	0.355	0.355 (0.00%)	0.357 (0.57%)	**0.331 (-6.81%)**	0.355 (0.00%)	0.354 (0.19%)
2	0.399	0.399 (0.00%)	0.393 (-1.52%)	**0.383 (-4.04%)**	0.385 (-3.37%)	0.401 (-0.51%)
3	0.446	0.446 (-0.15%)	0.436 (-2.26%)	**0.401 (-10.23%)** ‡	0.436 (-2.41%)	0.438 (-1.95%)
4	0.458	0.456 (-0.29%)	0.442 (-3.52%) †	**0.425 (-7.18%)** †	0.450 (-1.76%)	0.449 (-1.91%)
5	0.468	0.467 (-0.14%)	0.454 (-3.01%) †	**0.431 (-7.89%)** †	0.466 (-0.43%)	0.456 (-2.58%)

P@30

k	all.features	no.cori	no.gavg	no.redde.top	no.cats	no.click
1	0.224	0.224 (0.00%)	0.227 (1.40%)	**0.213 (-5.19%)**	0.229 (2.20%)	0.225 (-0.20%)
2	0.281	0.281 (0.16%)	0.274 (-2.39%)	**0.266 (-5.02%)**	0.271 (-3.51%)	0.277 (-1.44%)
3	0.319	0.317 (-0.77%)	0.311 (-2.59%)	**0.292 (-8.61%)** †	0.312 (-2.24%)	0.313 (-2.10%)
4	0.339	0.338 (-0.20%)	0.330 (-2.70%)	**0.319 (-5.80%)** †	0.331 (-2.38%)	0.336 (-0.79%)
5	0.352	0.350 (-0.51%)	0.344 (-2.35%)	**0.331 (-5.97%)** †	0.347 (-1.52%)	0.344 (-2.35%) †

Figure: Contribution of different features.

Classification-based Resource Selection - Results

Src Rank	TREC123		TREC4	
	Ind	Jnt	Ind	Jnt
@1	0.262	**0.319(21.8%)‡**	0.287	**0.309(7.7%)**
@3	0.309	**0.364(17.8%)‡**	0.324	**0.340(4.9%)**
@5	0.354	**0.400(13.0%)‡**	0.343	**0.355(3.5%)**
@10	0.426	0.426(0%)	0.414	0.414(0%)

Figure: Independent resource model (Ind) vs. model with resource relationships (Jnt): R_k.

Classification-based Resource Selection - Results

Docs Rank	TREC123		
	Full	Ind	Jnt
@5	0.446	0.392	**0.410(4.6%)**
@10	0.444	0.355	**0.360(1.4%)**
@15	0.435	0.332	**0.347(4.5%)**$*$
@20	0.430	0.309	**0.326(5.5%)**†
@30	0.414	0.280	**0.300(7.1%)**‡

Docs Rank	TREC4		
	Full	Ind	Jnt
@5	0.549	0.282	**0.290(2.8%)**
@10	0.459	0.238	**0.254(6.7%)**
@15	0.422	0.209	**0.224(7.2%)**
@20	0.384	0.186	**0.200(7.5%)**
@30	0.354	0.167	**0.170(1.8%)**

Figure: Independent resource model (Ind) vs. model with resource relationships (Jnt): P@N.

Classification-based Resource Selection - Discussion

+ Can incorporate all existing Resource Selection algorithms as features.
+ Usually at least as good as the best unsupervised Resource Selection technique.

- Needs training data!
  - Full centralized index that is usually unavailable in DIR.
  - Relevance judgements.

Classification-based Resource Selection References

J. Arguello, J. Callan, and F. Diaz.
Classification-based resource selection.
In *Proceedings of the ACM CIKM*, pages 1277–1286. ACM, 2009.

D. Hong, L. Si, P. Bracke, M. Witt, and T. Juchcinski.
A joint probabilistic classification model for resource selection.
In *Proceedings of the ACM SIGIR*, pages 98–105. ACM, 2010.

Questions?

Objectives of Blog Distillation

Blog Distillation (aka Feed Search) is concerned with finding blogs (feeds) with a recurring central interest.

Blog Distillation track was introduced in TREC 2007. It is a new and hot topic. Just a few methods are proposed (mostly in 2008).

We would like to thank Mostafa Keikha, a PhD student at the University of Lugano, for his help in creating this section.

Approaches to Blog Distillation

- **Ad-hoc**: considers blogs and posts as regular documents and applies standard ad-hoc IR retrieval techniques.
- **DIR**: considers blogs as federated resources and performs resource selection for them.
- **Expert search**: considers bloggers as experts and ranks them according to their expertise in a given query.

Ad-hoc & DIR Blog Distillation

Ad-hoc and DIR methods for Blog Distillation are highly interconnected and, therefore, will be discussed together.

There are two main groups of approaches:

- **Large Document Model** (LDM): treats each blog feed as a single monolithic document.
- **Small Document Model** (SDM): treats a blog feed as a collection of individual documents.

Blog Distillation with LDM

- Each blog feed is considered as a monolithic *large document* (LD).
- LDs are ranked with ad-hoc techniques.
- Similar to CORI Resource Selection algorithm.

