Distributed Information Retrieval

Prof. Fabio Crestani

Faculty of Informatics
University of Lugano, Switzerland
fabio.crestani@usi.ch

Acknowledgements

Large part of this material were presented at the Tutorial on Distributed Information Retrieval at ECIR 2010, in Milton Keynes, UK.
The tutorial was presented jointly with *Ilya Markov*, a PhD student at the University of Lugano, whose help was invaluable in the preparation of this material.

Objectives of the Course

This short course aims at providing the listener with an overview of the area of research of Distributed Information Retrieval (DIR). Highlighting the key elements of this technology and showing how the methods developed in the context of DIR can be applied also in other areas of research.

A set of references accompanies this material to help deepen the knowledge on specific aspects of the topics covered.

Topics covered in this Course

Outline

Background needed

DIR is a subarea of research of Information Retrieval (IR). It shares with IR the objectives and much of the underlying technology for indexing and retrieving information. So, it is not possible to understand DIR without some background knowledge of IR.

The objectives of this part of the course is to quickly get you up to speed with the main concepts of IR.

Topics Covered

1. Background
   - Information Retrieval
   - Indexing
   - Querying
   - IR Models
   - Evaluation

What is Information Retrieval?

- Search on the Web is a daily activity for many people throughout the world
- Applications involving search are everywhere
- The field of computer science that is most involved with R&D of search technology is *Information Retrieval (IR)*
- A definition: "Information retrieval is a field concerned with the structure, analysis, organisation, storage, searching, and retrieval of information." (Salton, 1968)
- Primary focus of IR since the 50s has been on *text* and *textual documents*

What is a Document?

- A document in IR can be anything with some *semantic content* and (maybe) with some structure
- Documents are very different from *database records*
- The core issue of IR is: comparing the query text to the documents' text and determining what is a *good match*
- The *exact match* of words is not sufficient: there are many different ways to write the same thing in a natural language

IR Tasks

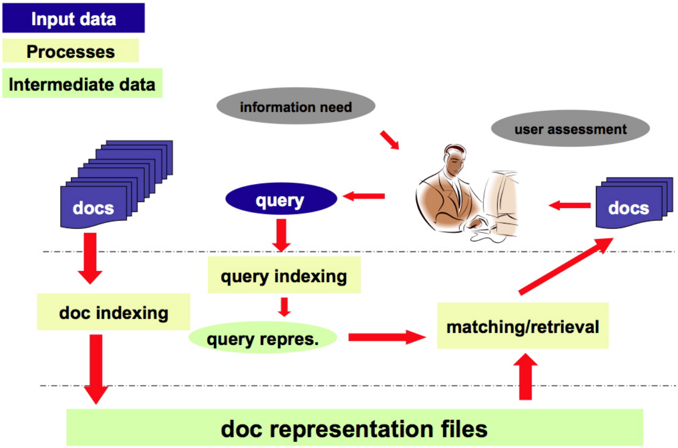
IR deals with many different types of content, applications and tasks. In particular, the most well known tasks are:

- Ad-hoc search: find relevant documents for an arbitrary text query
- Filtering: identify new documents relevant to user profiles
- Classification: identify relevant labels for documents
- Question answering: give a specific answer to a question

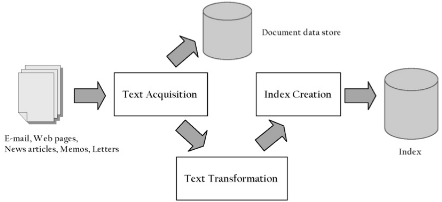
The Big Issue in IR: Relevance

- A simplistic definition of relevance: a document is relevant to a query if it contains the information that a person was looking for when he/she submitted the query to the IR system
- Many factors influence a persons decision about what is relevant: task, context, novelty, style, the usefulness, etc.
- An important distinction: topical relevance (same topic) vs. user relevance (everything else)
- An IR system can only *estimate* the relevance of a document to a query, using a *document representation*, and a *retrieval model* (that defines a view of relevance)
- It is important to verify if the system's estimate is correct, hence the importance of *evaluation* in IR

The IR Process



The Indexing Process



- Text acquisition: identifies and stores documents for indexing
- Text transformation: transforms documents into index terms or features
- Index creation: takes index terms and creates data structures (indexes) to support fast searching

Text Acquisition

Could be carried out in different ways:

- Collection: documents could be provided directly (e.g., digital libraries)
- Crawler: identifies and acquires documents for IR system (e.g., web, enterprise, desktop search)
- Feeds: real-time streams of documents (e.g., web feeds for news, blogs, video, radio, tv)

Documents end up in a *Document Data Store* that stores text, metadata, and other related content for documents for fast access to document contents in other phases of the IR process

Text Transformation

It usually involves:

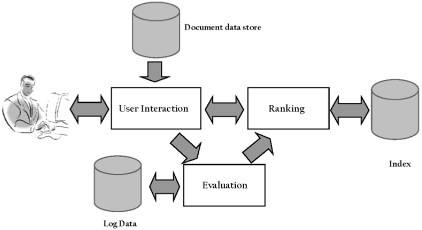
1. Parsing: processing the sequence of text tokens in the document to recognise structural elements and words in the text
2. Stopping: remove words with no semantic meaning and common words
3. Stemming: group words derived from a common stem
4. Link analysis: makes use of links and anchor text in web pages
5. Information Extraction: identify classes of index terms that are important for some applications
6. Classification: identifies class-related metadata for documents

Index Creation

It usually involves:

1. Calculating document statistics: gathers counts and positions of words and other features
2. Weighting: computes weights for index terms (used by ranking algorithm)
3. Inversion: converts document-term information to term-document for indexing (the core of indexing process: building an inverted file)
4. Index distribution: distributes indexes across multiple computers and/or multiple sites (essential for fast query processing with large numbers of documents)

The Querying Process



- User interaction: supports creation and refinement of query, and the display of results
- Ranking: uses query and indexes to generate ranked list of documents
- Evaluation: monitors and measures effectiveness and efficiency of the search process (primarily offline)

User Interaction

It usually involves:

1. Query input: provides interface and parser for query language
2. Query transformation: improves initial query, both before (same text transformation techniques used for documents) and after the initial search (query expansion and relevance feedback)
3. Results output: constructs the display of ranked list of documents for a query (including document surrogates)

Ranking

It usually involves:

1. Scoring: calculates scores of documents using a ranking algorithm (core component of the IR engine, with many variations depending on the retrieval model)
2. Performance optimisation: designing ranking algorithms for efficient processing
3. Query processing distribution: processing queries in a distributed environment (involving a query broker and caching)

More about this when we talk about retrieval models

Evaluation

It usually involves:

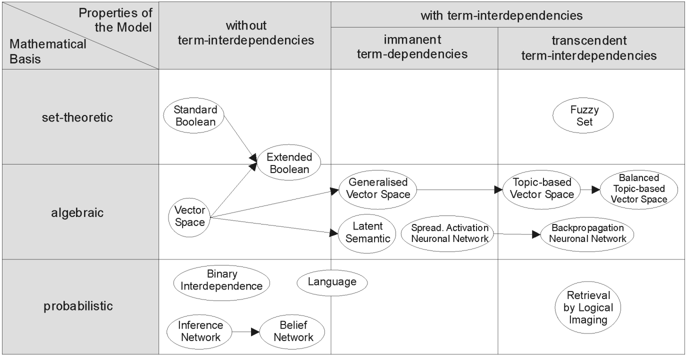
1. Logging: logging user queries and interaction to mine for improving search effectiveness and efficiency
2. Ranking analysis: measuring and tuning ranking effectiveness
3. Performance analysis: measuring and tuning system efficiency

More about this when we talk about IR evaluation

Information Retrieval Models

1. A retrieval model is a *mathematical framework* for defining the search process, including the underlying assumptions and the interpretation of the concept of relevance
2. Progress in retrieval models has corresponded with improvements in effectiveness
3. Retrieval models make various assumptions about document representations (e.g., bag of words, document independence) and relevance (e.g., topical relevance, binary relevance)

A Taxonomy of Information Retrieval Models



Source: Wikipedia

Information Retrieval Evaluation

- Evaluation is a key factor to building effective and efficient IR system
- Measurement usually carried out in controlled laboratory experiments, but online testing can also be done
- Core to IR evaluation are: collections (corpora), queries, relevance judgements, evaluation metrics
- IR evaluation has developed over 40 years thanks to evaluation frameworks like Cranfield and TREC

Corpora

- CACM: Titles and abstracts from the Communications of the ACM from 1958-1979. Queries and relevance judgments generated by computer scientists.
- AP: Associated Press newswire documents from 1988-1990 (from TREC disks 1-3). Queries are the title fields from TREC topics 51-150. Topics and relevance judgments generated by government information analysts.
- GOV2: Web pages crawled from websites in the .gov domain during early 2004. Queries are the title fields from TREC topics 701-850. Topics and relevance judgments generated by government analysts.

Collection	Number of documents	Size	Average number of words/doc.
CACM	3,204	2.2 Mb	64
AP	242,918	0.7 Gb	474
GOV2	25,205,179	426 Gb	1073

Queries and Relevance Judgements

<top>
<num> Number: 794

<title> pet therapy

<desc> Description:
How are pets or animals used in therapy for humans and what are the benefits?

<narr> Narrative:
Relevant documents must include details of how pet- or animal-assisted therapy is or has been used. Relevant details include information about pet therapy programs, descriptions of the circumstances in which pet therapy is used, the benefits of this type of therapy, the degree of success of this therapy, and any laws or regulations governing it.

