Information Retrieval

Jaime Arguello
jarguell@email.unc.edu

Outline

Information Retrieval

Search Engine Components

Document Representation

Retrieval Models

Evaluation

Federated Search and Cross-lingual IR

Open-source Toolkits

What is Information Retrieval?

- Information retrieval (IR) is the science and practice of developing and evaluating systems that match information seekers with the information they seek.

What is Information Retrieval?

- Gerard Salton, 1968:

  Information retrieval is a field concerned with the structure, organization, and retrieval of information.

Information Retrieval

document structure

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact Wikipedia

Toolbox

Print/export

Languages
Deutsch
Español
Bahasa Indonesia

Gerard Salton

From Wikipedia, the free encyclopedia

Gerard Salton (8 March 1927 in Nuremberg - 28 August 1995), also known as Gerry Salton, was a Professor of Computer Science at Cornell University. Salton was perhaps the leading computer scientist working in the field of information retrieval during his time. His group at Cornell developed the SMART Information Retrieval System, which he initiated when he was at Harvard.

Salton was born Gerhard Anton Sahlmann on March 8, 1927 in Nuremberg, Germany. He received a Bachelor's (1950) and Master's (1952) degree in mathematics from Brooklyn College, and a Ph.D. from Harvard in Applied Mathematics in 1958, the last of Howard Aiken's doctoral students, and taught there until 1965, when he joined Cornell University and co-founded its department of Computer Science.

Salton was perhaps most well known for developing the now widely used Vector Space Model for Information Retrieval[1]. In this model, both documents and queries are represented as vectors of term counts, and the similarity between a document and a query is given by the cosine between the term vector and the document vector. In this paper, he also introduced TF-IDF, or term-frequency-inverse-document frequency, a model in which the score of a term in the a document is the ratio of the number of terms in that document divided by the frequency of the number of documents in which that term occurs. (The concept of inverse document frequency, a measure of specificity, had been introduced in 1972 by Karen Sparck-Jones[2].) Later in life, he became interested in automatic text summarization and analysis[3], as well as automatic hypertext generation[4]. He published over 150 research articles and 5 books during his life.

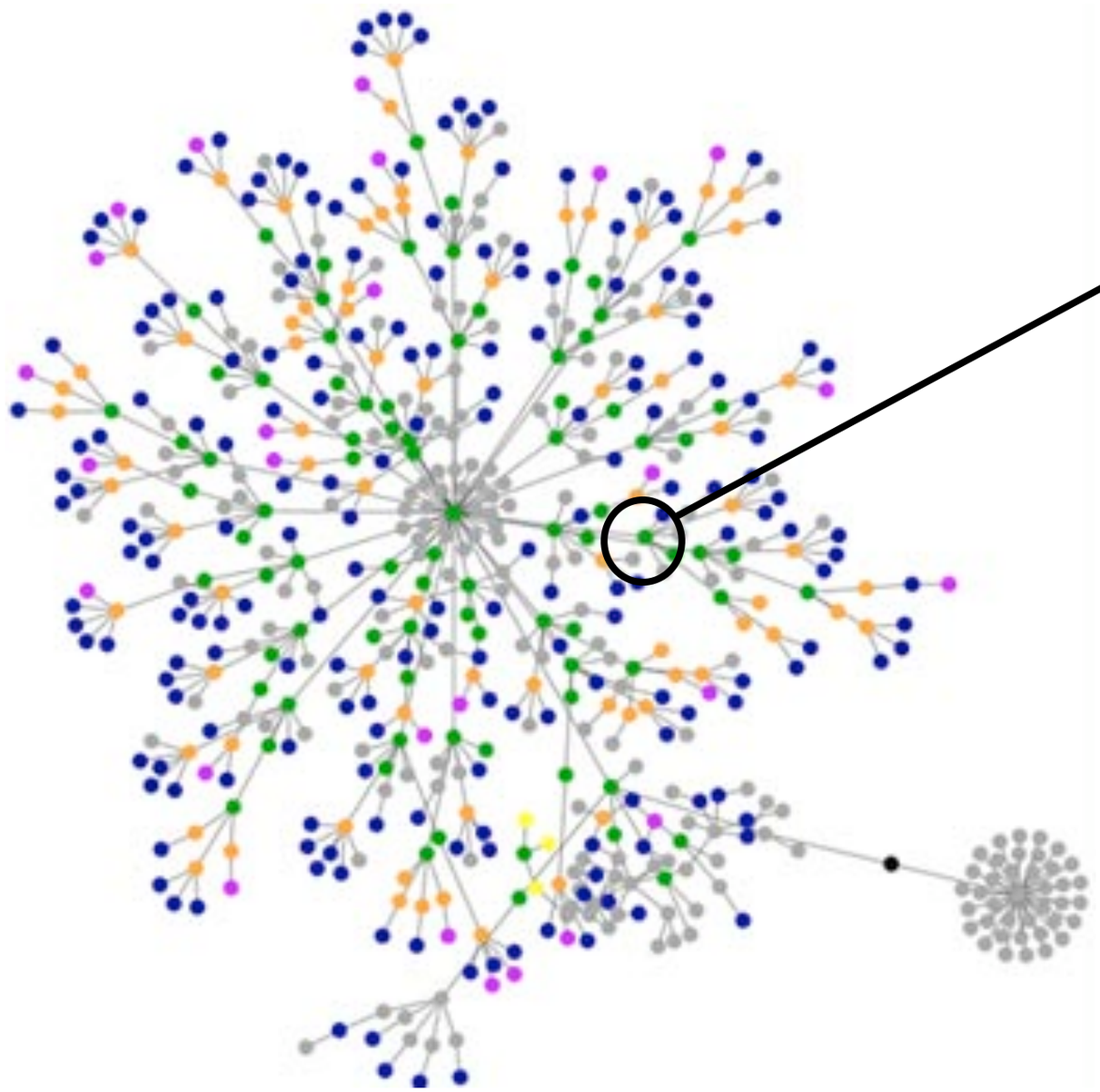
Salton was editor-in-chief of the Communications of the ACM and the Journal of the ACM, and chaired SIGIR. He was an associate editor of the ACM Transactions on Information Systems. He was an ACM Fellow (elected 1995), received an Award of Merit from the American Society for Information Science (1989), and was the first recipient of the SIGIR Award for outstanding contributions to study of information retrieval (1983) -- now called the Gerard Salton Award.

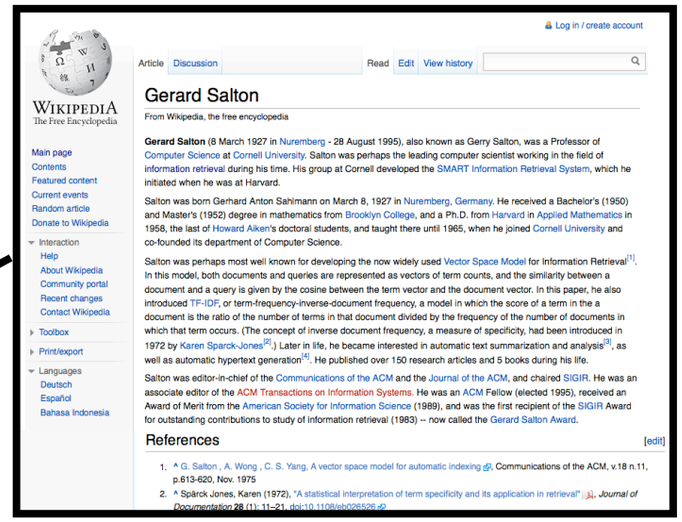
References [edit]

1. ^ G. Salton , A. Wong , C. S. Yang, A vector space model for automatic indexing, Communications of the ACM, v.18 n.11, p.613-620, Nov. 1975
2. ^ Spärck Jones, Karen (1972), "A statistical interpretation of term specificity and its application in retrieval", *Journal of Documentation* **28** (1): 11–21, doi:10.1108/eb026526

Information Retrieval

collection structure





Log in / create account

WIKIPEDIA
The Free Encyclopedia

Article Discussion Read Edit View history

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact Wikipedia
Toolbox
Print/export
Languages
Deutsch
Español
Bahasa Indonesia

Gerard Salton

From Wikipedia, the free encyclopedia

Gerard Salton (8 March 1927 in Nuremberg - 28 August 1995), also known as Gerry Salton, was a Professor of Computer Science at Cornell University. Salton was perhaps the leading computer scientist working in the field of information retrieval during his time. His group at Cornell developed the SMART Information Retrieval System, which he initiated when he was at Harvard.

Salton was born Gerhard Anton Salmann on March 8, 1927 in Nuremberg, Germany. He received a Bachelor's (1950) and Master's (1952) degree in mathematics from Brooklyn College, and a Ph.D. from Harvard in Applied Mathematics in 1958, the last of Howard Aiken's doctoral students, and taught there until 1965, when he joined Cornell University and co-founded its department of Computer Science.

Salton was perhaps most well known for developing the now widely used Vector Space Model for Information Retrieval[1]. In this model, both documents and queries are represented as vectors of term counts, and the similarity between a document and a query is given by the cosine between the term vector and the document vector. In this paper, he also introduced TF-IDF, or term-frequency-inverse-document frequency, a model in which the score of a term in the a document is the ratio of the number of terms in that document divided by the frequency of the number of documents in which that term occurs. (The concept of inverse document frequency, a measure of specificity, had been introduced in 1972 by Karen Sparck-Jones[2].) Later in life, he became interested in automatic text summarization and analysis[3], as well as automatic hypertext generation[4]. He published over 150 research articles and 5 books during his life.

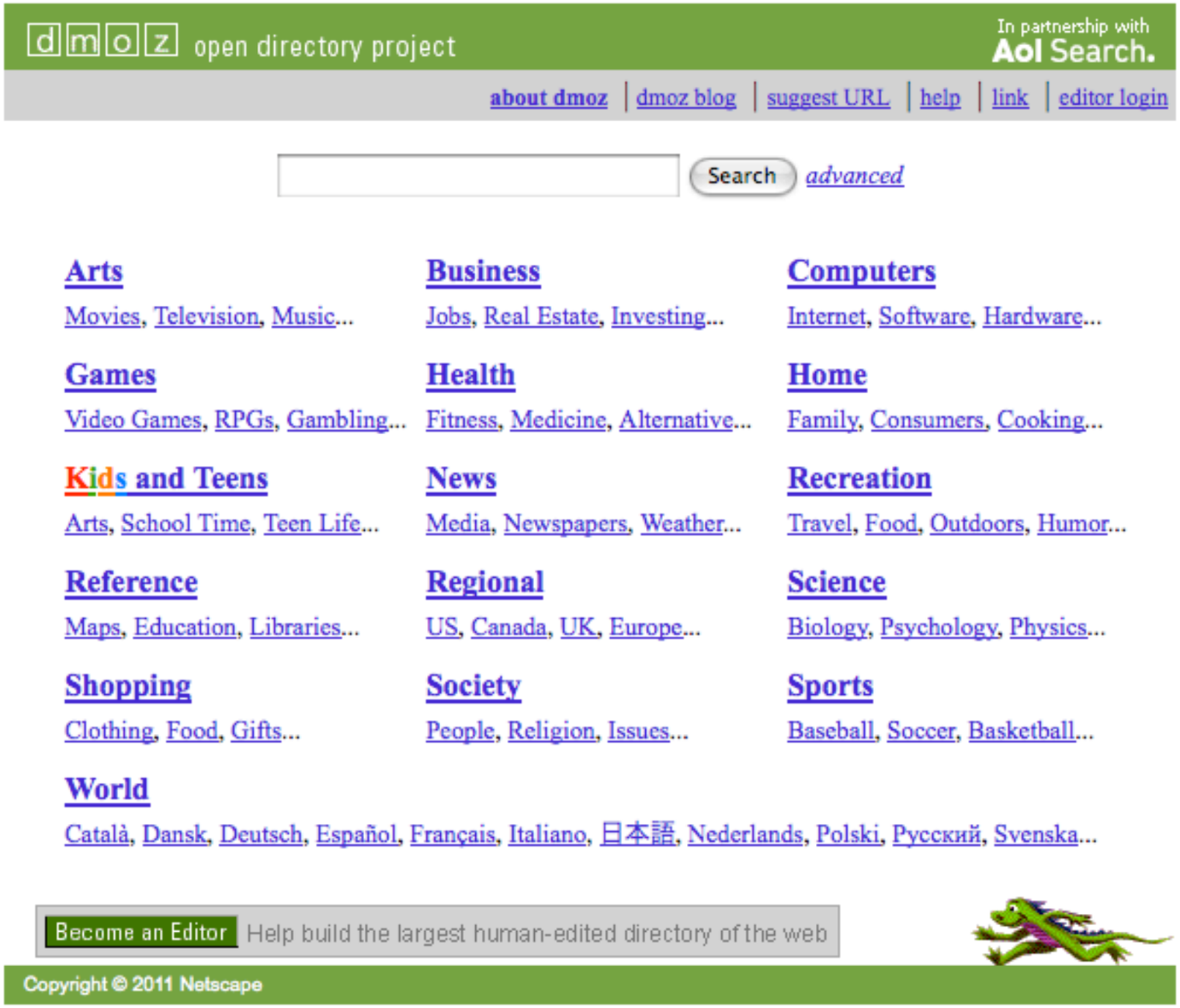
Salton was editor-in-chief of the Communications of the ACM and the Journal of the ACM, and chaired SIGIR. He was an associate editor of the ACM Transactions on Information Systems. He was an ACM Fellow (elected 1995), received an Award of Merit from the American Society for Information Science (1989), and was the first recipient of the SIGIR Award for outstanding contributions to study of information retrieval (1983) -- now called the Gerard Salton Award.

References [edit]

1. ^ G. Salton , A. Wong , C. S. Yang, A vector space model for automatic indexing, Communications of the ACM, v.18 n.11, p.613-620, Nov. 1975
2. ^ Spärck Jones, Karen (1972), "A statistical interpretation of term specificity and its application in retrieval", *Journal of Documentation* **28** (1): 11–21, doi:10.1108/eb026526

Information Retrieval

organization: cataloguing



http://www.dmoz.org

Information Retrieval

organization: recommendations

yelp
Real people. Real reviews.®

Search for (e.g. taco, cheap dinner, Max's): cosmic cantina
Near (Address, City, State or Zip): Chapel Hill, NC
Search

Welcome | About Me | Write a Review | **Find Reviews** | Invite Friends | Messaging | Talk | Events | Member Search

Cosmic Cantina

41 reviews | Rating Details

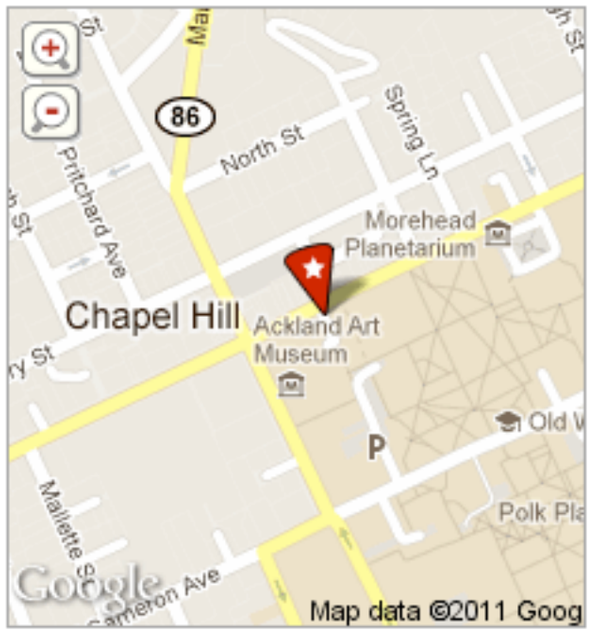
Category: Mexican [Edit]

128 E Franklin St
Chapel Hill, NC 27514

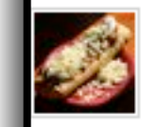
(919) 960-3955

Price Range: $
Accepts Credit Cards: Yes
Parking: Street
Attire: Casual
Good for Groups: No

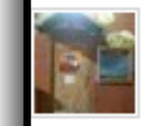
Good for Kids: Yes
Takes Reservations: No
Delivery: No
Take-out: Yes
Waiter Service: No



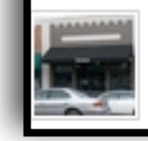
People Who Viewed This Also Viewed...


The Cosmic Cantina
84 reviews
Durham, NC


Carrburritos
93 reviews
Carrboro, NC


Joe's Joint
8 reviews
Chapel Hill, NC


Bandido's Mexican Cafe
20 reviews
Chapel Hill, NC


Pepper's Pizza
74 reviews
Chapel Hill, NC

http://www.yelp.com/biz/cosmic-cantina-chapel-hill
(not actual page)

Information Retrieval

retrieval

- Efficiency: retrieving results quickly
- Effectiveness: retrieving results that satisfy the user's information need

The Search Task

- Given a query and a corpus, find relevant items

  query: user's expression of their information need

  corpus: a repository of retrievable items

  relevance: satisfaction of the user's information need

Many Types of Search Engines

bing

twitter

okcupid

eHarmony

PANDORA

Google

Y!

match.com

ICON FINDER

mapquest mq

YAHOO! ANSWERS

flickr

Picasa

acm

LinkedIn

Westlaw

The New York Times

JSTOR

yelp

YouTube

Broadcast Yourself™

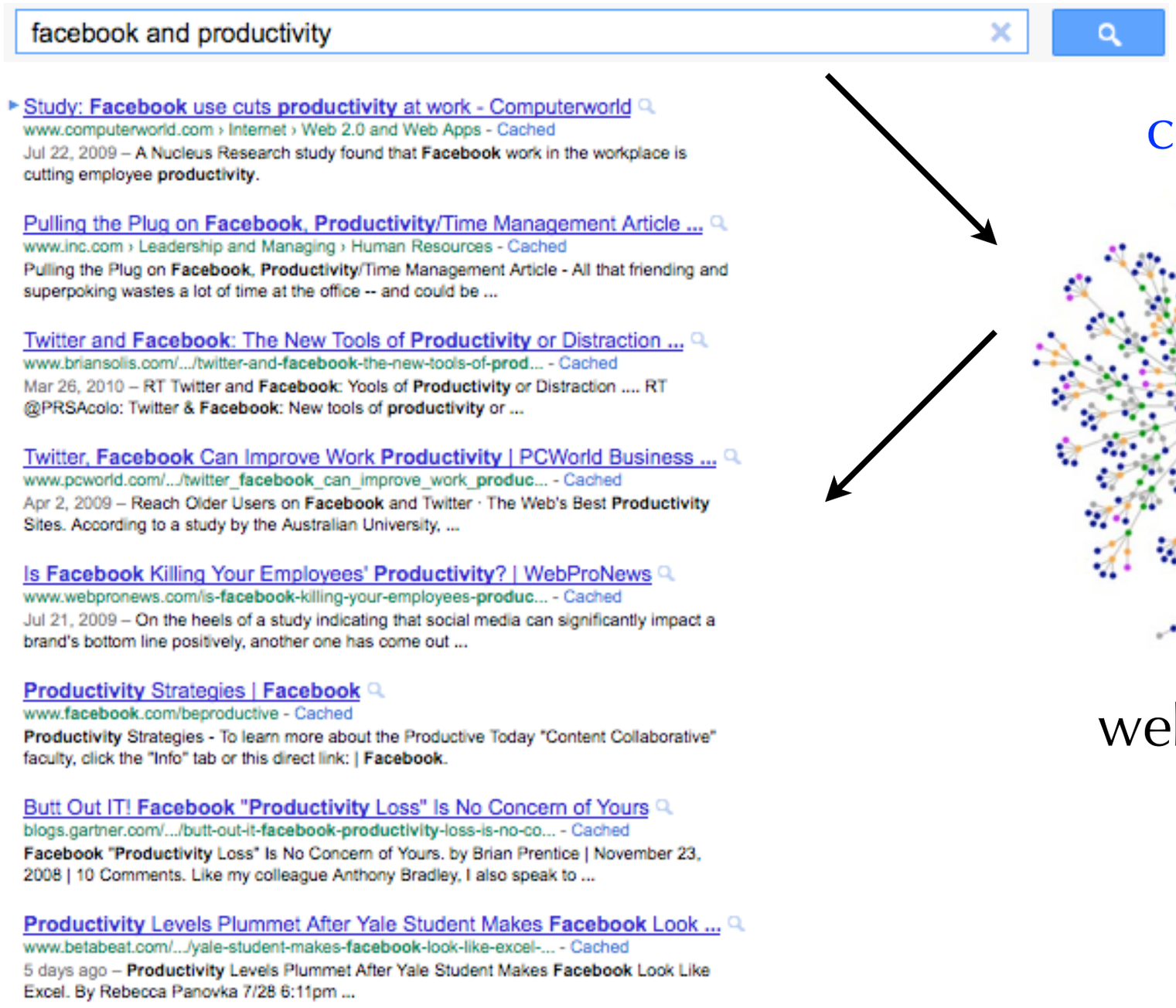
LexisNexis

Search Engines

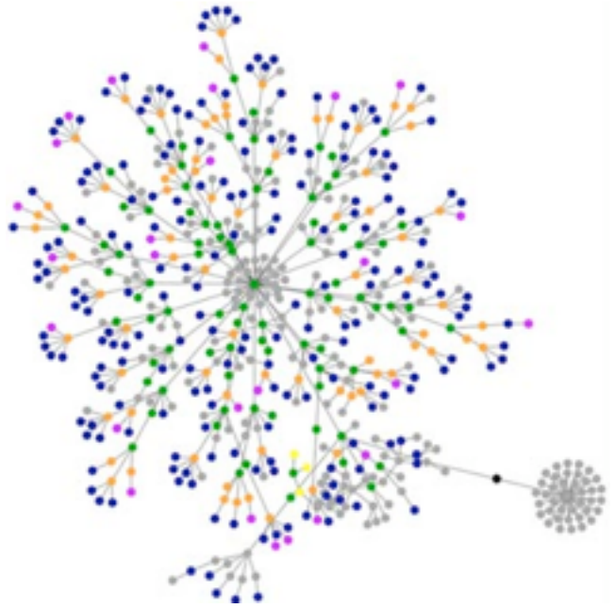
web search

query

results



corpus



web pages

Search Engines
digital library search

query

results

facebook productivity SEARCH

1 Effective teaching practices using free Google services: conference tutorial
Paul Gestwicki, Brian McNely
October 2010 **Journal of Computing Sciences in Colleges** , Volume 26 Issue 1
Publisher: Consortium for Computing Sciences in Colleges
Full text available: Pdf (22.76 KB)
Bibliometrics: Downloads (6 Weeks): 2, Downloads (12 Months): 48, Downloads (Overall): 48,

In this 90-minute tutorial, we will share our experiences using free Web services from Googl teaching effectiveness. Participants will engage with these services as part of the tutorial. We have used and studied these technologies, ...