Blog Distillation with LDM - Ranking Methods

- KL-Divergence between a query and LD language models

$$s(Q,F) = -KL(\theta_Q || \theta_F) = \sum_w P(w|\theta_Q) \log \frac{P(w|\theta_F)}{P(w|\theta_Q)}$$

- Query likelihood according to LD

$$P(Q,F) = P(Q|LD) = \prod_{q \in Q} P(q|LD)$$

- Probabilistic method

$$P_{LD}(F|Q) = \frac{\overbrace{P(F)}^{Feed\ Prior}\ \overbrace{P_{LD}(Q|F)}^{Query\ Likelihood}}{P(Q)}$$

$$P_{LD}(Q|F) = \prod_{w_i \in Q} P_{LD}(w_i|F)^{w_i}$$

Blog Distillation with SDM

- Each blog feed is considered as a collection of individual *small documents* (SD).
- SDs are ranked with ad-hoc techniques.
- Similar to ReDDE Resource Selection algorithm.

Blog Distillation with SDM - Ranking Method

$$
\begin{aligned}
P_{SD}(F|Q) &= \frac{1}{P(Q)} \sum_{E \in F} P_{SD}(Q, E, F) \\
&\stackrel{rank}{=} P(F) \sum_{E \in F} P(Q|E, F)\ P(E|F) \\
&\stackrel{rank}{=} \underbrace{P(F)}_{Feed\ \ Prior} \sum_{E \in F} \underbrace{P(Q|E)}_{Query\ \ Likelihood} \quad \underbrace{P(E|F)}_{Entry\ \ Centrality}
\end{aligned}
$$

$P(Q|E) = \prod_{q \in Q} P(q|E)$

$P(E|F) = \frac{Sim(E,F)}{\sum_{E_i \in F} Sim(E_i,F)}$

Expert Search for Blog Distillation

There are two approaches that apply Expert Search for Blog Distillation:

- **Probabilistic** - the same as Small Document Model.
- **Voting** - blog feed's score depends on the number of blog posts appearing in the ranked list.

Expert Search for Blog Distillation - Voting Model

- $R(Q)$ - centralized ranking of blog posts for a query Q.
- $posts(F)$ - posts in a blog feed F.
- $score_{votes}(Q, F) = |R(Q) \cap posts(F)|$
- *Comb*-family fusion methods are used to rank blog feeds.

$$score_{CombMAX}(Q, F) = \max_{E \in R(Q) \cap posts(F)} (Sim(Q, E))$$

$$score_{CombSUM}(Q, F) = \sum_{E \in R(Q) \cap posts(F)} \exp^{Sim(Q,E)}$$

$$score_{CombMNZ}(Q, F) = score_{votes}(Q, F) \cdot \sum_{E \in R(Q) \cap posts(F)} \exp^{Sim(Q,E)}$$

Blog Distillation Results

Method	MAP	R-prec	b-Bref	P@10
LDM 1	**0.3695**	**0.4245**	**0.3861**	**0.5356**
Voting Model	0.2923	0.3654	0.3210	0.5311
SDM	0.2552	0.3384	-	0.4267
LDM 2	0.2529	0.3334	0.2902	0.5111
KL-Divergence	0.2197	0.3100	0.2649	0.4511

Metadata for Blog Distillation

Blog Search is a particular case of cooperative DIR where additional metadata about blog feeds and posts is available.

Metadata used in Blog Distillation include:

- Temporal evidence - correlation between topics and time
- Link structure
  - linked posts may be related to each other
  - the number of incoming links is the evidence of authoritativeness of a post
- Authorship
- Comments
- Others...

Essential Blog Distillation References

TREC Proceedings starting from 2007, Blog Track at http://trec.nist.gov/proceedings/proceedings.html.

C. Macdonald, I. Ounis, and I. Soboroff.
Overview of the trec 2007 blog track.
In *TREC*, 2007.

M. Efron, D. Turnbull, and C. Ovalle.
University of texas school of information at trec 2007.
In *TREC*, 2007.

Jonathan L. Elsas, Jaime Arguello, Jamie Callan, and Jaime G. Carbonell.
Retrieval and feedback models for blog feed search.
In *Proceedings of the ACM SIGIR*, pages 347–354. ACM, 2008.

Jangwon Seo and W. Bruce Croft.
Blog site search using resource selection.
In *Proceeding of the ACM CIKM*, pages 1053–1062. ACM, 2008.

K. Balog, M. de Rijke, and W. Weerkamp.
Bloggers as experts: feed distillation using expert retrieval models.
In *Proceedings of the ACM SIGIR*, pages 753–754. ACM, 2008.

C. Macdonald and I. Ounis.
Key blog distillation: ranking aggregates.
In *Proceedings of the ACM CIKM*, pages 1043–1052. ACM, 2008.

Questions?

Other Applications

Other research areas where DIR techniques are used include:

- Expert Search
- Desktop Search
- and more...

Expert Search

Expert Search

The task of identifying experts with a given expertise.

The idea

Experts $\Longleftrightarrow$ documents authored by an expert

Resource Selection on different collections of documents.

Desktop Search

Desktop Search

The task of identifying desktop files and documents of different types relevant to a user query.

The idea

Resource Selection on different file and document types.
Results Fusion on different documents.

Your Application of DIR

Questions?