</top>

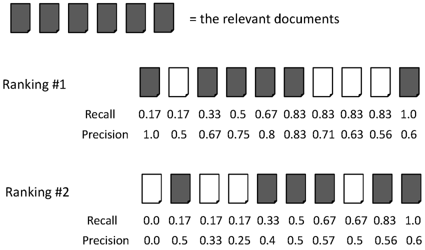
Collection	Number of queries	Average number of words/query	Average number of relevant docs/query
CACM	64	13.0	16
AP	100	4.3	220
GOV2	150	3.1	180

Evaluation Metrics: Precision and Recall

	Relevant	Non-Relevant
Retrieved	$A \cap B$	$\overline{A} \cap B$
Not Retrieved	$A \cap \overline{B}$	$\overline{A} \cap \overline{B}$

$$Recall = \frac{|A \cap B|}{|A|}$$

$$Precision = \frac{|A \cap B|}{|B|}$$



Other Evaluation Metrics

There are many other evaluation metrics

- False Positive (Fall out or type I error) and False Negative (or type II error)
- F Measure: harmonic mean of recall and precision
- Mean Average Precision (MAP)
- Discounted Cumulative Gain (DCG, for graded relevance) and Normalised Discounted Cumulative Gain (NDCG)

Essential Information Retrieval References

Keith Van Rijsbergen
Information Retrieval. Second Edition.
Butterworths, London, 1979.

Bruce Croft, Donald Metzler, Trevor Strohman
Search Engines: Information Retrieval in Practice.
Addison Wesley, New York, 2009.

Maristella Agosti, Fabio Crestani, Gabriella Pasi.
Lectures on Information Retrieval, Third European Summer-School, ESSIR 2000.
Varenna, Italy, September 11-15, 2000, Revised Lectures, Springer-Verlag, Berlin, 2001.

Questions?

Outline

Topics Covered

2. DIR: Introduction
   - What is DIR?
   - Motivations for 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 on a search engine and one or more collection of documents. Each resource is assumed to handle the search process on its own collection in a independent way
- Other names for DIR are: federated search and federated information retrieval
- Example of DIR systems are: FedStast, PubMed, US Census Bureau, Westlaw, MedLine, Cheshire, etc.

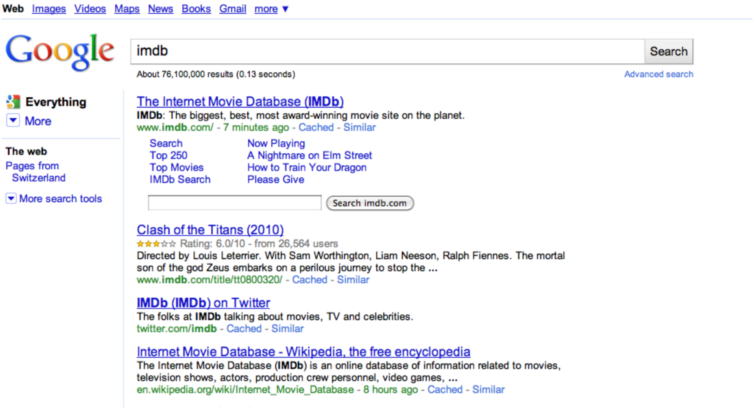
Motivations

- 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 is or can be harvested
  2. The "one size fits all" approach of web search engine 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 the resource (if possible), then the user has to deal directly with it

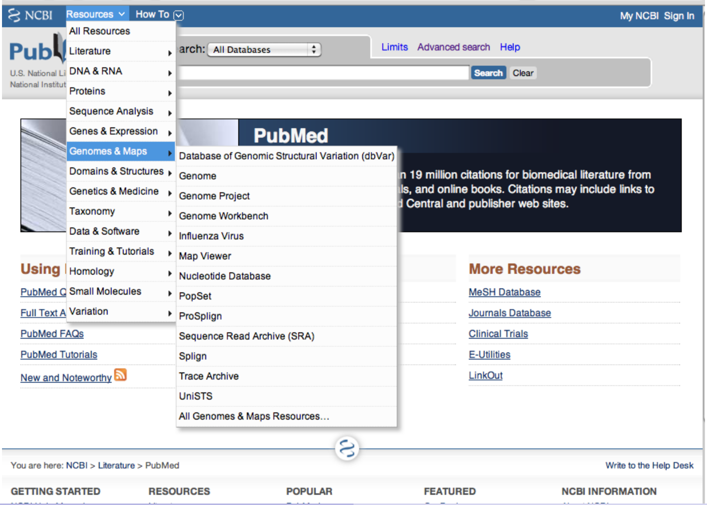
Deep Web: Example



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 the resource itself
- Why would this be better?
  - Preserves the property rights of the resource owner
  - Search facilities are optimised to the specific resource
  - Index is always up-to-date
  - The resource is curated and of high quality
- Examples of federate search systems: *PubMed*, *FedStats*, *WestLaw*, and *Cheshire*