2 Model-Based Engineering of Software: Three Productivity Perspectives
Shawn A. Bohner, Sriram Mohan
October 2009 **SEW '09:** Proceedings of the 2009 33rd Annual IEEE Software Engineering Workshop
Publisher: IEEE Computer Society
Full text available: Publisher Site
Bibliometrics: Downloads (6 Weeks): n/a, Downloads (12 Months): n/a, Downloads (Overall): n/a, Citation Count: 0

Evolving software products is a tricky business, especially when the domain is complex and changing rapidly. Like other fields of engineering, software engineering productivity advances have come about largely through abstraction reuse, process, and ...
Keywords: Agent-Based Software Systems, Model-Driven Architecture, Model-Driven Development, Model-Based Software Development, Model-Based Software Engineering

3 Absolute Beginner's Guide to Computer Basics, 5th edition
Michael Miller
September 2009 Absolute Beginner's Guide to Computer Basics, 5th edition
Publisher: Que Publishing Company
Bibliometrics: Downloads (6 Weeks): n/a, Downloads (12 Months): n/a, Downloads (Overall): n/a, Citation Count: 0

Everything casual users need to know to get the most out of their new Windows 7 PCs, software, and the Internet The best-selling beginner's guide, now completely updated for Windows 7 and today's most popular Internet tools -

corpus

ACM DL DIGITAL LIBRARY

scientific publications

Search Engines

news search

query

results

emily

Tropical Storm **Emily** Heads Toward Haiti, Dominican Republic

Voice of America - 23 minutes ago

August 03, 2011 Tropical Storm **Emily** Heads Toward Haiti, Dominican Republic VOA News Tropical storm warnings and watches are posted for parts of the ...

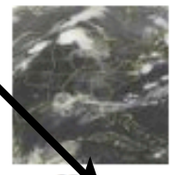
Video: Tropical Storm Emily on Path Toward Haiti The Associated Press

Local: **Emily** could reach hurricane strength Sarasota Herald-Tribune (blog)

Blog: Mubarak Trial Begins; Tropical Storm **Emily** Threatens East Coast NPR (blog)

The Guardian - Fox News

all 1804 news articles »



Caribbean

Emily Heads For Hispaniola

WAVY-TV (blog) - Jeremy Wheeler - 2 hours ago

Emily has had little change in strength since yesterday. She has winds of 50mph. The pressure is down a little to 1003 mb (millibars of pressure). ...

Tropical Storm **Emily** remains weak, hits Haiti tonight Examiner.com

August 2, 2011 Midday Tropics Update: TS **Emily** Now Moving More ... Wakulla.com

Emily the Effervescent Caribbean Hurricane Network

all 4 news articles »

'Snow White' Writer to Pen Universal's '**Emily** the Strange' (Exclusive)

Hollywood Reporter - Borys Kit - 13 hours ago

Melisa Wallack, who wrote the script that became Relativity's high-profile Snow White project, has been drafted by Universal to pen **Emily** the Strange. ...

Early Edition: **Emily** the Strange to Darken Big Screen; More News Moviefone (blog)

Writer Melisa Wallack Will Follow SNOW WHITE With **EMILY** THE STRANGE Collider.com

Details on the upcoming '**Emily** the Strange' movie Examiner.com

ComingSoon.net - 411mania.com

all 7 news articles »

Collider.com

High heat in Midwest and South

Reuters - Tim Sharp - Kevin Murphy - 22 hours ago

By Wendell Marsh WASHINGTON (Reuters) - Record-breaking heat continued to broil central and southern states on Tuesday as Tropical Storm **Emily** threatened to ...

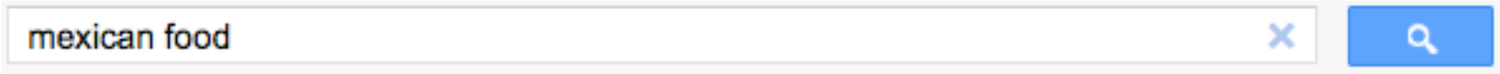
Reuters

corpus

news articles

Search Engines
local business search

query



results

Places for **mexican food** near Chapel Hill, NC

A Bandido's **Mexican** Cafe & Cantina - 14 reviews - Place page
www.bandidoscafe.com - 159 1/2 East Franklin Street, Chapel Hill - (919) 967-5048

B Las Potrillos **Mexican Restaurant** - 9 reviews - Place page
www.lospotrillos.net - 220 West Rosemary Street, Chapel Hill - (919) 932-4301

C monterrey **mexican restaurant** - 17 reviews - Place page
monterreychapelhill.com - 237 South Elliot Road, Chapel Hill - (919) 969-8750

D Margaret's Cantina - 19 reviews - Place page
www.margaretscantina.com - 1129 Weaver Dairy Road, Chapel Hill - (919) 942-4745

E Qdoba **Mexican** Grill - 19 reviews - Place page
www.qdoba.com - 100 West Franklin Street, Chapel Hill - (919) 929-8998

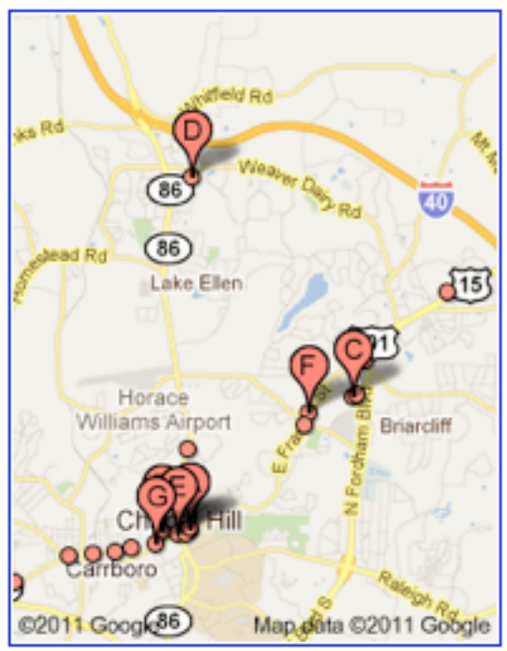
F Cinco de Mayo - 11 reviews - Place page
www.cincodemayorestaurants.net - 1502 East Franklin Street, Chapel Hill - (919) 929-6566

G Chipotle **Mexican** Grill - 15 reviews - Place page
www.chipotle.com - 301 W. Franklin St., Chapel Hill - (919) 942-2091

corpus

Google maps

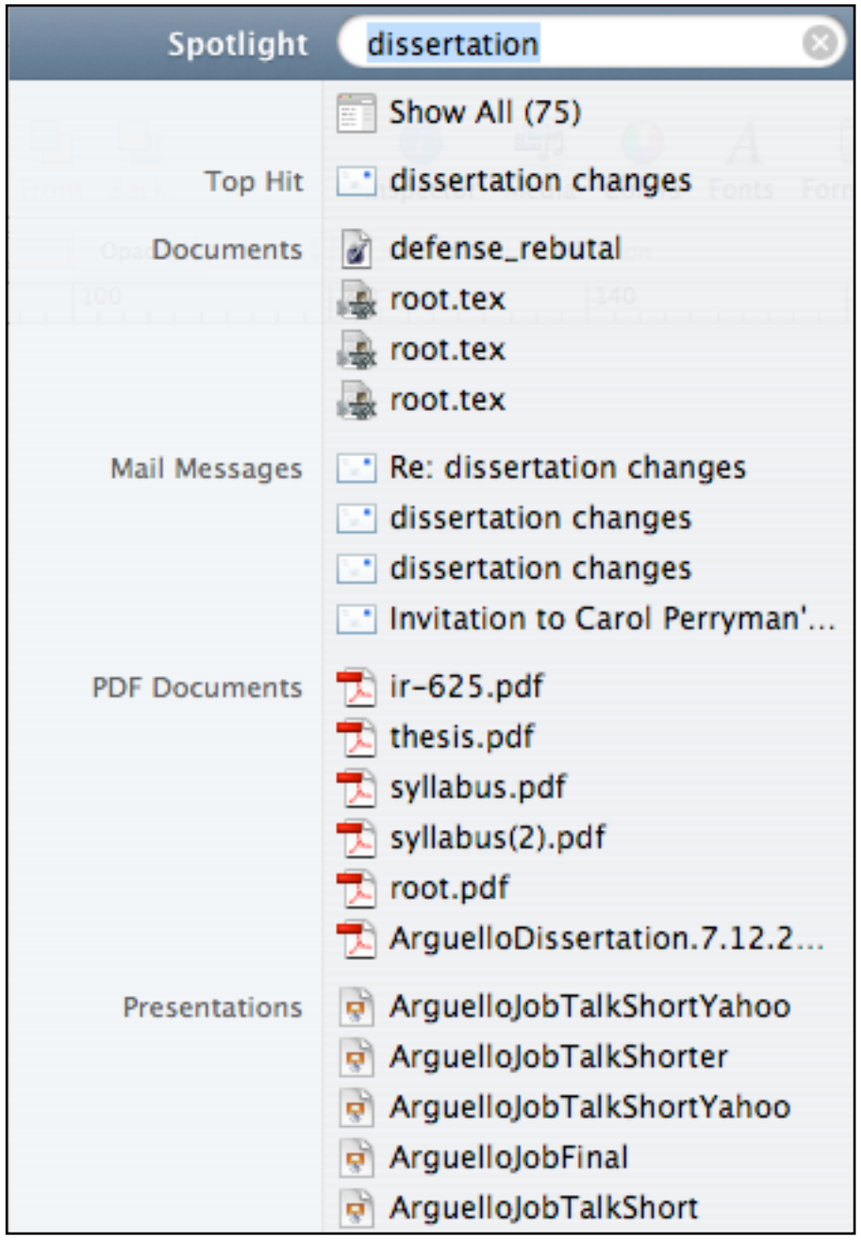
curated/synthesized business listings



Search Engines
desktop search

query

results



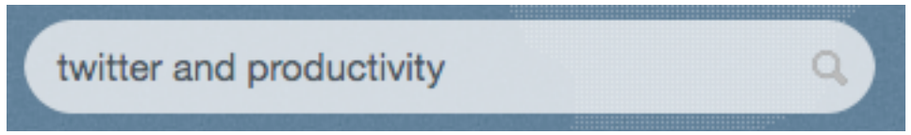
corpus

files in my laptop

Search Engines

micro-blog search

query



results

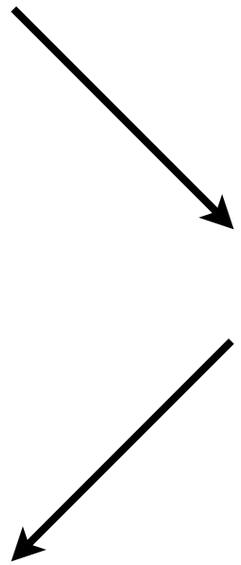
corpus

tweets

Search Engines
people/profile search

query

results



corpus

profiles

Information Retrieval Applications

digital library search

web search

enterprise search

news search

desktop search

question-answering

federated search

social search

local business search

image search

video search

(micro-)blog search

community Q&A search

expert search

product search

patent search

recommender systems

opinion mining

The Search Task
in this module

- Given a query and a corpus, find relevant items

  query: user's expression of their information need

  - a textual description of what the user wants

  corpus: a repository of retrievable items

  - a collection of textual documents

  relevance: satisfaction of the user's information need

  - the document contains information the user wants

Outline

Information Retrieval

Search Engine Components

Document Representation

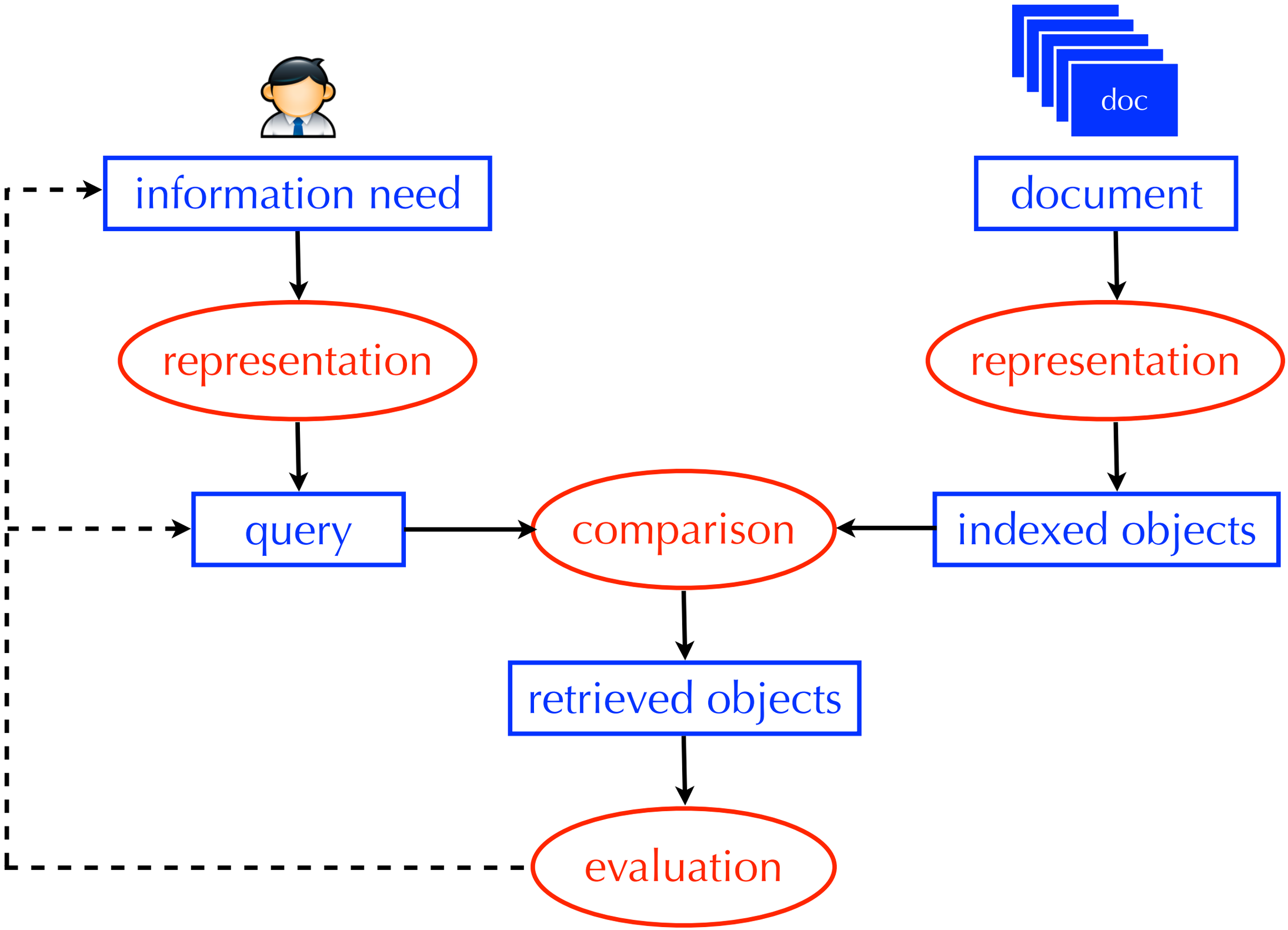
Retrieval Models

Evaluation

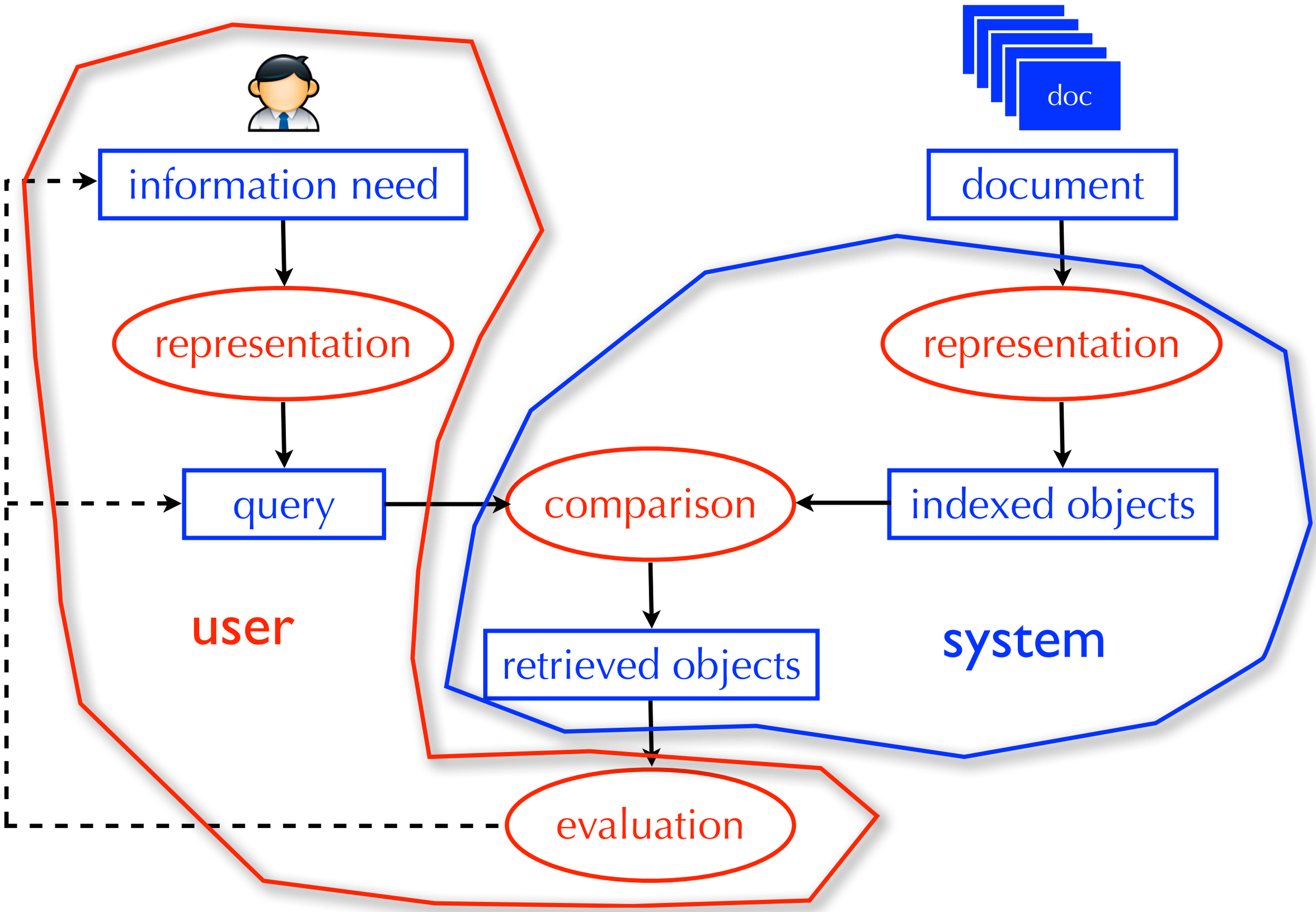
Federated Search and Cross-lingual IR

Open-source Toolkits

Information Retrieval Process



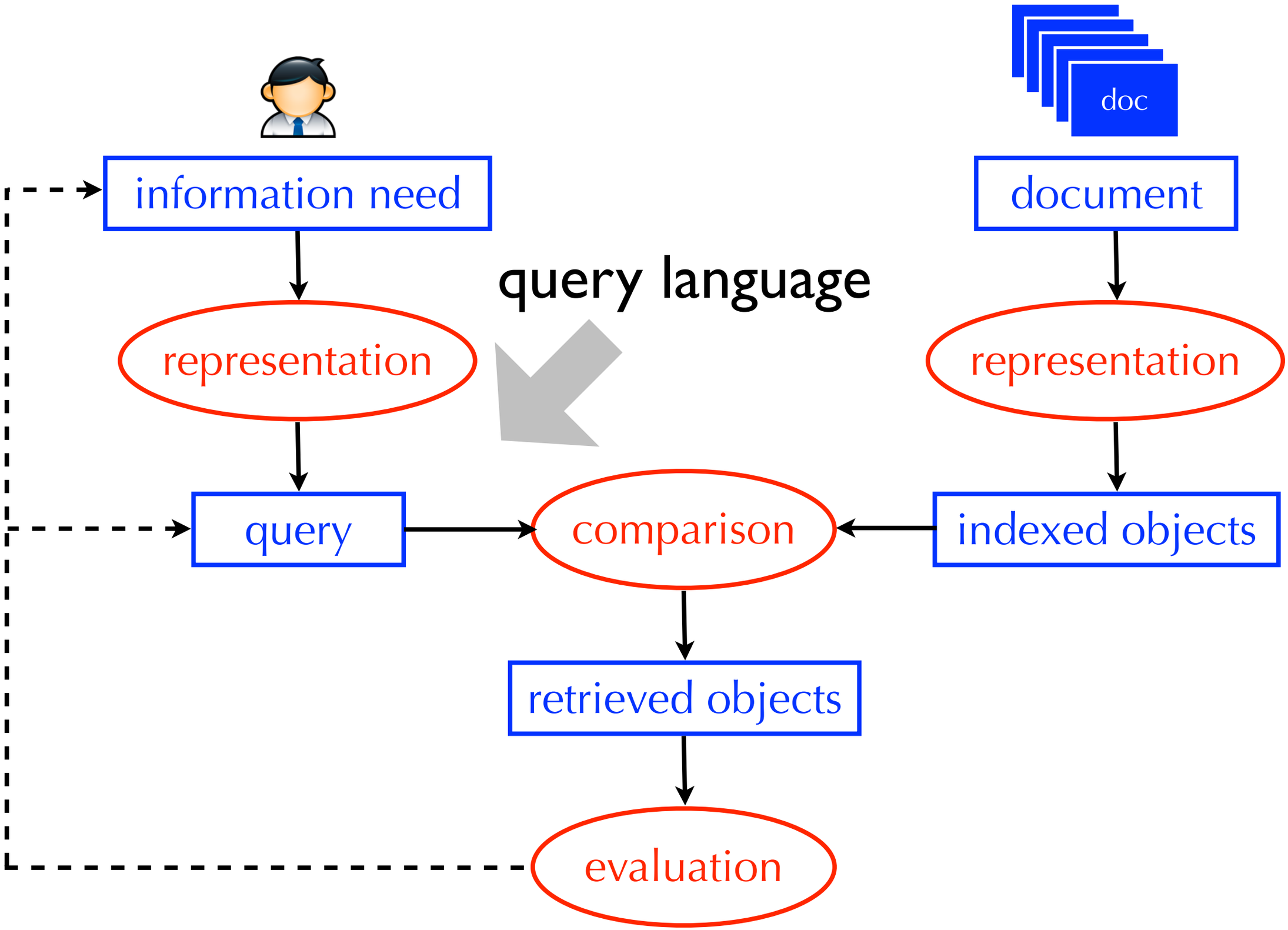
Information Retrieval Process



Search Engine Components

- Query language
- Document Representation and Indexing
- Retrieval Model

Information Retrieval Process

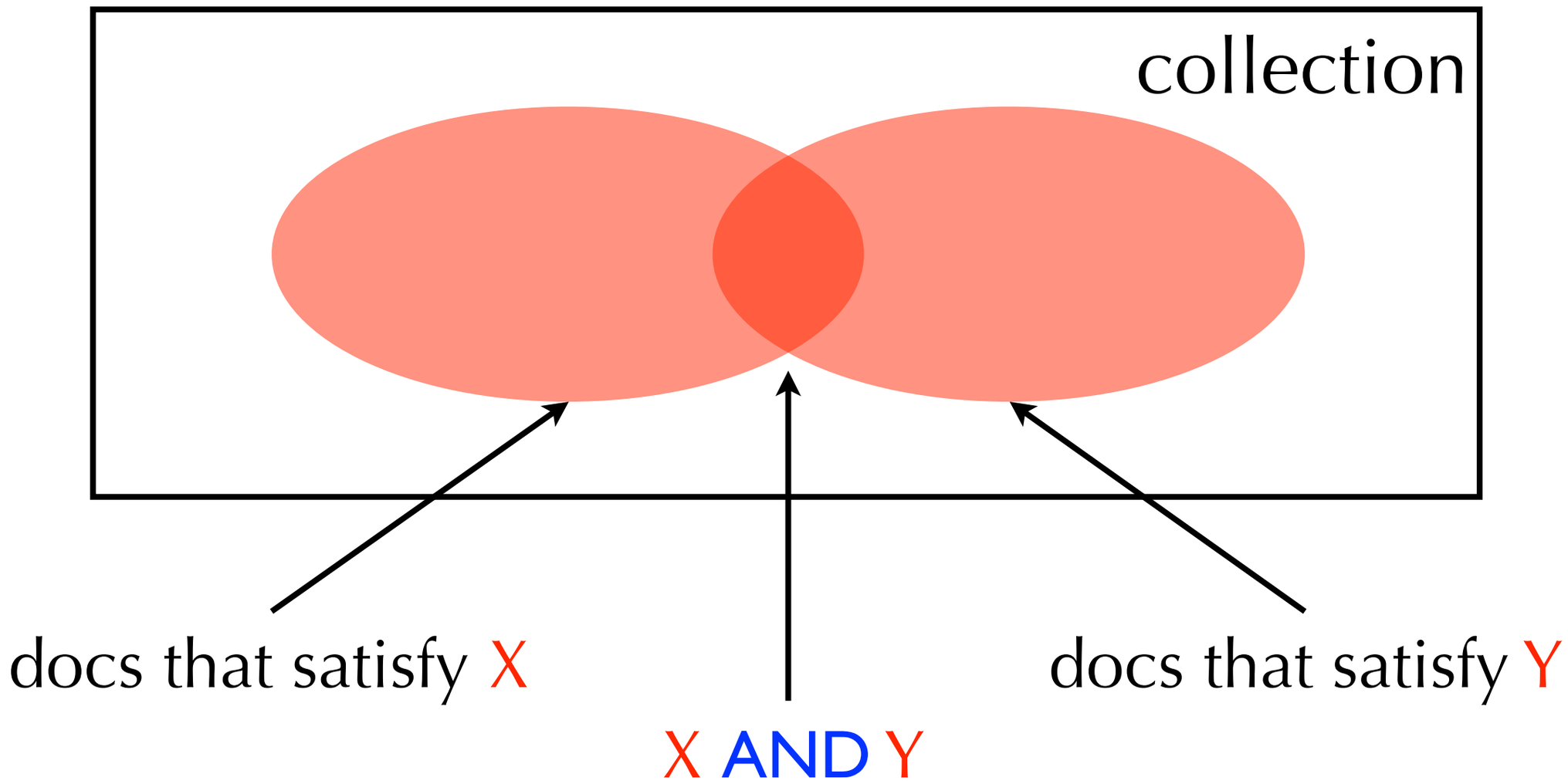


Boolean Query Language

- Assumption: the user can represent their information need using boolean constraints: AND, OR, and AND NOT
  - lincoln
  - president AND lincoln
  - president AND (lincoln OR abraham)
  - president AND (lincoln OR abraham) AND NOT car
  - president AND (lincoln OR abraham) AND NOT (car OR automobile)
- Parentheses specify the order of operations
  - A OR (B AND C) does not equal (A OR B) AND C

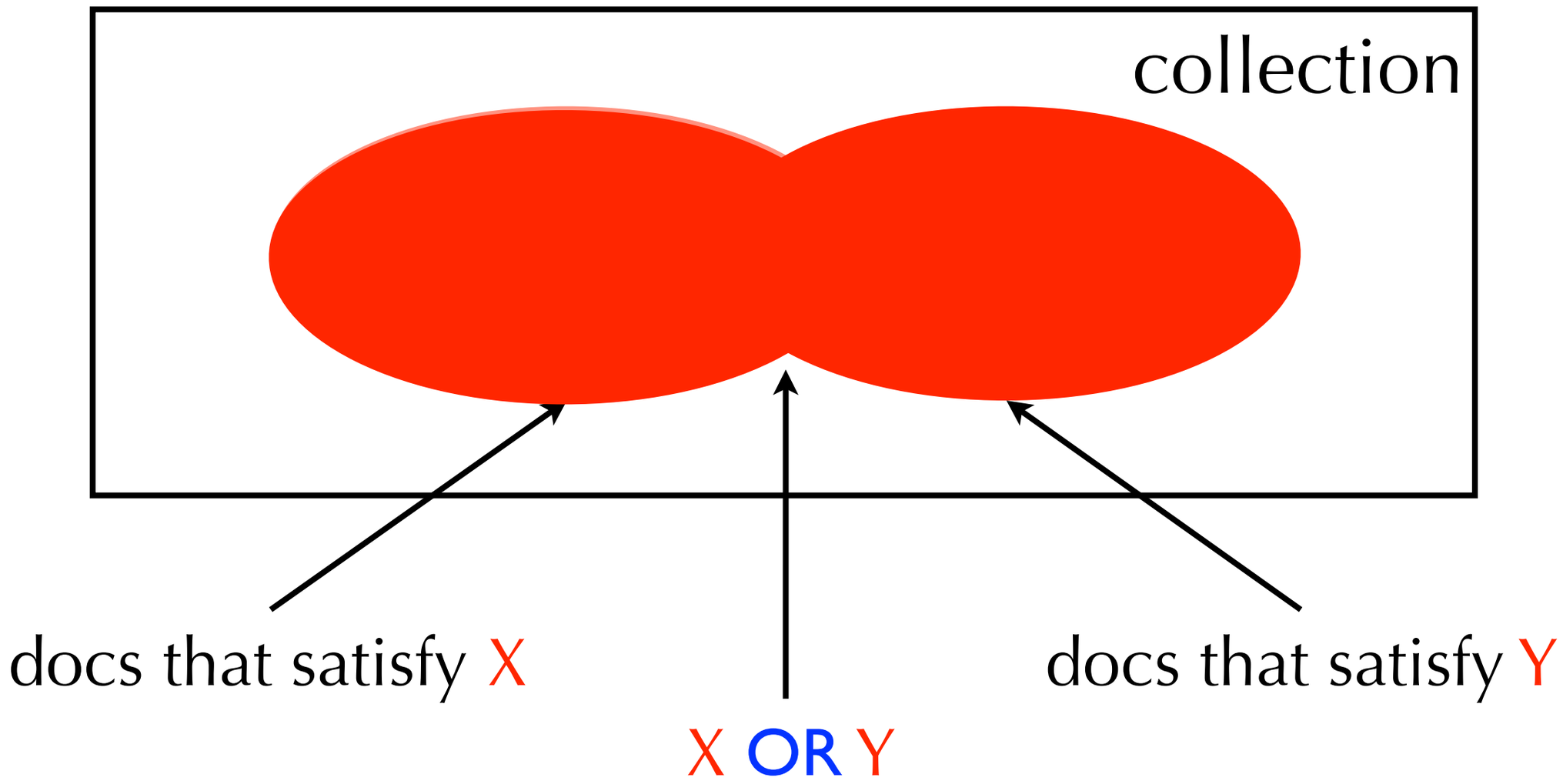
Boolean Query Language

- X AND Y



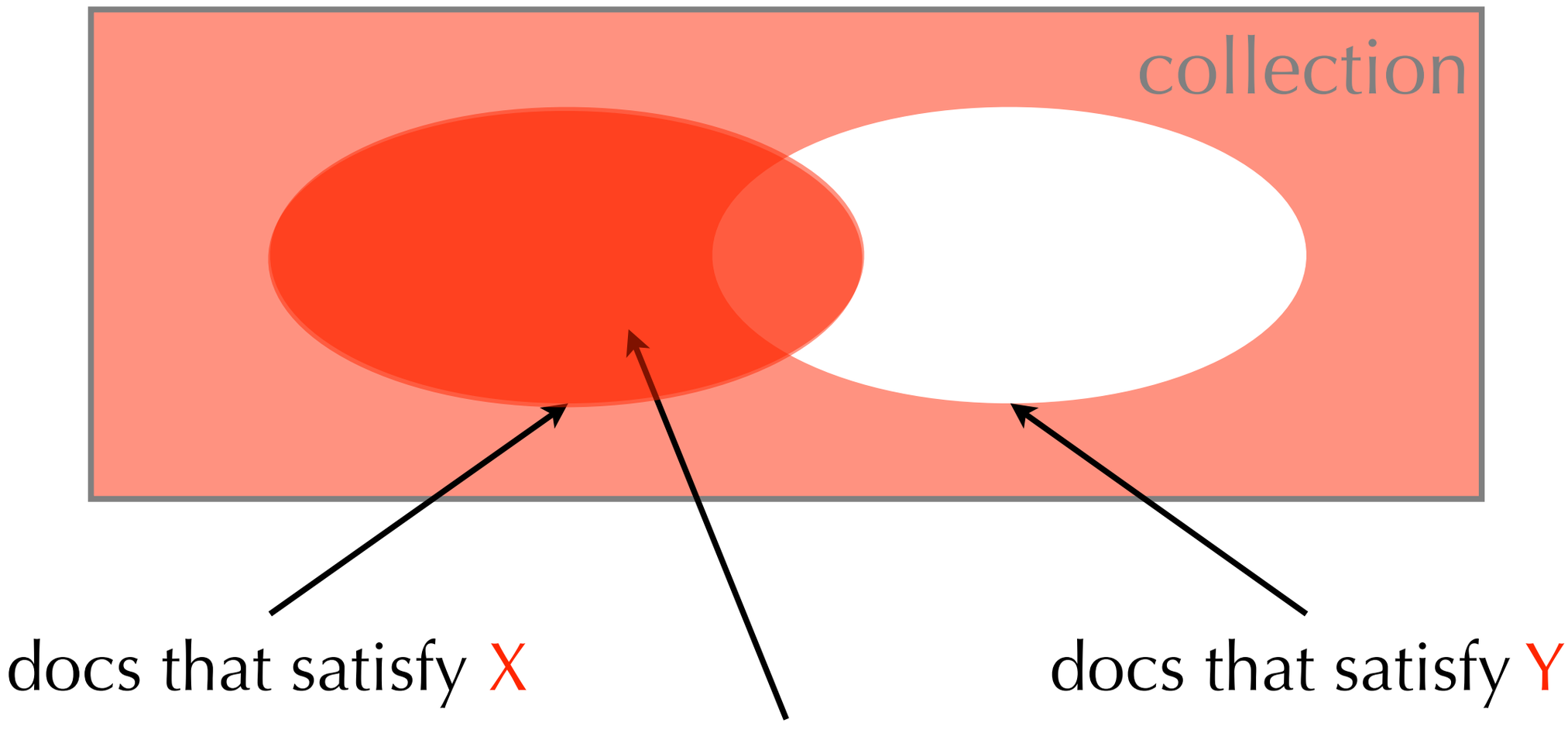
Boolean Query Language

- X OR Y



Boolean Query Language

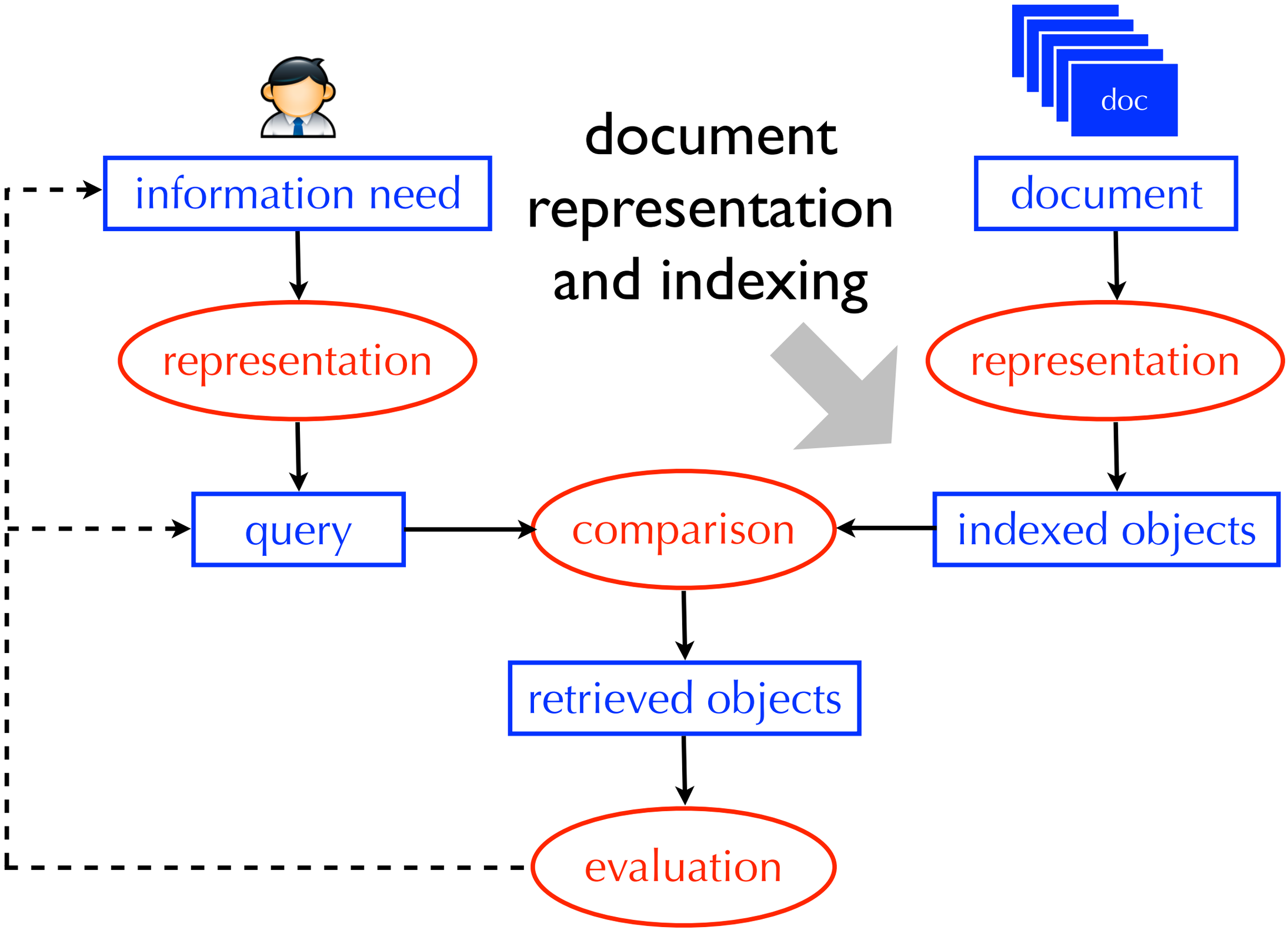
- X AND NOT Y



Boolean Query Language
advantages

- Easy for the system (no ambiguity in the query)
  - the burden is on the user to formulate the right query
- The user gets transparency and control
  - lots of results ➔ the query is too broad
  - no results ➔ the query is too narrow
- Common strategy for finding the right balance:
  - if the query is too broad, add AND or AND NOT constraints
  - if the query is too narrow, add OR constraints

Information Retrieval Process



Document Representation and Indexing

- A search engine uses an index to quickly locate documents that match the input query

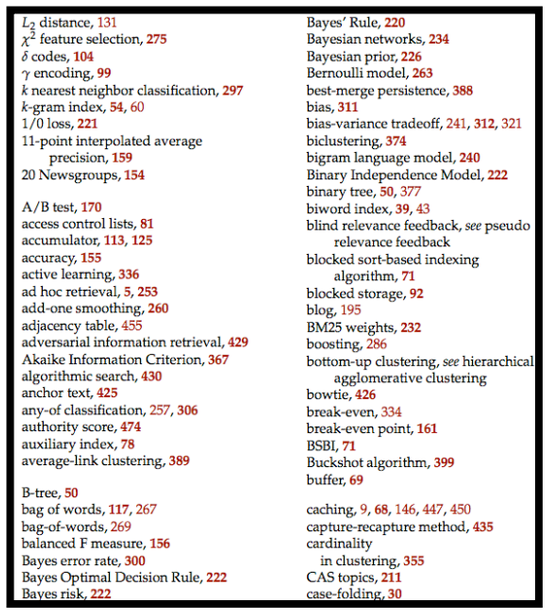
- Index: a list of concepts with pointers to the documents in the collection that discuss each concept

Index from Manning et al., 2008

Document Representation and Indexing

input query:

A/B testing



L_2 distance, 131
χ^2 feature selection, 275
δ codes, 104
γ encoding, 99
k nearest neighbor classification, 297
k-gram index, 54, 60
1/0 loss, 221
11-point interpolated average precision, 159
20 Newsgroups, 154

A/B test, 170
access control lists, 81
accumulator, 113, 125
accuracy, 155
active learning, 336
ad hoc retrieval, 5, 253
add-one smoothing, 260
adjacency table, 455
adversarial information retrieval, 429
Akaike Information Criterion, 367
algorithmic search, 430
anchor text, 425
any-of classification, 257, 306
authority score, 474
auxiliary index, 78
average-link clustering, 389

B-tree, 50
bag of words, 117, 267
bag-of-words, 269
balanced F measure, 156
Bayes error rate, 300
Bayes Optimal Decision Rule, 222
Bayes risk, 222
Bayes' Rule, 220
Bayesian networks, 234
Bayesian prior, 226
Bernoulli model, 263
best-merge persistence, 388
bias, 311
bias-variance tradeoff, 241, 312, 321
biclustering, 374
bigram language model, 240
Binary Independence Model, 222
binary tree, 50, 377
biword index, 39, 43
blind relevance feedback, *see* pseudo relevance feedback
blocked sort-based indexing algorithm, 71
blocked storage, 92
blog, 195
BM25 weights, 232
boosting, 286
bottom-up clustering, *see* hierarchical agglomerative clustering
bowtie, 426
break-even, 334
break-even point, 161
BSBI, 71
Buckshot algorithm, 399
buffer, 69

caching, 9, 68, 146, 447, 450
capture-recapture method, 435
cardinality
  in clustering, 355
CAS topics, 211
case-folding, 30

results:

170, 188, 2021

- So, what goes in the index is important!
- How might we combine concepts (e.g., A/B testing AND patent search)?