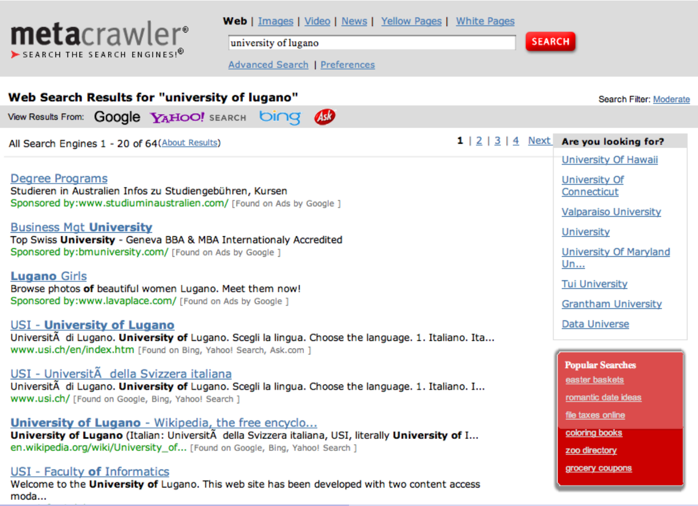
Federated Search: Example



Metasearch

- Even the largest search engine cannot crawl effectively the entire web
- Different search engines crawl different disjoint 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 results set
- Examples of federate search systems: *Dogpile*, *MataCrawler*, *AllInOneNews*, and *SavvySearch*

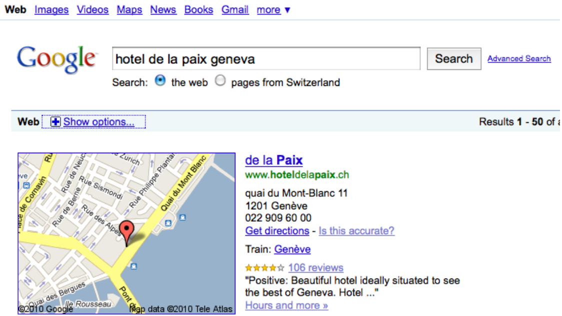
Metasearch: Example



Aggregated Search

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

Aggregated Search: Example



Questions?

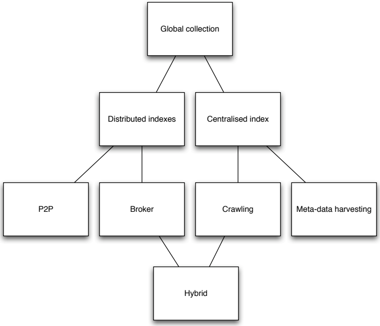
Outline

Topics Covered

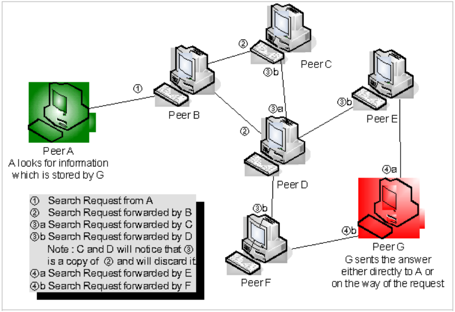
3. DIR Architectures
   - Peer-to-Peer Network
   - Broker-Based Architecture
   - Crawling
   - Metadata Harvesting
   - Hybrid

A Taxonomy of DIR Systems

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

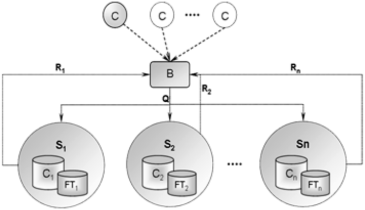


Peer-to-Peer Networks



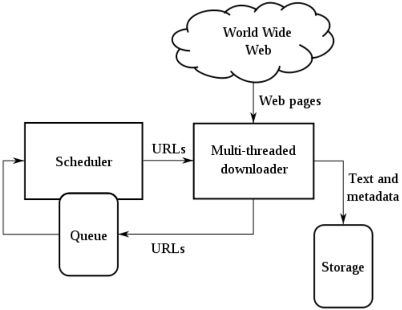
- Indexes are located with the resources
- Some part of the indexes are distributed to other resources
- Queries are distributed across the resources and results are merged by the peer that originated the query

Broker-Based Architecture



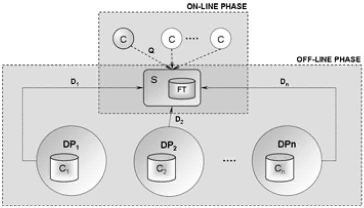
- Indexes are located with the resources
- Queries are forwarded to resources and results are merged by a *broker*

Crawling



- Resources are crawled and documents are harvested
- Indexes are centralised
- Queries are evaluated out in a centralised way and documents are fetched from resources or from a storage

Metadata Harvesting

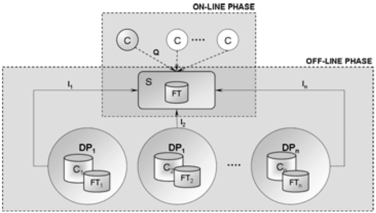


- Indexes are located with the resources, but metadata are harvested according to some protocol (off-line phase), like for example the OAI-PMH
- Queries are evaluated at the broker level (on-line phase) to identify relevant documents by the metadata, that are then requested from the resources.

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) and a set of tools that implements that
- 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



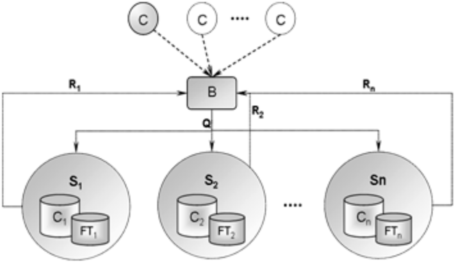
- It is possible to crawl the indexes, instead of the metadata according to some protocol (off-line phase), like for example the OAI-PMH
- Queries are evaluated out at the broker level (on-line phase) to identify relevant documents by the documents' full content, that are then requested from the resources

Questions?

Outline

1. Background
2. DIR: Introduction
3. DIR Architectures
4. Broker-Based DIR
5. DIR Evaluation
6. Applications of DIR

Architecture of a Broker-based DIR System



- Indexes are located with the resources
- Queries are forwarded to resources and results are merged by the 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

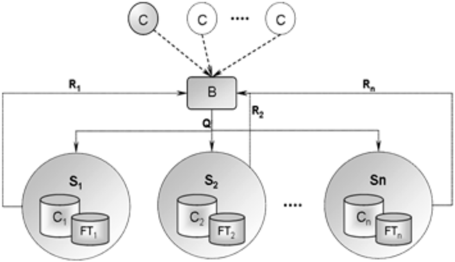
Topics Covered

4. Broker-Based DIR
   - Resource Description
   - Resource Selection
   - Results Merging

Objectives of the Resource Description Phase

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

This phase is required for all other subsequent phases.



DIR Cooperation

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

- *Cooperative* environments: the resource provides full access to the documents and the indexes and responds to queries
- *Uncooperative* environments: the resource does provides any access to the document and the indexes; it only respond to queries

Resource Description in Cooperative Environments

- Resource Description in cooperative environments can be very simple as the broker has full access to the collection(s) held at the resource
  - The broker could crawl or harvest the full collection(s) and deal with the query locally, but this might not be a good idea
  - The broker could receive from the resource information (a description) useful for the selection

Stanford Protocol Proposal for Internet and Retrieval Search (STARTS)

STARTS is similar to OAI. It stores for each resource 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 the 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 the broker does not have to full collections, or access to resource metadata and content summary
- The broker needs to acquire this information without any help from the 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

Questions

- How do we select the queries?
- When do we stop (stopping criterium)?

Selecting the Sampling Queries

Queries can be selected by:

- Other Resource Description (ORD): selecting terms from a reference dictionary
- Learned Resource Description (LRD): selecting 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 the Sampling Queries

The best experimental strategy proved to be based on random selection of terms that have an Average Term Frequency, where

$$AverageTermFrequency = \frac{CollectionTermFrequency}{DocumentFrequency}$$

Another important strategy would be to use personal query-logs to achieve a *personalised resource description*

Stopping Criteria

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

Adaptive Sampling

- Ideally we would need an adaptive stopping criterium, related to:
  - The proportion of documents sampled in relation of the size of the collection
  - The proportion of term sampled in relation to the size of the term space
  - The vocabulary growth
- There have been some attempts to propose methods for adaptive quey-based sampling

Estimating Collection Size

- The size of the collection is an important element of the resource description
- Knowing the size of the collection is useful for a better stopping criterium of query-based sampling
- It is also a useful parameter of the resource selection phase
- Two techniques have been proposed: capture-recapture and sample-resample

Capture-Recapture

The 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 the random samples should be generated

Sample-Resample

The 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

- Send the query t to the resource to estimate $df_{t,C}$
- Repeat several times and estimate collection size $|\hat{C}|$ as average value of estimates

The method relies on the resource giving the correct document frequency of the query terms

Essential Resource Description References

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.

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

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.

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.

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

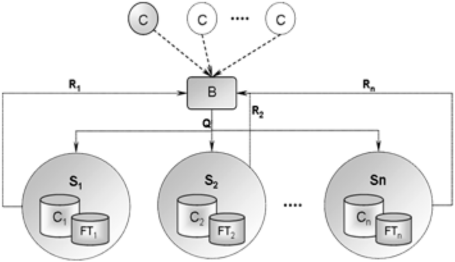
Milad Shokouhi, Justin Zobel, Seyed M. M. Tahaghoghi, and Falk Scholer.
Using query logs to establish vocabularies in distributed information retrieval.
Inf. Process. Manage., 43(1):169–180, 2007.