Document Representation

- Document representation: deciding what goes in the index
- Controlled vocabulary indexing: a set a manually constructed concepts that describe the major topics covered in the collection
- Free-text Indexing: the set of individual terms that occur in the collection

Controlled Vocabularies

example

NCBI Resources How To

MeSH MeSH light therapy Search

Phototherapy

Treatment of disease by exposure to light, especially by variously concentrated light rays or specific wavelengths.

Year introduced: 1981

PubMed search builder options

Subheadings:

sub-headings

- adverse effects
- classification
- contraindications
- economics
- history
- instrumentation
- methods
- nursing
- psychology
- standards
- statistics and numerical data
- supply and distribution
- trends
- utilization
- veterinary

All MeSH Categories

Analytical, Diagnostic and Therapeutic Techniques and Equipment Category

Therapeutics

Phototherapy

Color Therapy

Heliotherapy

Laser Therapy, Low-Level

Photochemotherapy

Hematoporphyrin Photoradiation

Ultraviolet Therapy

PUVA Therapy +

sub-tree within the hierarchy

Entry Terms:

- Phototherapies
- Therapy, Photoradiation
- Photoradiation Therapies
- Therapies, Photoradiation
- **Light Therapy**
- Light Therapies
- Therapies, Light
- Therapy, Light
- Photoradiation Therapy

entry-terms

Controlled Vocabularies

example

NCBI Resources How To

PubMed.gov
US National Library of Medicine
National Institutes of Health

PubMed | phototherapy/adverse effects | Search

RSS Save search Limits Advanced

Results: 1 to 20 of 2697

<< First < Prev Page 1 of 135 Next > Last >>

Burning daylight: balancing vitamin D requirements with sensible sun exposure.
1. Stalgis-Bilinski KL, Boyages J, Salisbury EL, Dunstan CR, Henderson SI, Talbot PL.
Med J Aust. 2011 Apr 4;194(7):345-8.
PMID: 21470084 [PubMed - indexed for MEDLINE]
Related citations

Time-lag between subretinal fluid and pigment epithelial detachment reduction after polypoidal choroidal vasculopathy treatment.
2. Chae JB, Lee JY, Yang SJ, Kim JG, Yoon YH.
Korean J Ophthalmol. 2011 Apr;25(2):98-104. Epub 2011 Mar 11.
PMID: 21461221 [PubMed - indexed for MEDLINE] **Free PMC Article**
Free full text Related citations

Metal stenting to resolve post-photodynamic therapy stricture in early esophageal cancer.
3. Cheon YK.
World J Gastroenterol. 2011 Mar 14;17(10):1379-82.
PMID: 21455341 [PubMed - indexed for MEDLINE] **Free PMC Article**
Free full text Related citations

A study of multiple full-face treatments with low-energy settings of a 2940-nm Er:YAG fractionated laser.
4. Goldberg DJ, Hussain M.
J Cosmet Laser Ther. 2011 Apr;13(2):42-6.
PMID: 21401375 [PubMed - indexed for MEDLINE]
Related citations

Controlled Vocabularies

example

Burning daylight: balancing vitamin D requirements with sensible sun exposure.

Stalgis-Bilinski KL, Boyages J, Salisbury EL, Dunstan CR, Henderson SI, Talbot PL.

Westmead Breast Cancer Institute, University of Sydney, Sydney, NSW, Australia. Kellie.Bilinski@bci.org.au

Abstract

OBJECTIVE: To examine the feasibility of balancing sunlight exposure to meet vitamin D requirements with sun protection guidelines.

DESIGN AND SETTING: We used standard erythemal dose and Ultraviolet Index (UVI) data for 1 June 1996 to 30 December 2005 for seven Australian cities to estimate duration of sun exposure required for fair-skinned individuals to synthesise 1000 IU (25 μg) of vitamin D, with 11% and 17% body exposure, for each season and hour of the day. Periods were classified according to whether the UVI was < 3 or ≥ 3 (when sun protection measures are recommended), and whether required duration of exposure was ≤ 30 min, 31-60 min, or > 60 min.

MAIN OUTCOME MEASURE: Duration of sunlight exposure required to achieve 1000 IU of vitamin D synthesis.

RESULTS: Duration of sunlight exposure required to synthesise 1000 IU of vitamin D varied by time of day, season and city. Although peak UVI periods are typically promoted as between 10 am and 3 pm, UVI was often ≥ 3 before 10 am or after 3 pm. When the UVI was < 3, there were few opportunities to synthesise 1000 IU of vitamin D within 30 min, with either 11% or 17% body exposure.

CONCLUSION: There is a delicate line between balancing the beneficial effects of sunlight exposure while avoiding its damaging effects. Physiological and geographical factors may reduce vitamin D synthesis, and supplementation may be necessary to achieve adequate vitamin D status for individuals at risk of deficiency.

MeSH Terms

Australia
Dose-Response Relationship, Radiation
Guideline Adherence
Health Policy*
Heliotherapy/adverse effects
Heliotherapy/methods*
Humans
Seasons
Skin Pigmentation
Sunlight/adverse effects*
Time Factors
Vitamin D/biosynthesis*
Vitamin D Deficiency/prevention & control*

Controlled Vocabularies

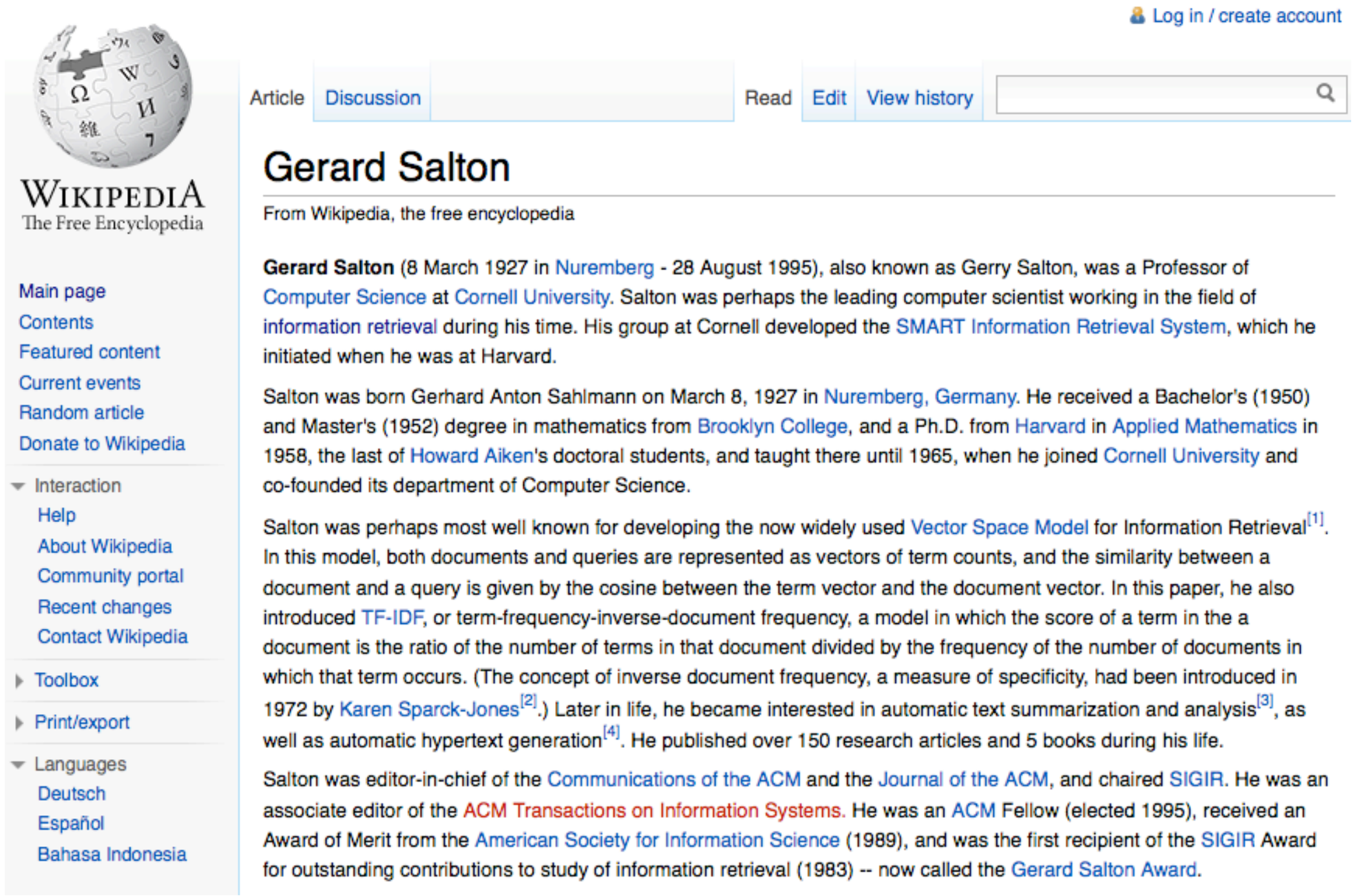
advantages

- Concepts do not need to appear explicitly in the text
- Describe the concepts that are most central to the document
- Relationships between concepts facilitate non-query-based navigation and exploration
- Developed by experts who know the data and the users
- Represent the concepts/relationships that users (presumably) care the most about
- Concepts are unambiguous and recognizable (necessary for annotators and good for users)

Free-text Indexing

- Represent documents using terms within the document
- Which terms? Only the most descriptive terms? Only the unambiguous ones? All of them?
- Usually, all of them (a.k.a. full-text indexing)
- The user will use term-combinations to express higher level concepts
- Query terms will hopefully disambiguate each other (e.g., “volkswagen golf”)
- The search engine will determine which terms are important (we’ll talk about this during “retrieval models”)

Free-text Indexing



Log in / create account

Article | Discussion | Read | Edit | View history

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact Wikipedia

Toolbox

Print/export

Languages
Deutsch
Español
Bahasa Indonesia

Gerard Salton

From Wikipedia, the free encyclopedia

Gerard Salton (8 March 1927 in Nuremberg - 28 August 1995), also known as Gerry Salton, was a Professor of Computer Science at Cornell University. Salton was perhaps the leading computer scientist working in the field of information retrieval during his time. His group at Cornell developed the SMART Information Retrieval System, which he initiated when he was at Harvard.

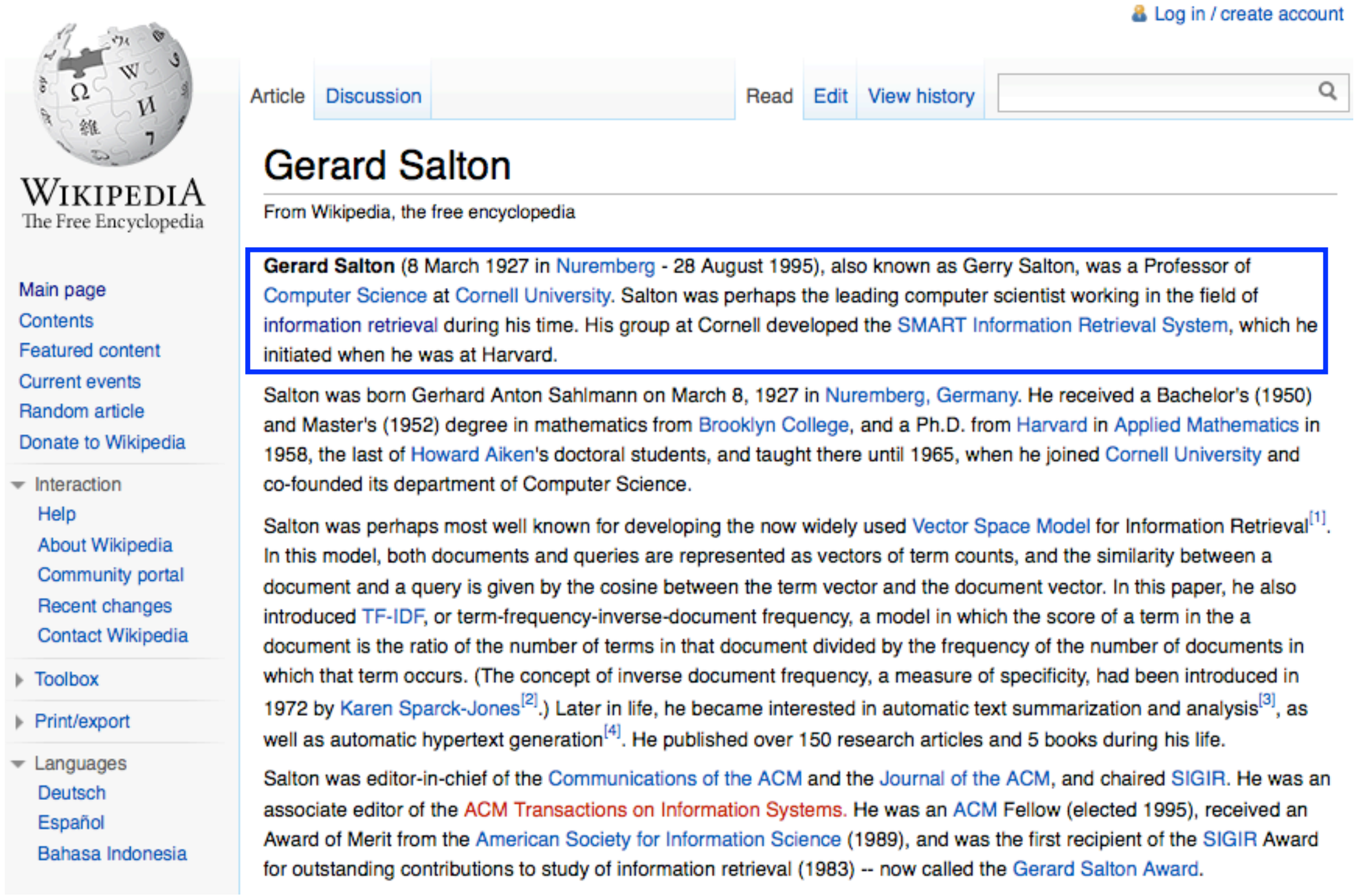
Salton was born Gerhard Anton Sahlmann on March 8, 1927 in Nuremberg, Germany. He received a Bachelor's (1950) and Master's (1952) degree in mathematics from Brooklyn College, and a Ph.D. from Harvard in Applied Mathematics in 1958, the last of Howard Aiken's doctoral students, and taught there until 1965, when he joined Cornell University and co-founded its department of Computer Science.

Salton was perhaps most well known for developing the now widely used Vector Space Model for Information Retrieval[1]. In this model, both documents and queries are represented as vectors of term counts, and the similarity between a document and a query is given by the cosine between the term vector and the document vector. In this paper, he also introduced TF-IDF, or term-frequency-inverse-document frequency, a model in which the score of a term in the a document is the ratio of the number of terms in that document divided by the frequency of the number of documents in which that term occurs. (The concept of inverse document frequency, a measure of specificity, had been introduced in 1972 by Karen Sparck-Jones[2].) Later in life, he became interested in automatic text summarization and analysis[3], as well as automatic hypertext generation[4]. He published over 150 research articles and 5 books during his life.

Salton was editor-in-chief of the Communications of the ACM and the Journal of the ACM, and chaired SIGIR. He was an associate editor of the ACM Transactions on Information Systems. He was an ACM Fellow (elected 1995), received an Award of Merit from the American Society for Information Science (1989), and was the first recipient of the SIGIR Award for outstanding contributions to study of information retrieval (1983) -- now called the Gerard Salton Award.