Questions?

Objectives of the Resource Selection Phase

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

Resource selection uses the representations built by the resource description phase and other parameters.



Approaches to Resource Selection

There are three approaches:

- Lexicon-Based: used for cooperative environments, the broker calculates the similarity of the query with the resource description using the detailed lexicon statistics of the resource
- Document Surrogate-Based: used for uncooperative environments, in addition to the above the broker use also sampled documents from each resource
- Classification-Based: used for uncooperative environments, the broker classify resources into a topic hierarchy and make selection decisions based on the matching between the query and the categories of the hierarchy

Lexicon-Based Approaches

- In Lexicon based approaches collections at resources are treated as a large single documents or vocabulary distributions.
- The "super document" that is most relevant to the query identifies the collection to select.
- The two best known methods are CORI and GIOSS

Collection Retrieval Inference Network (CORI)

- Collection $\Longrightarrow$ Super-Document
- Bayesian inference network on super-documents
- Adapted Okapi

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

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

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

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

Glossary-of-Servers Server (GlOSS)

We assume cooperation and the availability of documents and terms statistics

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

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

Document Surrogate-Based Approaches

- In document surrogate-based approaches collections at resources are selected based on their similarity with some sample documents retrieved from each resource
- Away from the super document approach and retaining document boundaries
- The two best known methods are ReDDE and CRCS

Relevant Document Distribution Estimation (ReDDE)

Idea

Estimate number of relevant documents in the collection and rank them

If one sampled document is relevant to a query $\iff \frac{|C|}{|S_C|}$ similar documents in a collection c are relevant to a query.

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

where $P(\mathcal{R}|d)$ is the prob of relevance of an arbitrary document in the description

Relevant Document Distribution Estimation (ReDDE)

Ranked sampled documents $\Longrightarrow$ Ranked documents in a centralised retrieval system

Idea

A document d_j appears before a document d_i in a sample $\Longleftrightarrow$ $\frac{|C_j|}{|S_{C_j}|}$ documents appear before d_i in a centralised retrieval system.

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

Relevant Document Distribution Estimation (ReDDE)

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

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

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

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

Centralised-Rank Collection Selection (CRCS)

Different variation of the ReDDE algorithms, based on analysis of the top ranked sampled documents to estimate relevance documents distribution in the collection

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

- **Linear**

$$\mathcal{R}(d) = \begin{cases} \gamma - Rank_{sample}(d) & \text{if } Rank_{sample}(d) < \gamma \\ 0 & \text{otherwise.} \end{cases}$$

- **Exponential**

$$\mathcal{R}(d) = \alpha \exp(-\beta \cdot Rank_{sample}(d))$$

$$P(\mathcal{R}|d) = \frac{\mathcal{R}(d)}{|C_{max}|}$$

Resource Selection Comparison

Table 2. *Performance of different methods for the Trec4 (trec4-kmeans) testbed.* TREC *topics 201–250 (long) were used as queries*

	Cutoff=1				Cutoff=5			
	P@5	**P@10**	**P@15**	**P@20**	**P@5**	**P@10**	**P@15**	**P@20**
CORI	0.3000	0.2380	0.2133†	0.1910†	0.3480	0.2980	0.2587	0.2380
ReDDE	0.2160	0.1620	0.1373	0.1210	0.3480	0.2860	0.2467	0.2190
CRCS(l)	0.2960	0.2260	0.2013	0.1810†	0.3520	0.2920	0.2533	0.2310
CRCS(e)	0.3080	0.2400	0.2173†	0.1910†	0.3880	0.3160	0.2680	0.2510

Table 3. *Performance of collection selection methods for the uniform (trec123-100col-bysource) testbed.* TREC *topics 51–100 (short) were used as queries*

	Cutoff=1				Cutoff=5			
	P@5	**P@10**	**P@15**	**P@20**	**P@5**	**P@10**	**P@15**	**P@20**
CORI	0.2520	0.2140	0.1960	0.1710	0.3080	0.3060	0.2867	0.2730
ReDDE	0.1920	0.1660	0.1413	0.1280	0.2960	0.2820	0.2653	0.2510
CRCS(l)	0.2120	0.1760	0.1520	0.1330	0.3440	0.3240	0.3067	0.2860
CRCS(e)	0.3800‡	0.3060‡	0.2613‡	0.2260†	0.3960	0.3700†	0.3480†	0.3310†

Resource Selection Comparison

Table 5. *Performance of collection selection methods for the relevant (trec123-AP-WSJ-60col) testbed.* TREC *topics 51–100 (short) were used as queries*