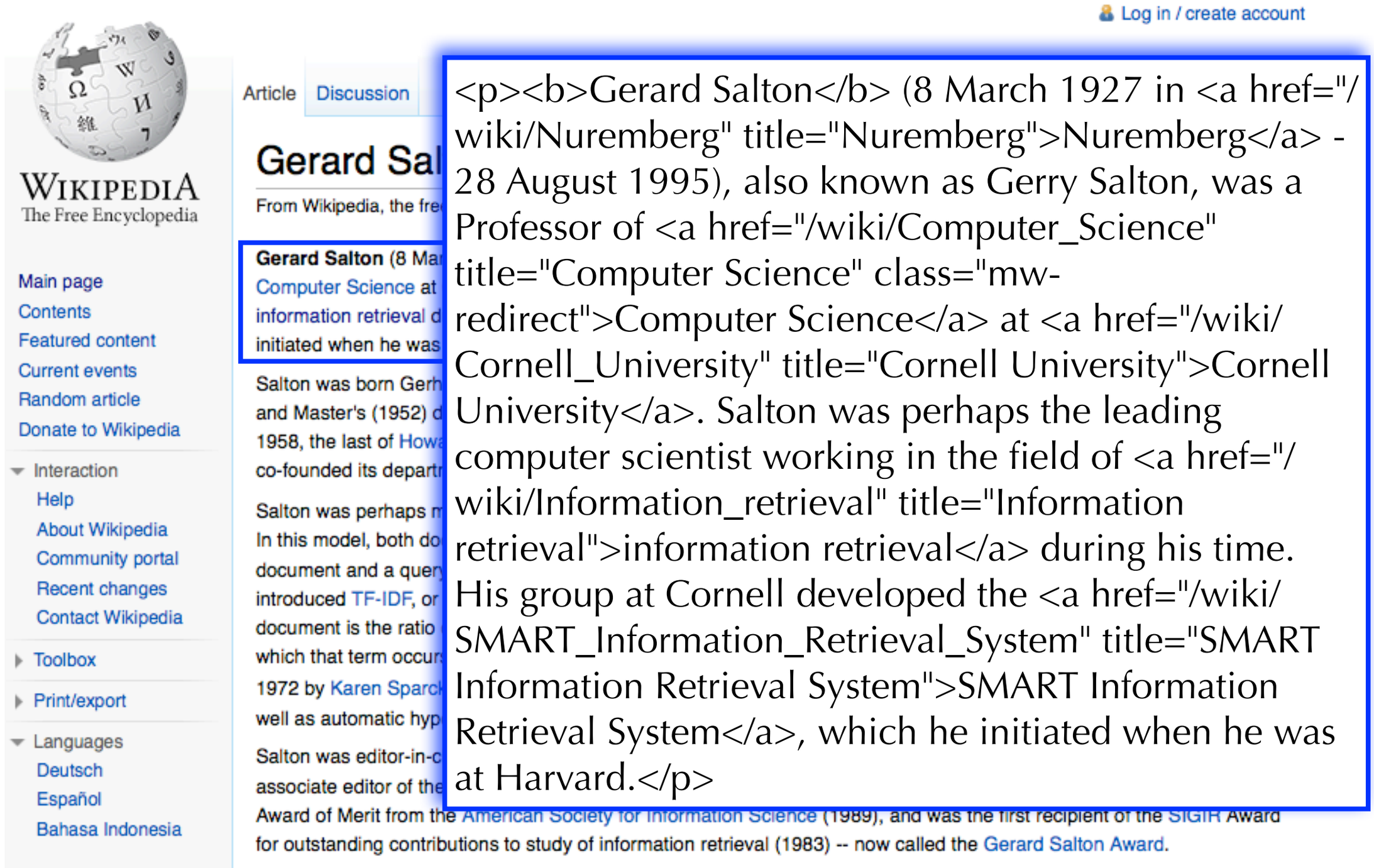
Free-text Indexing

what you see



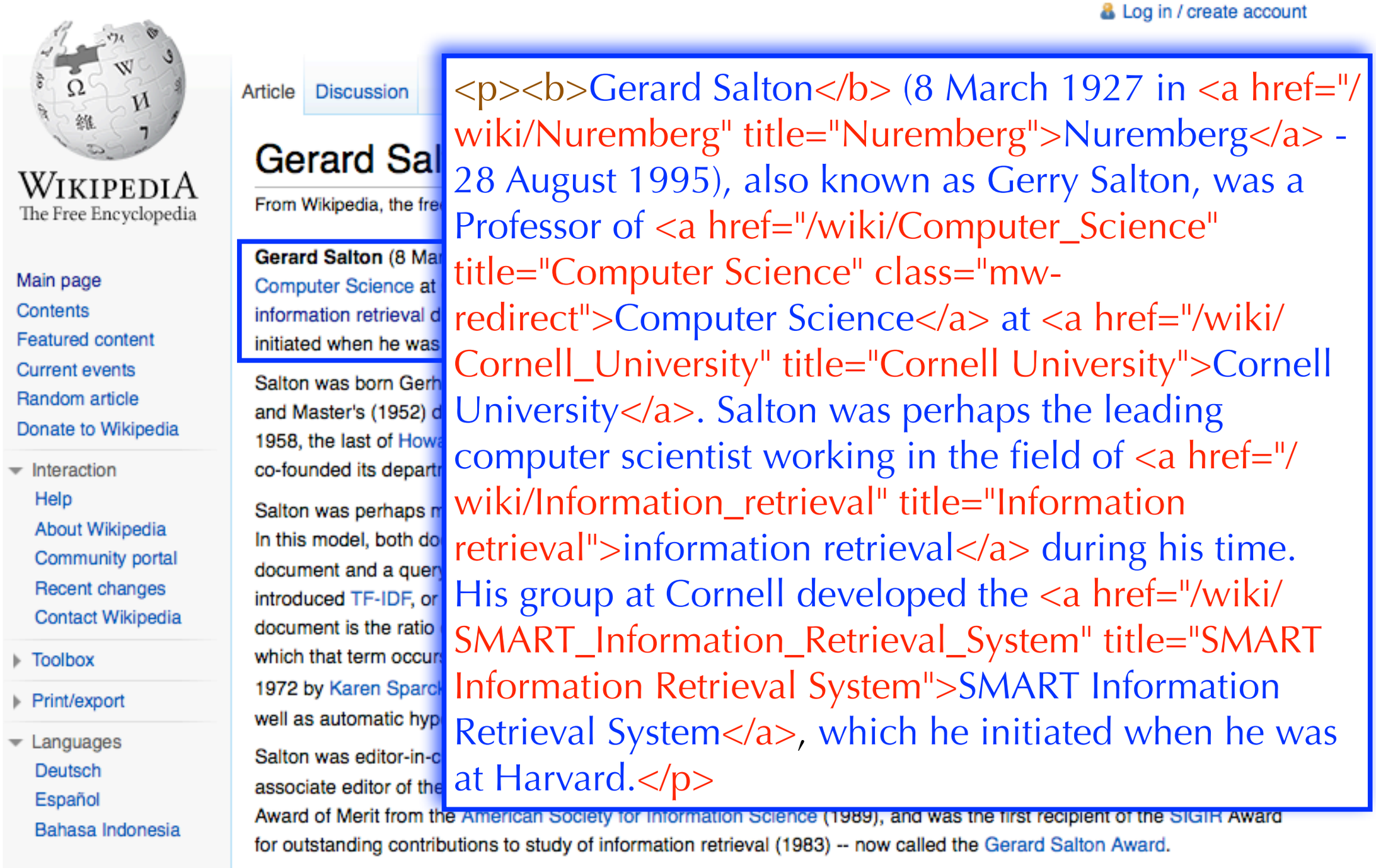
Free-text Indexing

what your computer sees



Free-text Indexing

mark-up vs. content



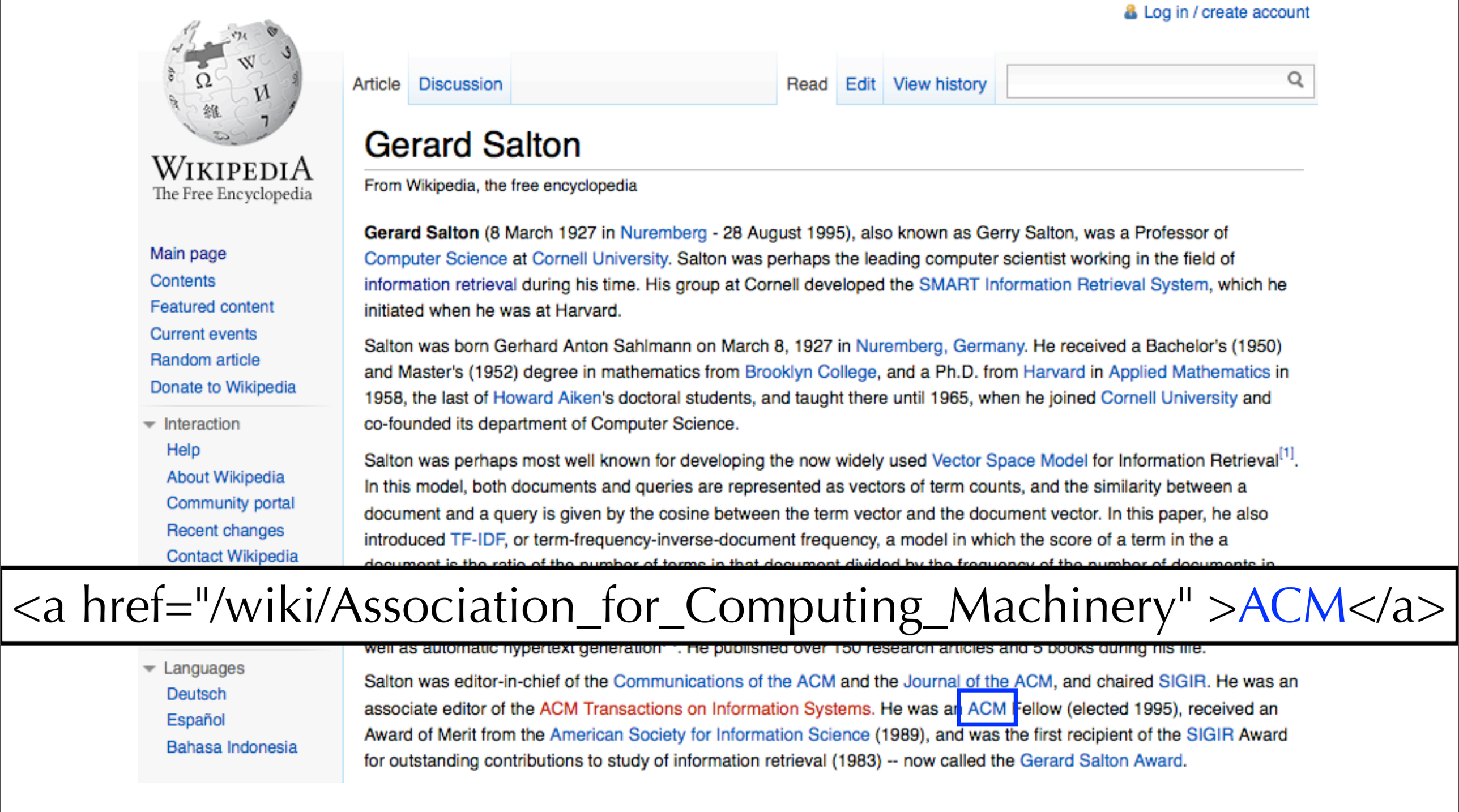
Free-text Indexing

mark-up

- Describes how the content should be presented
  - e.g., your browser interprets html mark-up and presents the page as intended by the author
- Can provide evidence of what text is important for search
- It may also provide useful, “unseen” information!

Free-text Indexing

mark-up



ACM

Free-text Indexing

text-processing

Gerard Salton (8 March 1927 in Nuremberg - 28 August 1995), also known as Gerry Salton, was a Professor of Computer Science at Cornell University. Salton was perhaps the leading computer scientist working in the field of information retrieval during his time. His group at Cornell developed the SMART Information Retrieval System, which he initiated when he was at Harvard.

- Step 1: mark-up removal

Free-text Indexing

text-processing

Gerard Salton (8 March 1927 in Nuremberg 28 August 1995), also known as Gerry Salton, was a Professor of Computer Science at Cornell University. Salton was perhaps the leading computer scientist working in the field of information retrieval during his time. His group at Cornell developed the SMART Information Retrieval System, which he initiated when he was at Harvard.

- Step 1: mark-up removal

Free-text Indexing

text-processing

gerard salton (8 march 1927 in nuremberg – 28 august 1995), also known as gerry salton, was a Professor of computer science at cornell university. salton was perhaps the leading computer scientist working in the field of information retrieval during his time. his group at cornell developed the smart information retrieval system, which he initiated when he was at harvard.

- Step 2: down-casing
- Can change a word's meaning, but we do it anyway
  - Information = information ???
  - SMART = smart ???

Free-text Indexing

text-processing

gerard salton 8 march 1978 in nuremberg 28 august 1995 also know as gerry salton was professor of computer science at cornell university salton was perhaps the leading computer scientist working in the field of information retrieval during his time his group at cornell developed the smart information retrieval system which he initiated when he was at harvard

- Step 3: tokenization
- Tokenization: splitting text into words (in this case, based on sequences of non-alphanumeric characters)
- Problematic cases: ph.d. = pd d, isn't = isn t

Free-text Indexing

text-processing

gerard salton 8 march 1978 in nuremberg 28 august 1995 also know as gerry salton was professor of computer science at cornell university salton was perhaps the leading computer scientist working in the field of information retrieval during his time his group at cornell developed the smart information retrieval system which he initiated when he was at harvard

- Step 4: stopword removal
- Stopwords: words that we choose to ignore because we expect them to not be useful in distinguishing between relevant and non-relevant documents for any query

Free-text Indexing

text-processing

gerard salton 8 march 1978 nuremberg 28 august 1995 know gerry salton professor computer science cornell university salton perhaps leading computer scientist working field information retrieval time group cornell developed smart information retrieval system initiated harvard

- Step 4: stopword removal
- Stopwords: words that we choose to ignore because we expect them to not be useful in distinguishing between relevant and non-relevant documents for any query

Free-text Indexing

text-processing

gerard salton 8 march 1978 nuremberg 28 august 1995 gerry salton professor computer science cornell university salton leading computer scientist working field information retrieval during time group cornell developed smart information retrieval system initiated harvard

- Step 5: do this to every document in the collection and create an index using the union of all remaining terms

Document Representation

controlled vocabulary vs. free-text indexing

	Cost of assigning index terms	Ambiguity of index terms	Detail of representation
Controlled Vocabularies	High/Low?	Ambiguous/ Unambiguous?	Can represent arbitrary level of detail?
Free-text Indexing	High/Low?	Ambiguous/ Unambiguous?	Can represent arbitrary level of detail?

Document Representation

controlled vocabulary vs. free-text indexing

	Cost of assigning index terms	Ambiguity of index terms	Detail of representation
Controlled Vocabularies	High	Not ambiguous	Can't represent arbitrary detail
Free-text Indexing	Low	Can be ambiguous	Any level of detail

- Both are effective and used often
- We will focus on free-text indexing
  - cheap and easy
  - most search engines use it (even those that adopt a controlled vocabulary)

Morphological Analysis

Morphology

- the study and description of word formation (as inflection, derivation, and compounding) in language

Merriam-Webster Dictionary

Morphology

- Inflectional morphology: changes to a word that encode its grammatical usage (e.g., tense, number, person)
  - say vs. said, cat vs. cats, see vs. sees
- Derivational morphology: changes to a word to make a new word with related meaning
  - organize, organization, organizational
- Compounding: combining words to form new ones
  - shipwreck, outbound, beefsteak
  - more common in other languages (e.g., german)
  - lebensversicherungsgesellschaftangestellter

Morphological Analysis
in information retrieval

- Basic question: words occur in different forms. Do we want to treat different forms as different index terms?

- Conflation: treating different (inflectional and derivational) variants as the same index term

Morphological Analysis
in information retrieval

- What are we trying to achieve by conflating morphological variants?
- Goal: help the system ignore unimportant variations of language usage

Morphological Analysis
in information retrieval

- The query "computer repairs" will match all combinations of:

computer	and	repair
computers		repairs
computing		repaired
computation		repairing
computational		repairable
::		::

Morphological Analysis
in information retrieval

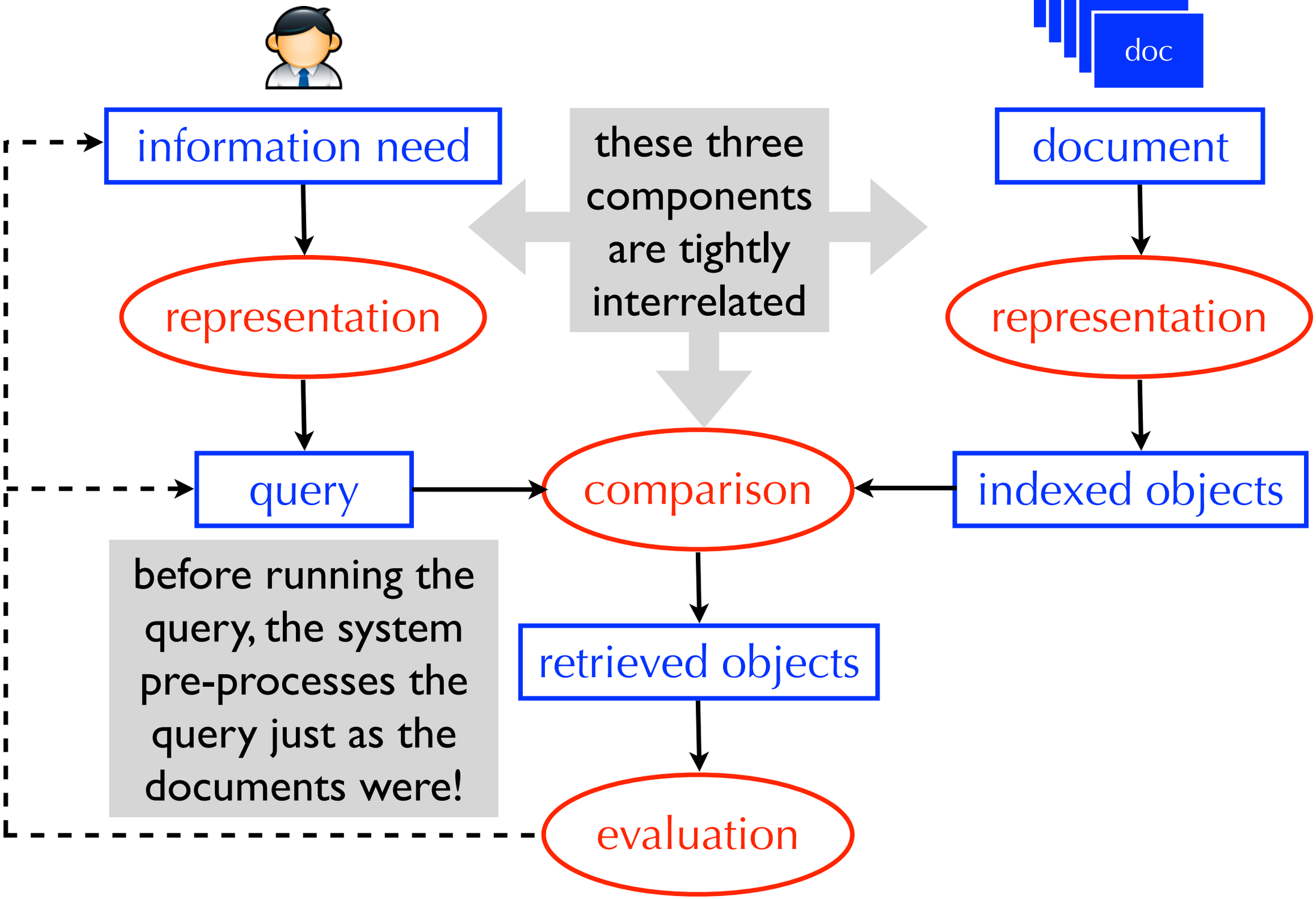
- In English, conflating morphological variants is usually done using a stemmer
- Stemming: automatic suffix-stripping
- English word variations occur at the end of a word
- root/stem + suffix
  - repair + s/ed/ing/able
- A stemmer conflates different variations by reducing them to a common root/stem

Morphological Analysis
in information retrieval

- In some cases, whatever is left after suffix-stripping is not even a word (e.g., comput)
- Is this a problem?

computer	repair
computers	repairs
computing	repaired
computation	repairing
computational	repairable
::	::

Morphological Analysis
in information retrieval



Morphological Analysis

the porter stemmer (porter '80)

- A long list of rules that are applied in sequence
  - apply the rule that removes the longest suffix
  - check to see that the stem is likely to be a root (replac+ement vs. c+ement)
- Fast, effective, and, therefore, very popular

Martin Porter's Home Page

No doubt you came here out of idle curiosity from the Porter Stemming Algorithm page. Before you hastily return, you are welcome to look at the following.

This (jerkily) spinning can is the work of Philip Holmes Esquire, ingenious graphic designer and inventor of visual puns. I could never have thought up anything so clever. (Apologies to the Dr Pepper people!)

Morphological Analysis
the porter stemmer (porter '80)

- Original Text

gerard salton 8 march 1978 in nuremberg 28 august 1995 also know as gerry salton was professor of computer science at cornell university salton was perhaps the leading computer scientist working in the field of information retrieval during his time his group at cornell developed the smart information retrieval system which he initiated when he was at harvard

- Stemmed Text

gerard salton 8 march 1978 in nuremberg 28 august 1995 also know as gerri salton wa professor of comput scienc at cornel univers salton wa perhap the lead comput scientist work in the field of inform retriev dure hi time hi group at cornel develop the smart inform retriev system which he initi when he wa at harvard

Morphological Analysis
the porter stemmer (porter '80)

- false positives: two words conflated to the same root when they shouldn't have been

organization/organ
generalization/generic
numerical/numerous
policy/police
university/universe
addition/additive
negligible/negligent
execute/executive
past/paste
ignore/ignorant
special/specialized
head/heading

Morphological Analysis

the porter stemmer (porter '80)

- false negatives: two words not conflated to the same root word when they should have been

european/europe
cylinder/cylindrical
matrices/matrix
urgency/urgent
create/creation
analysis/analyses
useful/usefully
noise/noisy
decompose/decomposition
sparse/sparsity
resolve/resolution
triangle/triangular

AOL Query-log Examples

stemmed queries

russian translat
- russian translations
- russian translator
- russian translation
- russian translate

secret
- secret
- secretions
- secrets
- secretion

stock for sale
- stockings for sale
- stocking for sale
- stocks for sale

smokei mountain nation park
- smokey mountains national park
- smokey mountain national park
- smokey mountains national parks

cat fenc
- cat fencing
- cat fences
- cat fence

strawberri plant
- strawberry planting
- strawberry plants
- strawberries planting

Morphological Analysis
evaluation results

- Stemming
  - English: 0-5% improvements
  - Finnish: 30% improvement
  - Spanish: 10% improvement
- Compound Splitting
  - German: 15% improvements
  - Swedish: 25% improvement

(Hollink *et al.*, 2004)

Morphology Across Languages

European Parliament Corpus

- Number of unique terms (translations of the same text):
  - English: 150,725
  - Spanish: 213,486
  - Portuguese: 219,121
  - Danish: 367,282
  - Finnish: 709,049
  - German: 401,929

Outline

Information Retrieval

Search Engine Components

Document Representation

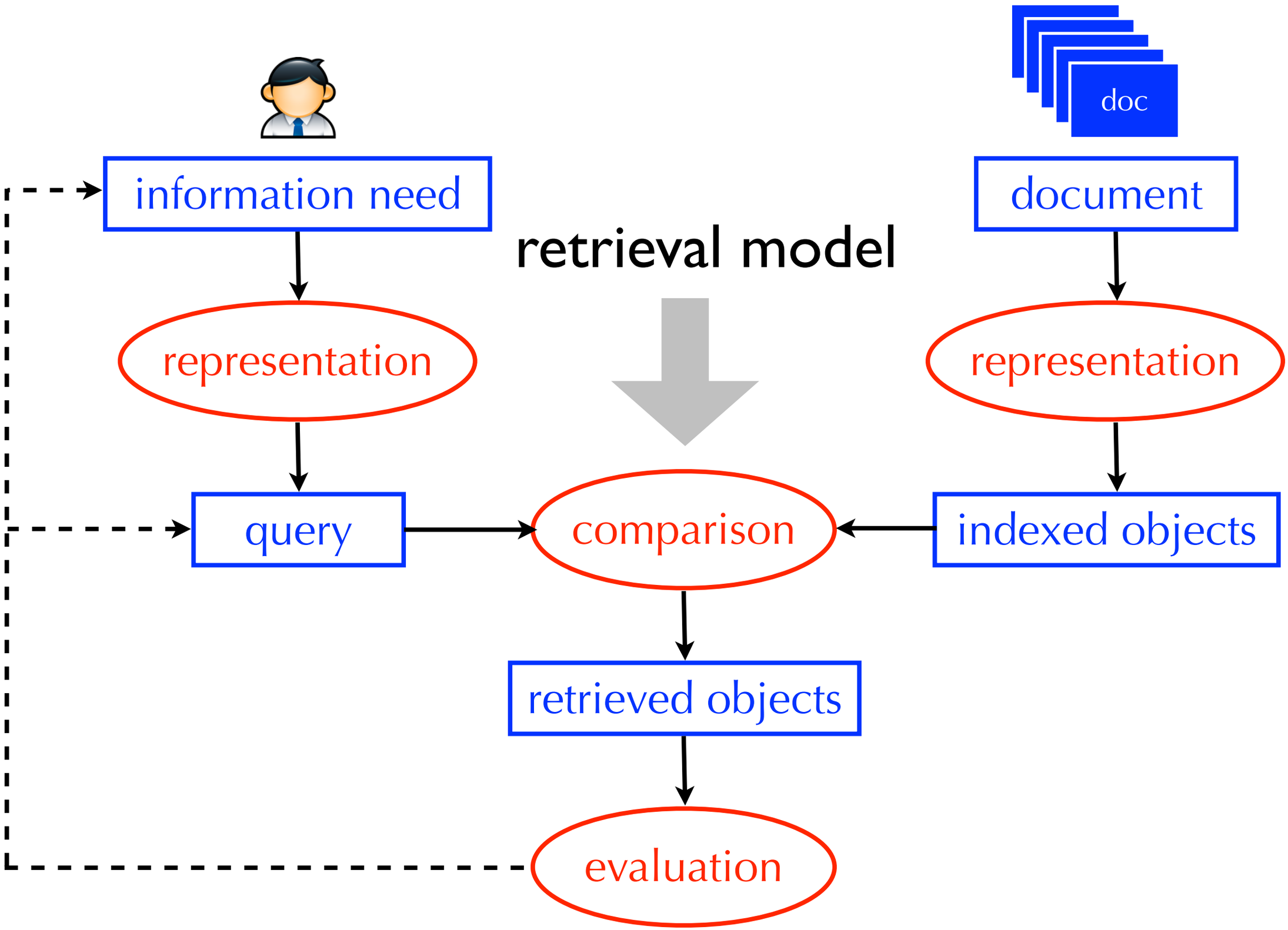
Retrieval Models

Evaluation

Federated Search and Cross-lingual IR

Open-source Toolkits

Information Retrieval Process



Ranked Boolean

University	*North*	*Carolina*	*UNC*
df=6	*df=4*	*df=3*	*df=5*
1, 4	1, 4	1, 4	1, 4
10, 1	10, 5	10, 5	10, 1
15, 2	16, 1	16, 1	16, 4
16, 1	68, 1		33, 2
33, 5			56, 10
67, 7			

- *docid = document identifier*
- *tf = term frequency (# of times the term appears in the document)*

Ranked Boolean

- At each step, keep a list of documents that match the query and their scores (a.k.a. a "priority queue")
- Score computation:
  - A AND B: adjust the document score based on the minimum frequency/score associated with expression A and expression B
  - A OR B: adjust the document score based on the sum of frequencies/scores associated with expression A and expression B

Ranked Boolean

- Query: (*University* AND *North* AND *Carolina*) OR *UNC*

University	*North*	*Carolina*	*UNC*
df=6	*df=4*	*df=3*	*df=5*
1, 4	1, 4	1, 4	1, 4
10, 1	10, 5	10, 5	10, 1
15, 2	16, 1	16, 1	16, 4
16, 1	68, 1		33, 2
33, 5			56, 10
68, 7			

- AND → min
- OR → sum

Ranked Boolean

- Query: (*University* AND *North* AND *Carolina*) OR *UNC*

University	*North*	*Carolina*	*Result_1*
df=6	*df=4*	*df=3*	*count=??*
1, 4	1, 4	1, 4	
10, 1	10, 5	10, 5	
15, 2	16, 1	16, 1	
16, 1	68, 1		
33, 5			
68, 7			

- AND → min
- OR → sum

Ranked Boolean

- Query: (*University* AND *North* AND *Carolina*) OR *UNC*

University	*North*	*Carolina*	*Result_1*
df=6	*df=4*	*df=3*	*count=3*
1, 4	1, 4	1, 4	1, 4
10, 1	10, 5	10, 5	10, 1
15, 2	16, 1	16, 1	16, 1
16, 1	68, 1		
33, 5			
68, 7			

- AND → min
- OR → sum

Ranked Boolean

- Query: (*University* AND *North* AND *Carolina*) OR *UNC*

Result_1	*UNC*	*Query*
count=3	*df=5*	*count=??*
1, 4	1, 4	
10, 1	10, 1	
16, 1	16, 4	
	33, 2	
	56, 10	

- AND → min
- OR → sum

Ranked Boolean

- Query: (*University* AND *North* AND *Carolina*) OR *UNC*

Result_1	*UNC*	*Query*
count=3	*df=5*	*count=5*
1, 4	1, 4	1, 8
10, 1	10, 1	10, 2
16, 1	16, 4	16, 5
	33, 2	33, 2
	56, 10	56, 10

- AND → min
- OR → sum

Ranked Boolean

- Query: (*University* AND *North* AND *Carolina*) OR *UNC*

University	*North*	*Carolina*	*UNC*	*Query*
df=6	*df=4*	*df=3*	*df=5*	*count=5*
1, 4	1, 4	1, 4	1, 4	1, 8
10, 1	10, 5	10, 5	10, 1	10, 2
15, 2	16, 1	16, 1	16, 4	16, 5
16, 1	68, 1		33, 2	33, 2
33, 5			56, 10	56, 10
68, 7				

- Conceptually, what do these document scores indicate?

Ranked Boolean

- Query: (*University* AND *North* AND *Carolina*) OR *UNC*

University	*North*	*Carolina*	*UNC*	*Query*
df=6	*df=4*	*df=3*	*df=5*	*count=5*
1, 4	1, 4	1, 4	1, 4	1, 8
10, 1	10, 5	10, 5	10, 1	10, 2
15, 2	16, 1	16, 1	16, 4	16, 5
16, 1	68, 1		33, 2	33, 2
33, 5			56, 10	56, 10
68, 7				

- The *scores* correspond to the number of ways in which the document redundantly satisfies the query

Ranked Boolean

- Advantages:
  - same as unranked boolean: efficient, predictable, easy to understand, works well when the user knows what to look for
  - the user may be able to find relevant documents quicker and may not need to examine the entire result set
- Disadvantages:
  - same as unranked boolean: works well when the user knows what to look for

Retrieval Model 2: Ranked Boolean

- Query: (*University* AND *North* AND *Carolina*) OR *UNC*

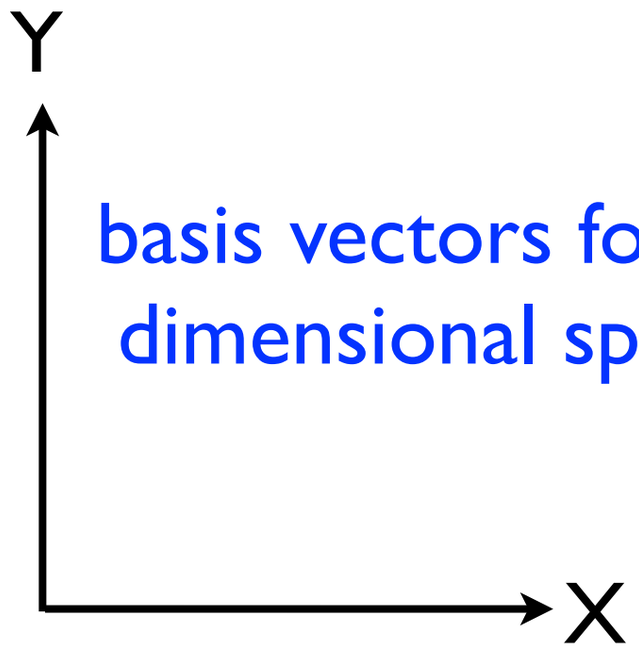
University	*North*	*Carolina*	*UNC*	*Query*
df=6	*df=4*	*df=3*	*df=5*	*count=5*
1, 4	1, 4	1, 4	1, 4	1, 8
10, 1	10, 5	10, 5	10, 1	10, 2
15, 2	16, 1	16, 1	16, 4	16, 5
16, 1	68, 1		33, 2	33, 2
33, 5			56, 10	56, 10
68, 7				

- Are all these terms equally important?
- We will explore more effective term-weighting schemes!

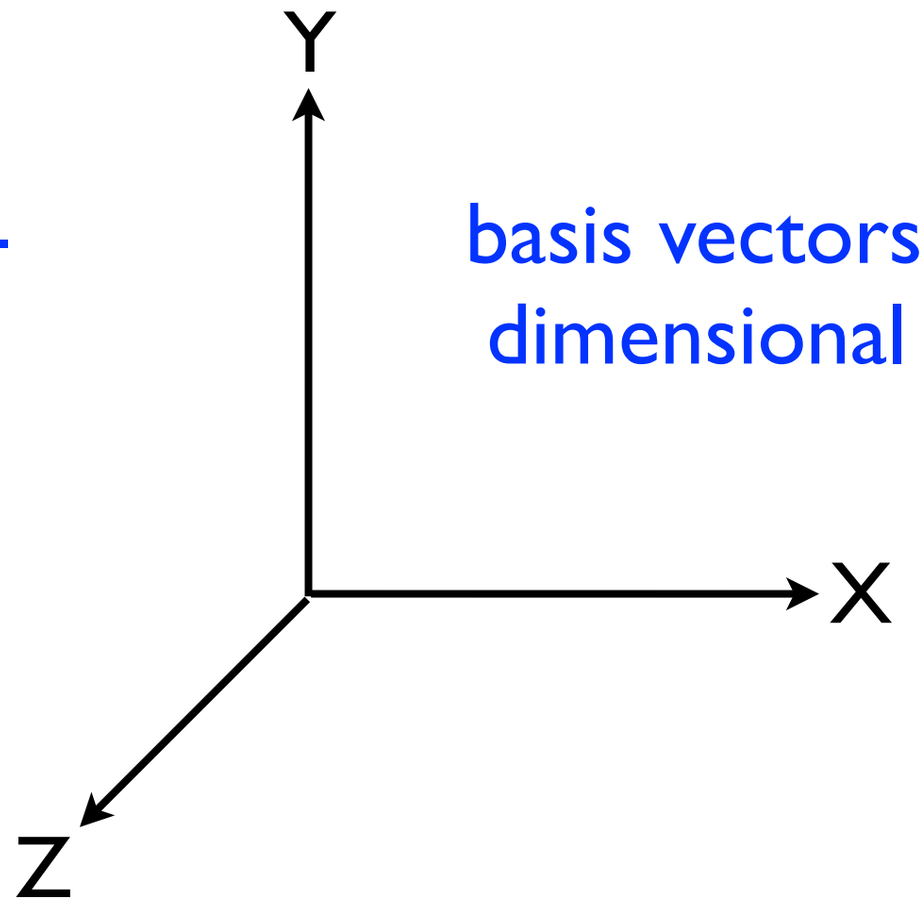
Vector Space Model

What is a Vector Space?

- Formally, a vector space is defined by a set of linearly independent basis vectors
- The basis vectors correspond to the dimensions or directions of the vector space



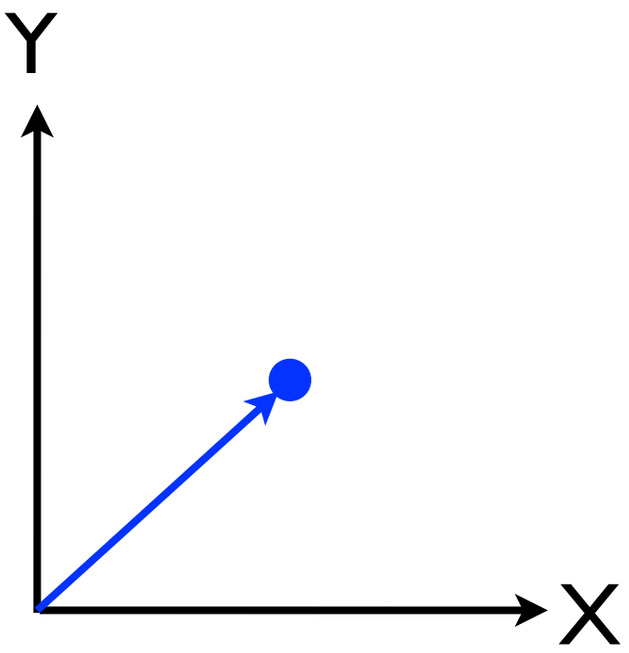
basis vectors for 2-dimensional space

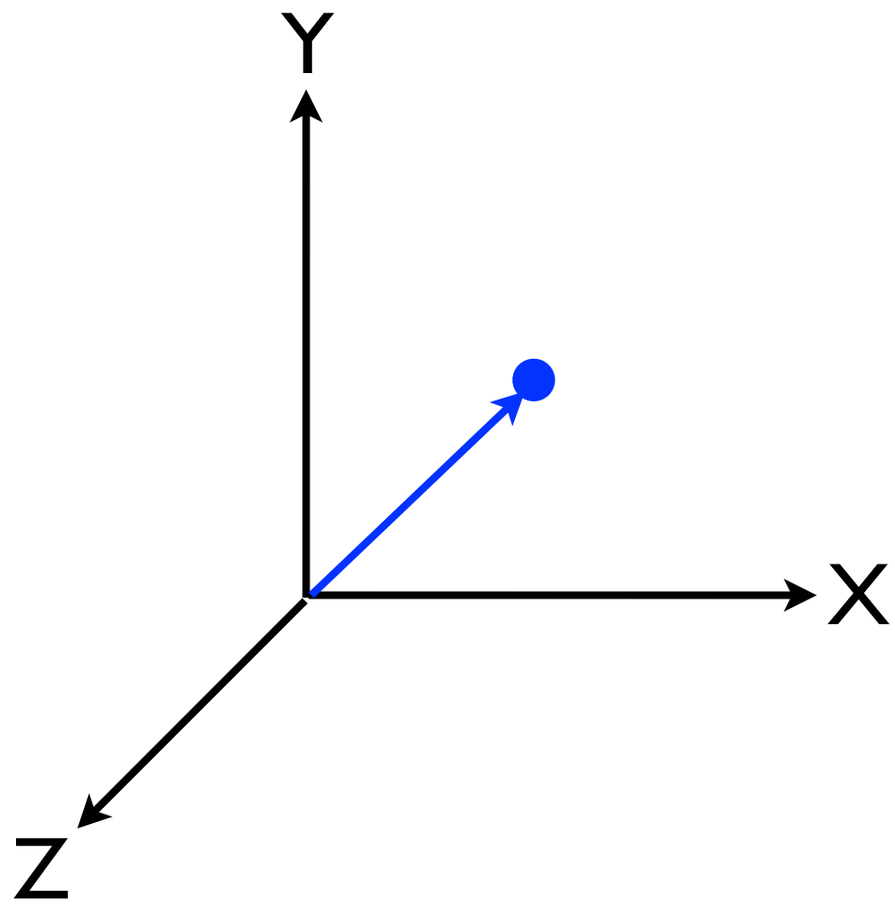


basis vectors for 3-dimensional space

What is a Vector?

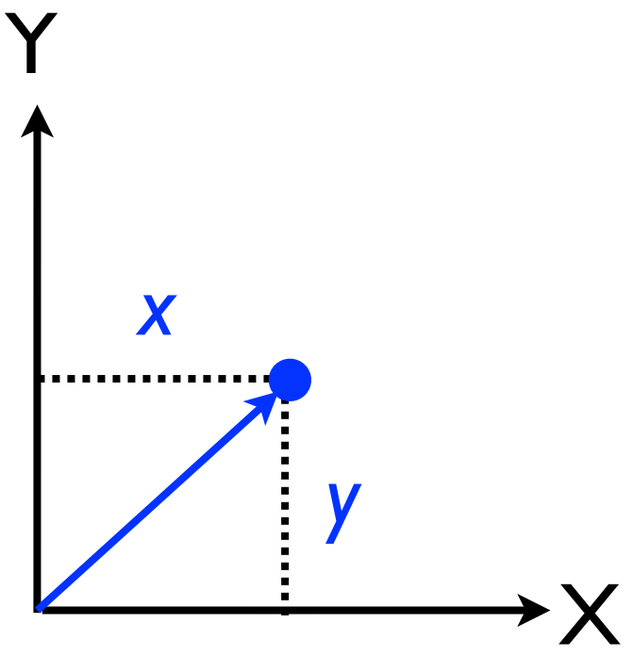
- A vector is a point in a vector space and has length (from the origin to the point) and direction

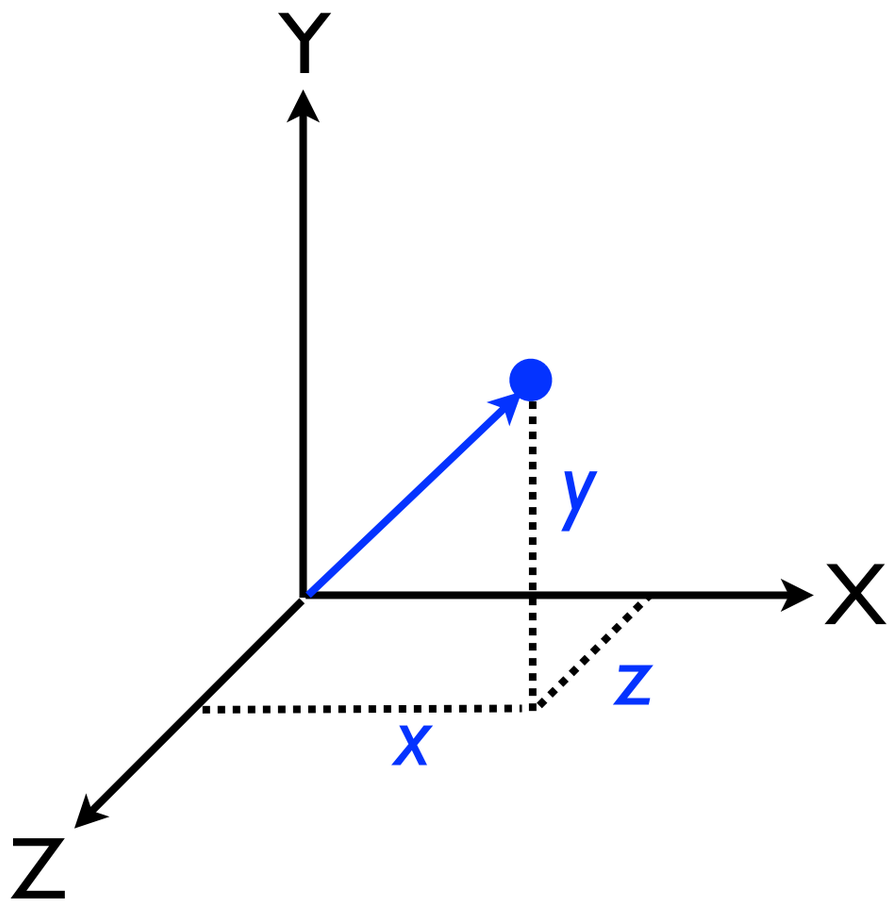




What is a Vector?