	Cutoff=1				Cutoff=5			
	P@5	**P@10**	**P@15**	**P@20**	**P@5**	**P@10**	**P@15**	**P@20**
CORI	*0.1440*	*0.1280*	*0.1160*	*0.1090*	*0.2440*	*0.2340*	*0.2333*	*0.2210*
ReDDE	0.3960	0.3660	0.3360	0.3270	0.3920	0.3900	0.3640	0.3490
CRCS(l)	0.3840	0.3580	0.3293	0.3120	0.3800	0.3640	0.3467	0.3250
CRCS(e)	0.3080	0.2860	0.2813	0.2680	0.3480	0.3420	0.3280	0.3170

Table 7. *Performance of collection selection methods for the* GOV*2 (100-col-*GOV*2) testbed.* TREC *topics 701–750 (short) were used as queries*

	Cutoff=1				Cutoff=5			
	P@5	**P@10**	**P@15**	**P@20**	**P@5**	**P@10**	**P@15**	**P@20**
CORI	$0.1592^{\dagger}$	$0.1347^{\dagger}$	$0.1143^{\dagger}$	$0.0969^{\dagger}$	0.2735	0.2347	0.2041	0.1827
ReDDE	0.0490	0.0327	0.0286	0.0235	0.2163	0.1837	0.1687	0.1551
CRCS(l)	0.0980	0.0755	0.0667	0.0531	0.1959	0.1510	0.1442	0.1286
CRCS(e)	0.0857	0.0714	0.0748	0.0643	0.2776	0.2469	0.2272	0.2122

Essential Resource Selection References

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.
GIOSS: text-source discovery over the internet.
ACM Trans. Database Syst., 24(2):229–264, 1999.

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.

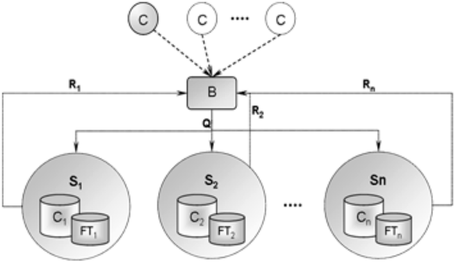
Shengli Wu, Fabio Crestani.
Distributed Information Retrieval: A Multi-Objective Resource Selection Approach.
In *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 11, pages 83–100, 2003.

Questions?

Objectives of the Results Merging Phase

The results merging phase is concerned with the broker merging the list of top-ranked documents returned from the different resources and returning a fused list to the user

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



Results Merging Issues

- The results merging process involves a number of issues:
  1. Duplicate detection and removal
  2. Normalising and merging relevance scores
- Different solutions have been proposed for these issue, depending in the DIR environment

Results Merging in Cooperative Environments

- The results merging in cooperative environments is much simpler and has different solutions:
  1. Fetch documents from each resource, reindex and rank according to the broker IR model.
  2. Get information about the way the document score is calculated and normalise score.
- At the highest level of collaboration it is possible to ask the resources to adopt the same retrieval model!

Collection Retrieval Inference Network (CORI)

The idea

Linear combination of the score of the database and the score of the document.

Normalised scores

- Normalised collection section score: $C_i' = \frac{(R_i - R_{max})}{(R_{max} - R_{min})}$
- Normalised document score: $D_j' = \frac{(D_j - D_{max})}{(D_{max} - D_{min})}$
- Heuristic linear combination: $D_j'' = \frac{D_j' + 0.4 * D_j' * C_i'}{1.4}$

Results Merging in Uncooperative Environments

- In uncooperative environments resources might provide scores:
  - But the broker does not have any information on how these score are computed.
  - Score normalisation requires some way of comparing scores.
- Alternatively the resources might provide only rank positions:
  - But the broker does not have any information on the relevance of each document in the rank lists.
  - Merging the ranks requires some way of comparing rank positions.

Semi-Supervised Learning (SSL)

The idea

Train a regression model for each collection that maps resource document scores to normalised scores.
Requires that some returned documents are found in the Collection Selection Index (CSI).

Two cases:

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

SSL with Identical Retrieval Models

The idea

SSL uses documents found in CSI to train a single regression model to estimate the normalised score ($D'_{i,j}$) from resource document scores ($D_{i,j}$) and the score of the same document computed from the CSI ($E_{i,j}$).

Normalised scores

Having:

$$\begin{bmatrix} D_{1,1} & C_1 D_{1,1} \\ D_{1,2} & C_1 D_{1,2} \\ \dots & \dots \\ D_{n,m} & C_n D_{n,m} \end{bmatrix} \times [a \;\; b] = \begin{bmatrix} E_{1,1} \\ E_{1,2} \\ \dots \\ E_{n,m} \end{bmatrix}$$

Train:

$$D'_{i,j} = a * E_{i,j} + b * E_{i,j} * C_i$$

SSL with Different Retrieval Models

The idea

SSL uses documents found in CSI to train a different regression models for each resource.