- A 2-dimensional vector can be written as [*x*,*y*]
- A 3-dimensional vector can be written as [*x*,*y*,*z*]





Binary Text Representation

	a	*aardvark*	*abacus*	*abba*	*able*	...	*zoom*
doc_1	1	0	0	0	0	...	1
doc_2	0	0	0	0	1	...	1
::	::	::	::	::	::	...	0
doc_m	0	0	1	1	0	...	0

- 1 = the word appears in the document
- 0 = the word does not appear in the document
- Does not represent word frequency, word location, or word order information

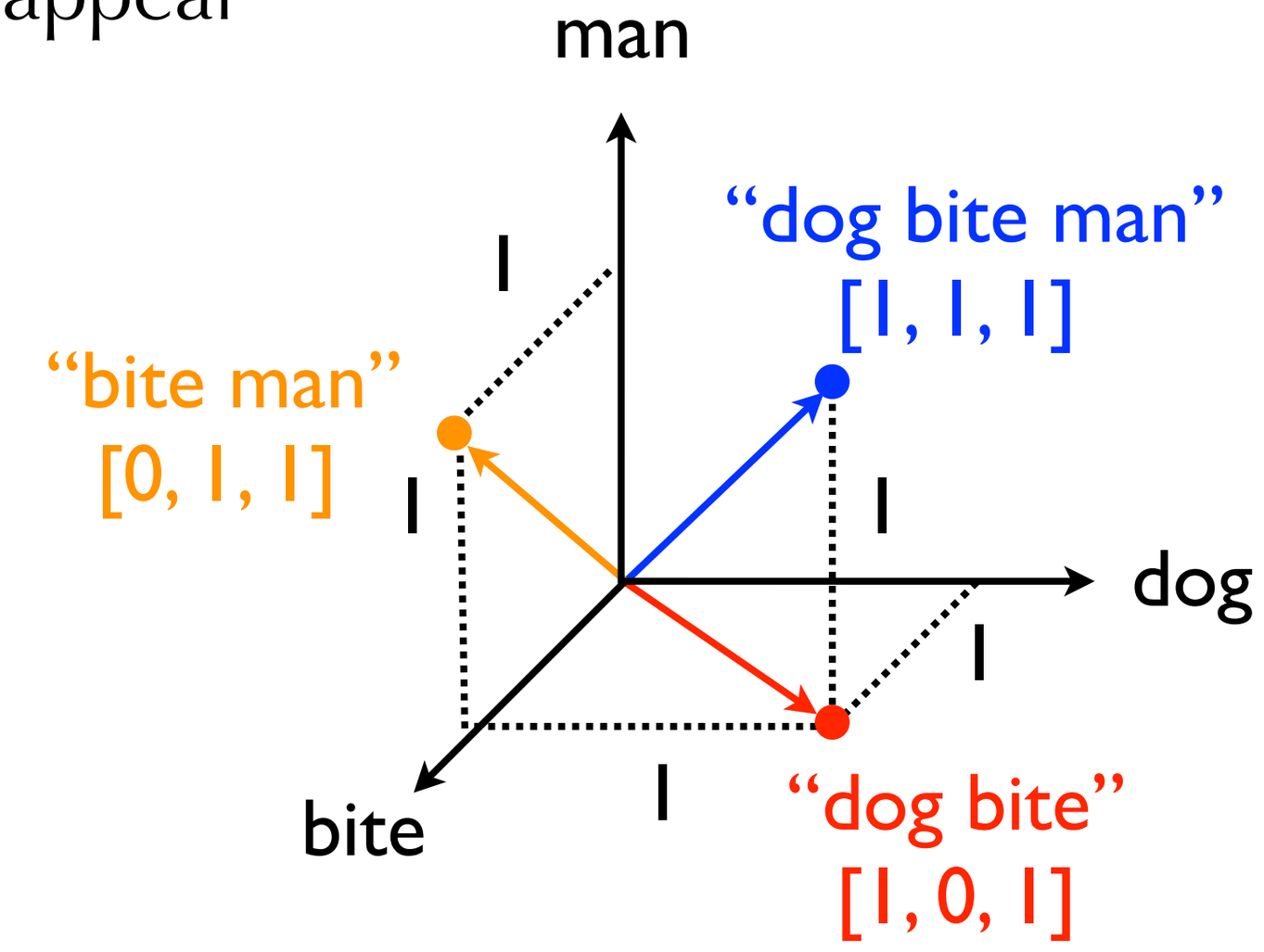
Vector Space Representation

- Let V denote the size of the indexed vocabulary
  - V = the number of unique terms,
  - V = the number of unique terms excluding stopwords,
  - V = the number of unique stems, etc...
- Any arbitrary span of text (i.e., a document, or a query) can be represented as a vector in V-dimensional space
- For simplicity, let's assume three index terms: dog, bite, man (i.e., V=3)
- Why? Because it's easy to visualize 3-D space

Vector Space Representation
with binary weights

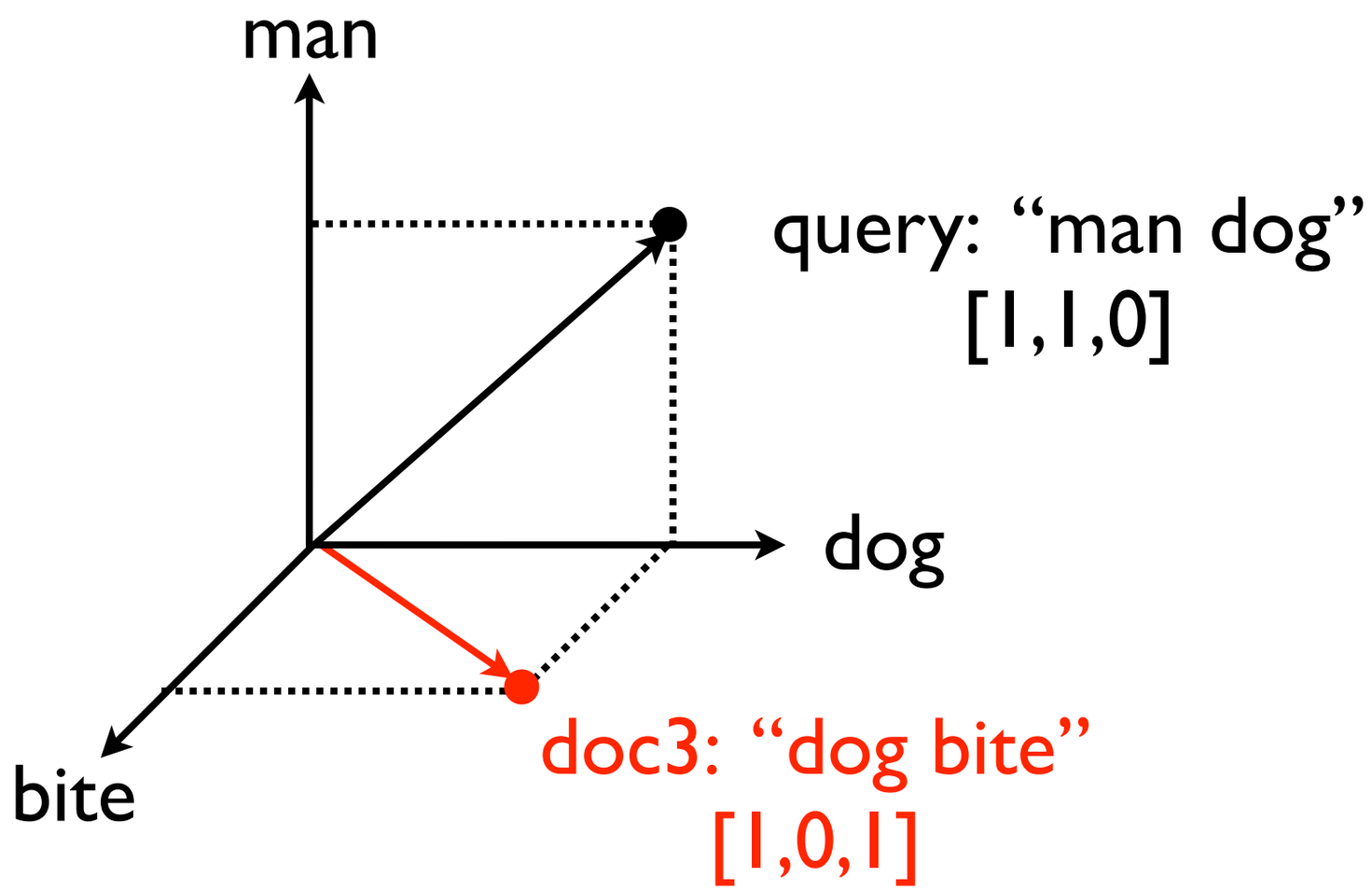
- 1 = the term appears at least once
- 0 = the term does not appear

	dog	*man*	*bite*
doc_1	1	1	1
doc_2	1	0	1
doc_3	0	1	1



Vector Space Representation

- A query is a vector in V-dimensional space, where V is the number of terms in the vocabulary



Vector Space Similarity

- The vector space model ranks documents based on the vector-space similarity between the query vector and the document vector
- There are many ways to compute the similarity between two vectors
- One way is to compute the inner product

$$\sum_{i=1}^{V} x_i \times y_i$$

The Inner Product

- Multiply corresponding components and then sum those products

$$\sum_{i=1}^{V} x_i \times y_i$$

	x_i	y_i	$x_i \times y_i$
a	1	1	1
aardvark	0	1	0
abacus	1	1	1
abba	1	0	0
able	0	1	0
::	::	::	::
zoom	0	0	0
		inner product =>	2

The Inner Product

- When using 0's and 1's, this is just the number of unique terms in common between the query and the document

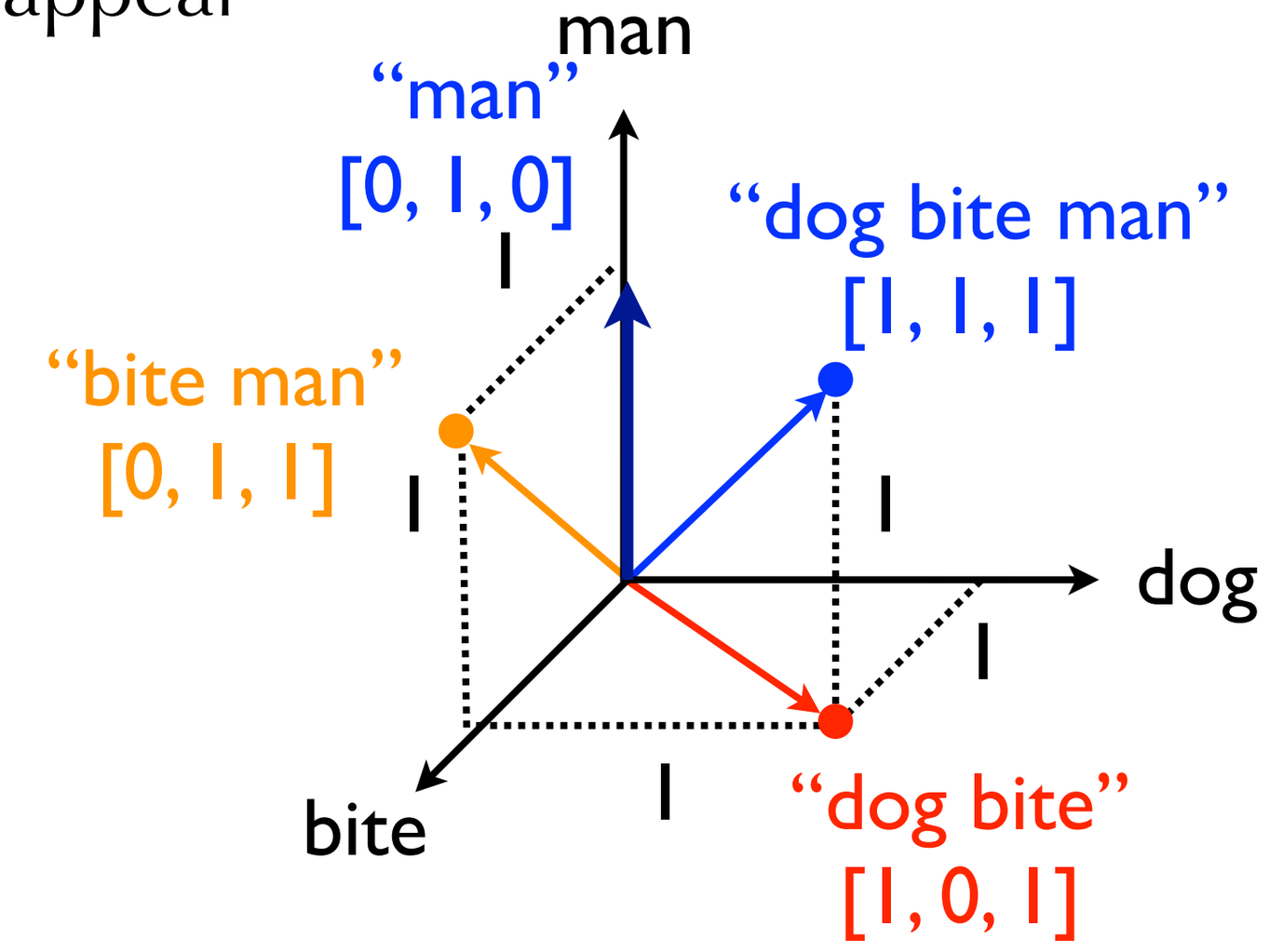
$$\sum_{i=1}^{V} x_i \times y_i$$

	x_i	y_i	$x_i \times y_i$
a	1	1	1
aardvark	0	1	0
abacus	1	1	1
abba	1	0	0
able	0	1	0
::	::	::	::
zoom	0	0	0
		inner product =>	2

The Inner Product

- 1 = the term appears at least once
- 0 = the term does not appear

	dog	*man*	*bite*
doc_1	1	1	1
doc_2	1	0	1
doc_3	0	1	1
doc_4	0	1	0



The Inner Product

- Multiply corresponding components and then sum those products
- Using a binary representation, the inner product corresponds to the number of terms appearing (at least once) in both spans of text
- Scoring documents based on their inner-product with the query has one major issue. Any ideas?

The Inner Product

- What is more relevant to a query?
  - A 50-word document which contains 3 of the query-terms?
  - A 100-word document which contains 3 of the query-terms?
- All things being equal, longer documents are more likely to have the query-terms
- The inner-product doesn't account for the fact that documents have widely varying lengths
- So, the inner-product favors long documents

The Cosine Similarity

- The numerator is the inner product
- The denominator is the product of the two vector-lengths
- The cosine of the angle between the two vectors
- Ranges from 0 to 1
- 0 if the angle is 90 degrees
- 1 if the angle is 0 degrees

$$\frac{\sum_{i=1}^{V} x_i \times y_i}{\sqrt{\sum_{i=1}^{V} x_i^2} \times \sqrt{\sum_{i=1}^{V} y_i^2}}$$

length of vector x

length of vector y

Vector Space Model

cosine similarity example (binary weights)

$$\frac{\sum_{i=1}^{V} x_i \times y_i}{\sqrt{\sum_{i=1}^{V} x_i^2} \times \sqrt{\sum_{i=1}^{V} y_i^2}}$$

cosine([1,0,1] , [1,1,0]) =

$$\frac{(1 \times 1) + (0 \times 1) + (1 \times 0)}{\sqrt{1^2 + 0^2 + 1^2} \times \sqrt{1^2 + 1^2 + 0^2}} = 0.5$$

Vector Space Representation

	a	*aardvark*	*abacus*	*abba*	*able*	...	*zoom*
doc_1	1	0	0	0	0	...	1
doc_2	0	0	0	0	1	...	1
::	::	::	::	::	::	...	0
doc_m	0	0	1	1	0	...	0

	a	*aardvark*	*abacus*	*abba*	*able*	...	*zoom*
query	0	1	0	0	1	...	1

- So far, we've assumed binary vectors
- 0's and 1's indicate whether the term occurs (at least once) in the document/query
- Let's explore a more sophisticated representation

Term-Weighting

what are the most important terms?

- Movie: Rocky (1976)
- Plot:

Rocky Balboa is a struggling boxer trying to make the big time. Working in a meat factory in Philadelphia for a pittance, he also earns extra cash as a debt collector. When heavyweight champion Apollo Creed visits Philadelphia, his managers want to set up an exhibition match between Creed and a struggling boxer, touting the fight as a chance for a "nobody" to become a "somebody". The match is supposed to be easily won by Creed, but someone forgot to tell Rocky, who sees this as his only shot at the big time. Rocky Balboa is a small-time boxer who lives in an apartment in Philadelphia, Pennsylvania, and his career has so far not gotten off the canvas. Rocky earns a living by collecting debts for a loan shark named Gazzo, but Gazzo doesn't think Rocky has the viciousness it takes to beat up deadbeats. Rocky still boxes every once in a while to keep his boxing skills sharp, and his ex-trainer, Mickey, believes he could've made it to the top if he was willing to work for it. Rocky, goes to a pet store that sells pet supplies, and this is where he meets a young woman named Adrian, who is extremely shy, with no ability to talk to men. Rocky befriends her. Adrain later surprised Rocky with a dog from the pet shop that Rocky had befriended. Adrian's brother Paulie, who works for a meat packing company, is thrilled that someone has become interested in Adrian, and Adrian spends Thanksgiving with Rocky. Later, they go to Rocky's apartment, where Adrian explains that she has never been in a man's apartment before. Rocky sets her mind at ease, and they become lovers. Current world heavyweight boxing champion Apollo Creed comes up with the idea of giving an unknown a shot at the title. Apollo checks out the Philadelphia boxing scene, and chooses Rocky. Fight promoter Jergens gets things in gear, and Rocky starts training with Mickey. After a lot of training, Rocky is ready for the match, and he wants to prove that he can go the distance with Apollo. The 'Italian Stallion', Rocky Balboa, is an aspiring boxer in downtown Philadelphia. His one chance to make a better life for himself is through his boxing and Adrian, a girl who works in the local pet store. Through a publicity stunt, Rocky is set up to fight Apollo Creed, the current heavyweight champion who is already set to win. But Rocky really needs to triumph, against all the odds...