Normalised scores

Having:

$$\begin{bmatrix} D_{1,1} & 1 \\ D_{1,2} & 1 \\ \dots & \dots \\ D_{n,m} & 1 \end{bmatrix} \times [a_i \;\; b_i] = \begin{bmatrix} E_{1,1} \\ E_{1,2} \\ \dots \\ E_{n,m} \end{bmatrix}$$

Train:

$$D'_{i,j} = a_i * E_{i,j} + b_i$$

Sample-Agglomerate Fitting Estimate (SAFE)

The idea

For a given query the results from the CSI is a subranking of the original collection, so curve fitting to the subranking can be used to estimate the original scores.
It does not require the presence of overlap documents in CSI.

Sample-Agglomerate Fitting Estimate (SAFE)

Normalised scores

1. The broker ranks the documents available in the CSI for the query.
2. For each resource the sample documents (with non zero score) are used to estimate the merging score, where each sample document is assumed to be representative of a fraction $|Sc|/|c|$ of the resource.
3. Use regression to fit a curve on the adjusted scores to predict the score of the document returned by the resource.

More Results Merging

There are other approaches to results merging:

- STARTS uses the returned term frequency, document frequency, and document weight information to calculate the merging score based on similarities between documents.
- CVV calculates the merging score according to the collection score and the position of a document in the returned collection rank list.
- Another approach download small parts of the top returned documents and used a reference index of term statistics for reranking and merging the downloaded documents.

Data Fusion in Metasearch

- In data fusion methods documents in a single collection are ranked with different search engines
- The goal is to generate a single accurate ranking list from the ranking lists of different retrieval models.
- There are no collection samples and no CSI.

The idea

Use the *voting principle*: a document returned by many search systems should be ranked higher than the other documents.
If available, also take the rank of documents into account.

Metasearch Data Fusion Methods

Many methods have been proposed:

Data Fusion

- Round Robin.
- CombMNZ, CombSum, CombMax, CombMin.
- Logistic regression (convert ranks to estimated probabilities of relevance).

A comparison between score-based and rank-based methods suggests that rank-based methods are generally less effective.

Results Merging in Metasearch

- We cannot use data fusion methods when collections are overlapping, but are not the same.
- We cannot use data fusion methods when the retrieval model are different.
- Web metasearch is the most typical example.

The idea

Normalise the document scores returned by multiple search engines using a regression function that compares the scores of overlapped documents between the returned ranked lists.

- In the absence of overlap between the results, most metasearch merging techniques become ineffective.

Essential Results Merging References

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.

N. Craswell, D. Hawking, and P. Thistlewaite.
Merging results from isolated search engines.
In *Proceedings of the Australasian Database Conference*, pages 189-200, 1999.

E. Fox and J. Shaw.
Combination of multiple searches.
In *Proceedings of TREC*, pages 105-108,1994.

L. Gravano, C. Chang, H. Garcia-Molina, and A. Paepcke.
STARTS: Stanford proposal for internet meta-searching.
In *Proceedings of the ACM SIGMOD*, pages 207-218, 1997.

L. Si and J. Callan.
Using sampled data and regression to merge search engine results.
In *Proceedings of the ACM SIGIR*, pages 29-26. ACM, 2002.

M. Shokouhi and J. Zobel.
Robust result merging using sample-based score estimates.
In *ACM Transactions on Information Systems*, 27(3): 129, 2009.

B. Yuwono and D. Lee.
Server ranking for distributed text retrieval systems on the internet.
In *Proceedings of the Conference on Database Systems for Advanced Applications*, pages 41-50, 1997.

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

Questions?

Outline

1. Background
2. DIR: Introduction
3. DIR Architectures
4. Broker-Based DIR
5. DIR Evaluation
6. Applications of DIR

Topics Covered

5. DIR Evaluation
   - Objectives
   - Test Collections
   - Evaluation Metrics

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, her I report only some examples

Datasets available

Table 6.1 *Testbed statistics.*

Testbed	Size (GB)	# docs (×1000)			Size (MB)		
		Min	Avg	Max	Min	Avg	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 the 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

1. Background
2. DIR: Introduction
3. DIR Architectures
4. Broker-Based DIR
5. DIR Evaluation
6. Applications of DIR

Topics Covered

6. Applications of DIR
   - Vertical Search
   - Blog Distillation
   - Expert Search
   - Desktop Search

Vertical Search

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's query.

Idea

- Classification (does the query require a vertical search?)
- Resource Selection = Vertical Selection

Blog Distillation

Blog Distillation

The task of identifying blogs with a recurring central interest.

Idea

$$\text{Blog Feed} \iff \text{Posts}$$
$$\text{Federated Collection} \iff \text{Documents}$$

Expert Search

Expert Search

The task of identifying experts with a given expertise

Idea

Experts $\Longleftrightarrow$ documents authored by expert

Resource Selection on different collections of documents

Desktop Search

Desktop Search

The task of identifying different file and document types on a desktop relevant to a user query

Idea

Resource Selection on the type
Results Fusion on different documents

Questions?