Term-Frequency

how important is a term?

rank	term	freq.	rank	term	freq.
1	a	22	16	creed	5
2	rocky	19	17	philadelphia	5
3	to	18	18	has	4
4	the	17	19	pet	4
5	is	11	20	boxing	4
6	and	10	21	up	4
7	in	10	22	an	4
8	for	7	23	boxer	4
9	his	7	24	s	3
10	he	6	25	balboa	3
11	adrian	6	26	it	3
12	with	6	27	heavyweigh	3
13	who	6	28	champion	3
14	that	5	29	fight	3
15	apollo	5	30	become	3

Term-Frequency

how important is a term?

rank	term	freq.	rank	term	freq.
1	a	22	16	creed	5
2	rocky	19	17	philadelphia	5
3	to	18	18	has	4
4	the	17	19	pet	4
5	is	11	20	boxing	4
6	and	10	21	up	4
7	in	10	22	an	4
8	for	7	23	boxer	4
9	his	7	24	s	3
10	he	6	25	balboa	3
11	adrian	6	26	it	3
12	with	6	27	heavyweigh	3
13	who	6	28	champion	3
14	that	5	29	fight	3
15	apollo	5	30	become	3

Inverse Document Frequency (IDF)

how important is a term?

$$idf_t = \log(\frac{N}{df_t})$$

- N = number of documents in the collection
- df_t = number of documents in which term t appears

Inverse Document Frequency (IDF)

how important is a term?

rank	term	idf	rank	term	idf
1	doesn	11.66	16	creed	6.84
2	adrain	10.96	17	paulie	6.82
3	viciousness	9.95	18	packing	6.81
4	deadbeats	9.86	19	boxes	6.75
5	touting	9.64	20	forgot	6.72
6	jergens	9.35	21	ease	6.53
7	gazzo	9.21	22	thanksgivin	6.52
8	pittance	9.05	23	earns	6.51
9	balboa	8.61	24	pennsylvani	6.50
10	heavyweigh	7.18	25	promoter	6.43
11	stallion	7.17	26	befriended	6.38
12	canvas	7.10	27	exhibition	6.31
13	ve	6.96	28	collecting	6.23
14	managers	6.88	29	philadelphia	6.19
15	apollo	6.84	30	gear	6.18

TF.IDF

how important is a term?

$$tf_t \times idf_t$$

greater when the term is frequent in in the document

greater when the term is rare in the collection (does not appear in many documents)

TF.IDF

$$tf_t \times log\left(\frac{N}{df_t}\right)$$

term	tf	N	df	idf	tf.idf
rocky	19	230721	1420	5.09	96.72
philadelphia	5	230721	473	6.19	30.95
boxer	4	230721	900	5.55	22.19
fight	3	230721	8170	3.34	10.02
mickey	2	230721	2621	4.48	8.96
for	7	230721	117137	0.68	4.75

TF.IDF

how important is a term?

rank	term	tf.idf	rank	term	tf.idf
1	rocky	96.72	16	meat	11.76
2	apollo	34.20	17	doesn	11.66
3	creed	34.18	18	adrain	10.96
4	philadelphia	30.95	19	fight	10.02
5	adrian	26.44	20	viciousness	9.95
6	balboa	25.83	21	deadbeats	9.86
7	boxing	22.37	22	touting	9.64
8	boxer	22.19	23	current	9.57
9	heavyweigh	21.54	24	jergens	9.35
10	pet	21.17	25	s	9.29
11	gazzo	18.43	26	struggling	9.21
12	champion	15.08	27	training	9.17
13	match	13.96	28	pittance	9.05
14	earns	13.01	29	become	8.96
15	apartment	11.82	30	mickey	8.96

Queries as TF.IDF Vectors

- Terms tend to appear only once in the query
- TF usually equals 1
- IDF is computed using the collection statistics

$$idf_t = \log(\frac{N}{df_t})$$

- Terms appearing in fewer documents get a higher weight

Queries as TF.IDF Vectors

examples from AOL queries with clicks on IMDB results

term 1	tf.idf	term 2	tf.idf	term 3	tf.idf
central	4.89	casting	6.05	ny	5.99
wizard	6.04	of	0.18	oz	6.14
sam	2.80	jones	3.15	iii	2.26
film	2.31	technical	6.34	advisors	8.74
edie	7.41	sands	5.88	singer	3.88
high	3.09	fidelity	7.66	quotes	8.11
quotes	8.11	about	1.61	brides	6.71
title	4.71	wave	5.68	pics	10.96
saw	4.87	3	2.43	trailers	7.83
the	0.03	rainmaker	9.09	movie	0.00
nancy	5.50	and	0.09	sluggo	9.46
audrey	6.30	rose	4.52	movie	0.00
mark	2.43	sway	7.53	photo	5.14
piece	4.59	of	0.18	cheese	6.38
date	3.93	movie	0.00	cast	0.00

Vector Space Model

cosine similarity example (tf.idf weights)

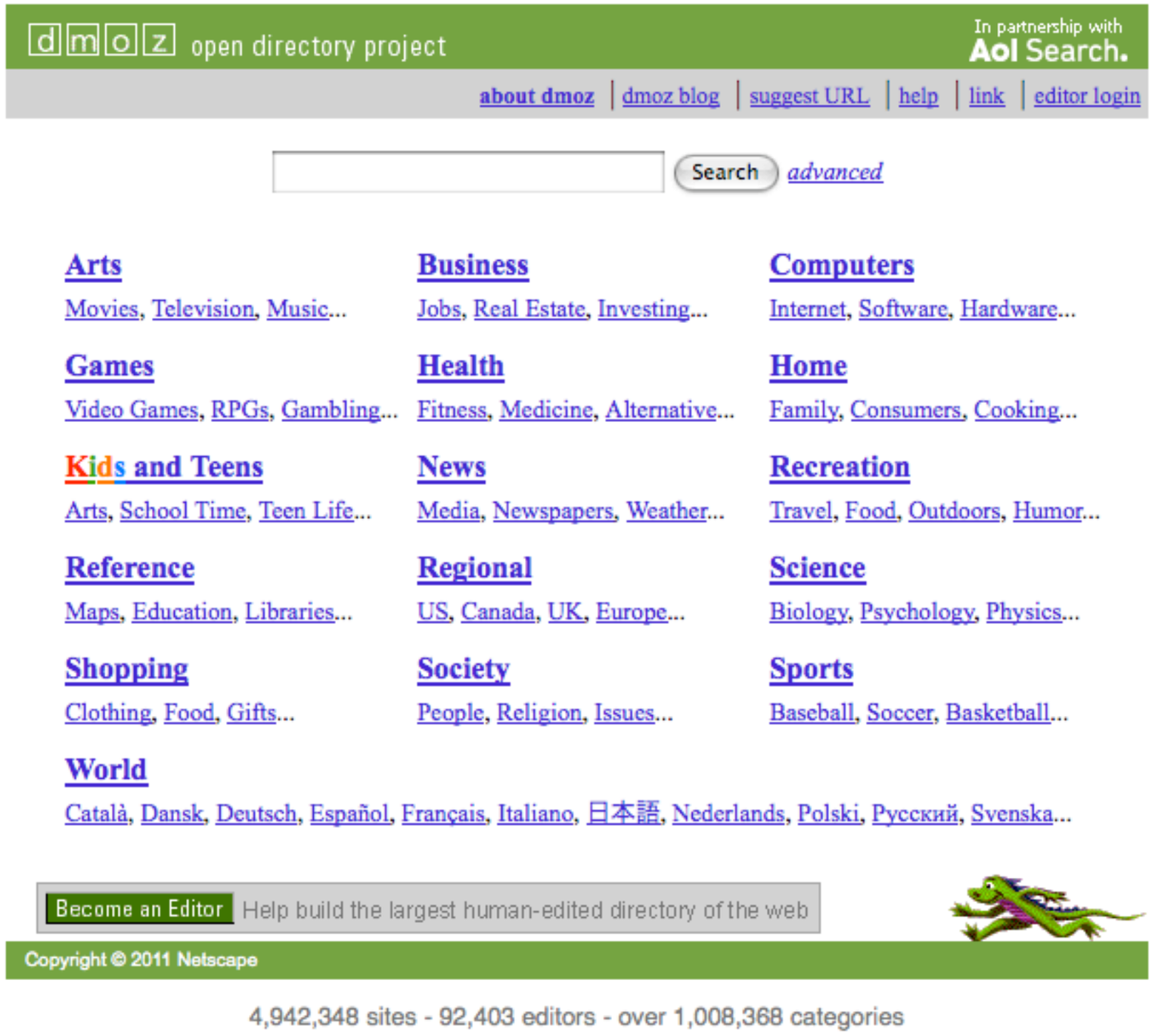
$$\frac{\sum_{i=1}^{V} x_i \times y_i}{\sqrt{\sum_{i=1}^{V} x_i^2} \times \sqrt{\sum_{i=1}^{V} y_i^2}}$$

cosine([2.3, 0.0, 1.5] , [5.4, 2.0, 0.0]) =

$$\frac{(2.3 \times 5.4) + (0.0 \times 2.0) + (1.5 \times 0.0)}{\sqrt{2.3^2 + 0.0^2 + 1.5^2} \times \sqrt{5.4^2 + 2.0^2 + 0.0^2}}$$

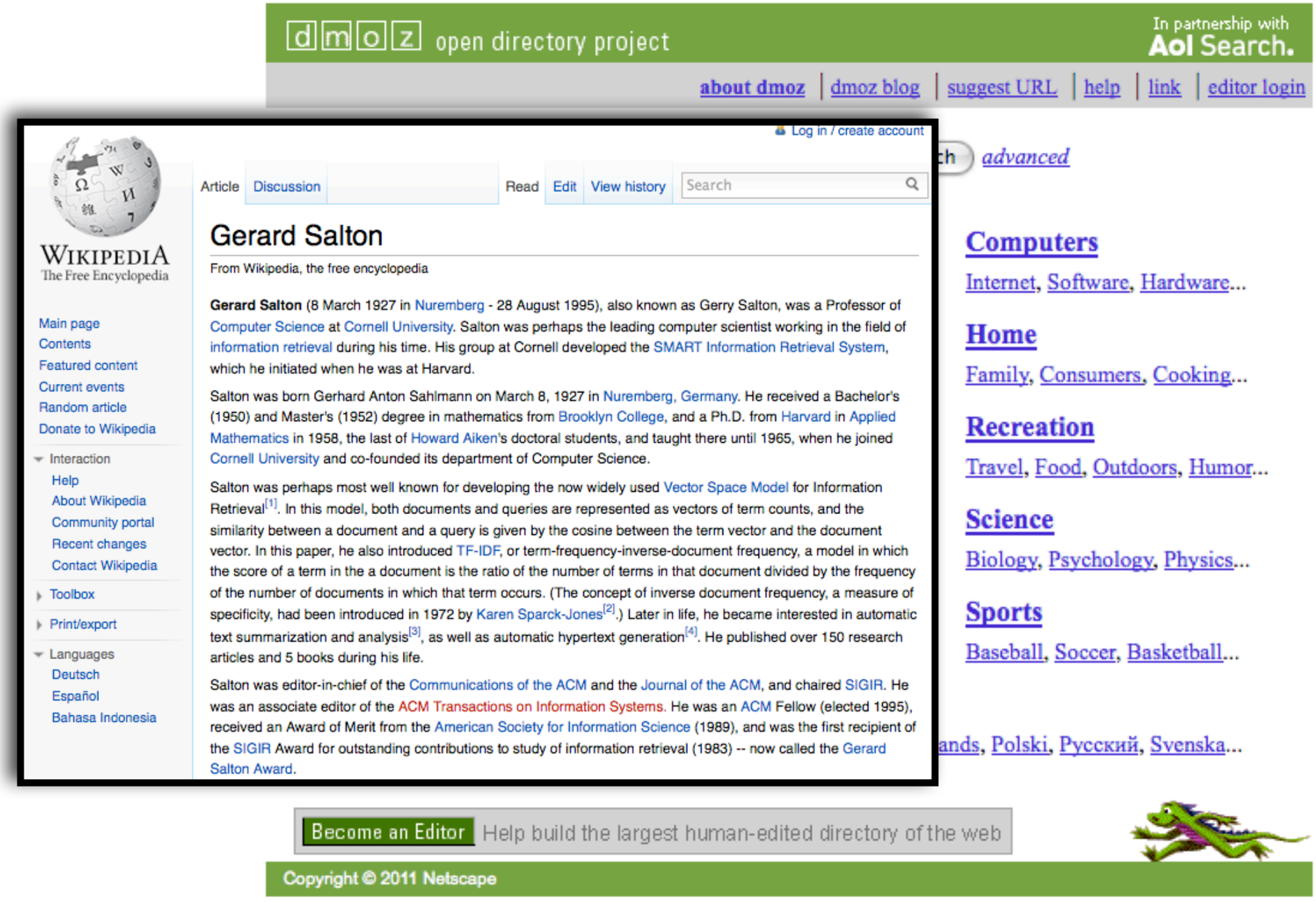
Vector Space Representation

- Topic categorization: automatically assigning a document to a category



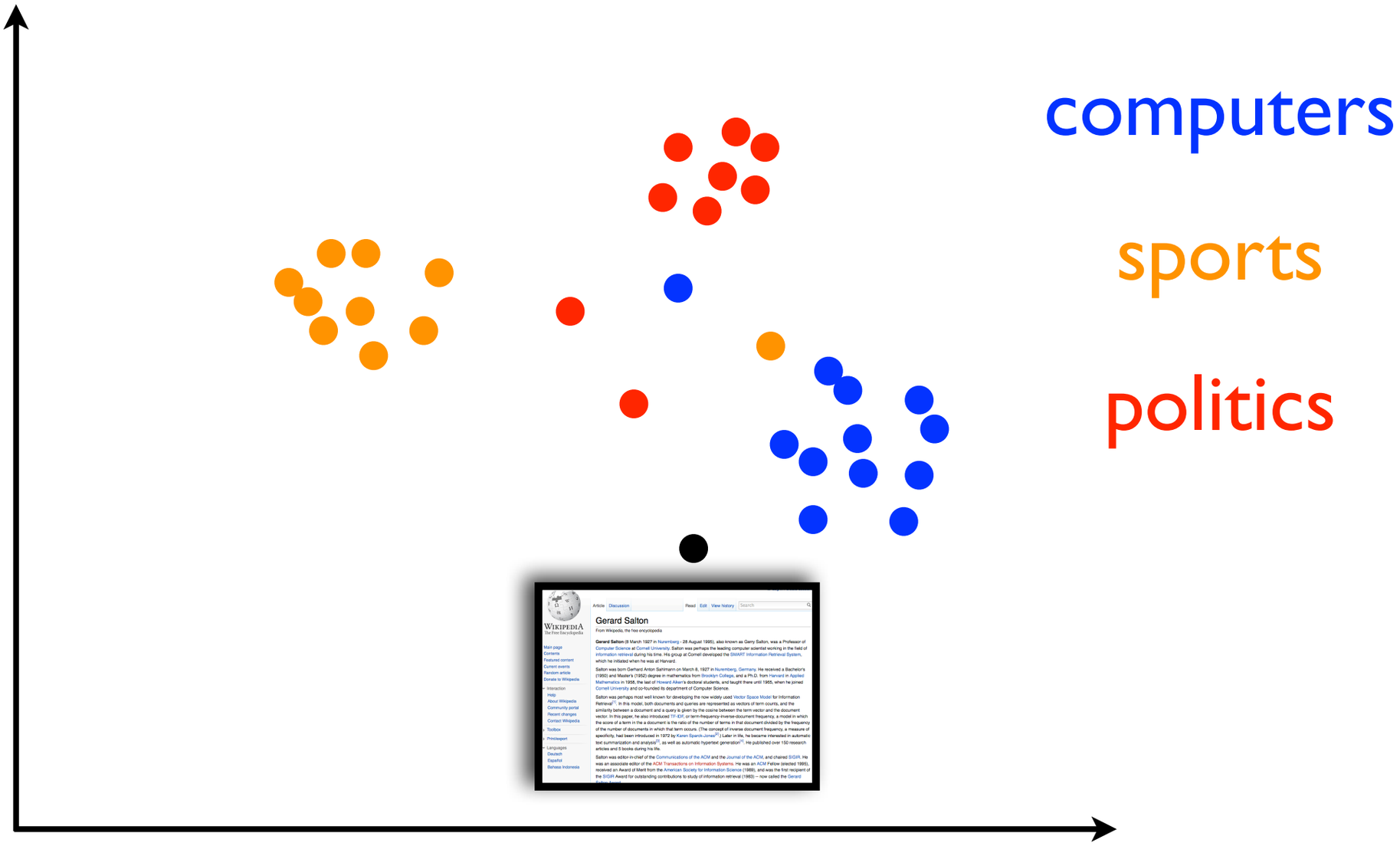
Vector Space Representation

- Find documents (with a known category assignment) that are similar to this document



Vector Space Representation

- Find documents (with a known category assignment) that are similar to this document



Query Likelihood Model

What is a language model?

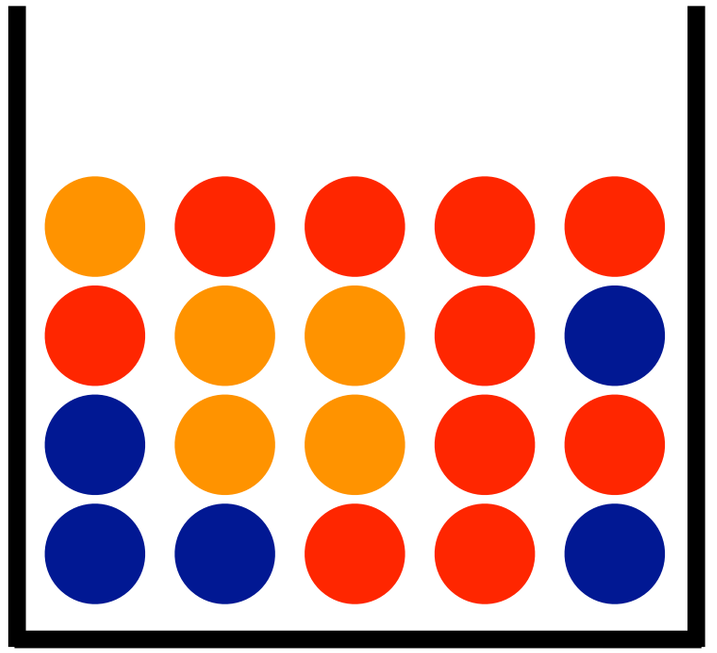
"The goal of a language model is to assign a probability to a sequence of words by means of a probability distribution"

--Wikipedia

What is a probability distribution?

- A probability distribution gives the probability of each possible outcome of a random variable

- P(RED) = probability that you will reach into this bag and pull out a red ball



- P(BLUE) = probability that you will reach into this bag and pull out a blue ball

- P(ORANGE) = probability that you will reach into this bag and pull out an orange ball

What is a probability distribution?

- For it to be a probability distribution, two conditions must be satisfied:
  - the probability assigned to each possible outcome must be between 0 and 1 (inclusive)
  - the sum of probabilities across outcomes must be 1

$$0 \leq P(\text{RED}) \leq 1$$

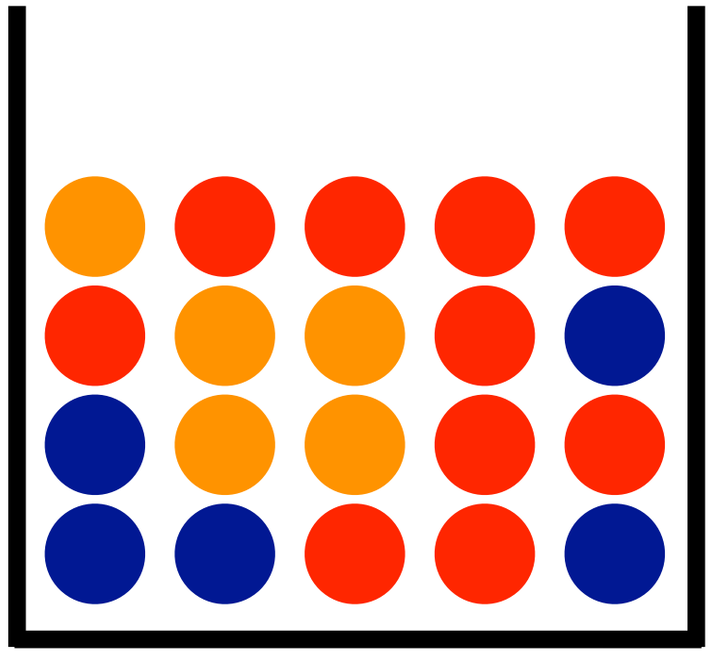
$$0 \leq P(\text{BLUE}) \leq 1$$

$$0 \leq P(\text{ORANGE}) \leq 1$$

$$P(\text{RED}) + P(\text{BLUE}) + P(\text{ORANGE}) = 1$$

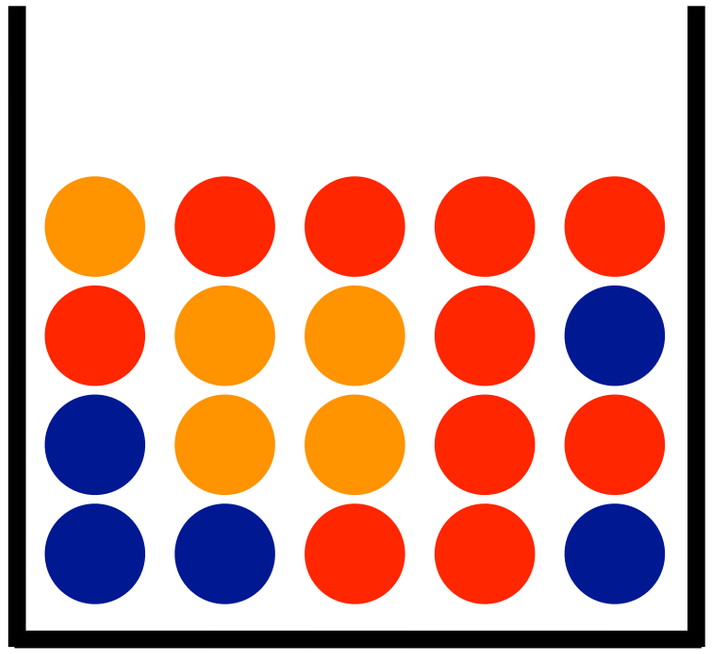
Estimating a Probability Distribution

- Let's estimate these probabilities based on what we know about the contents of the bag
- P(RED) = ?
- P(BLUE) = ?
- P(ORANGE) = ?



Estimating a Probability Distribution

- Let's estimate these probabilities based on what we know about the contents of the bag



- P(RED) = 10/20 = 0.5
- P(BLUE) = 5/20 = 0.25
- P(ORANGE) = 5/20 = 0.25
- P(RED) + P(BLUE) + P(ORANGE) = 1.0

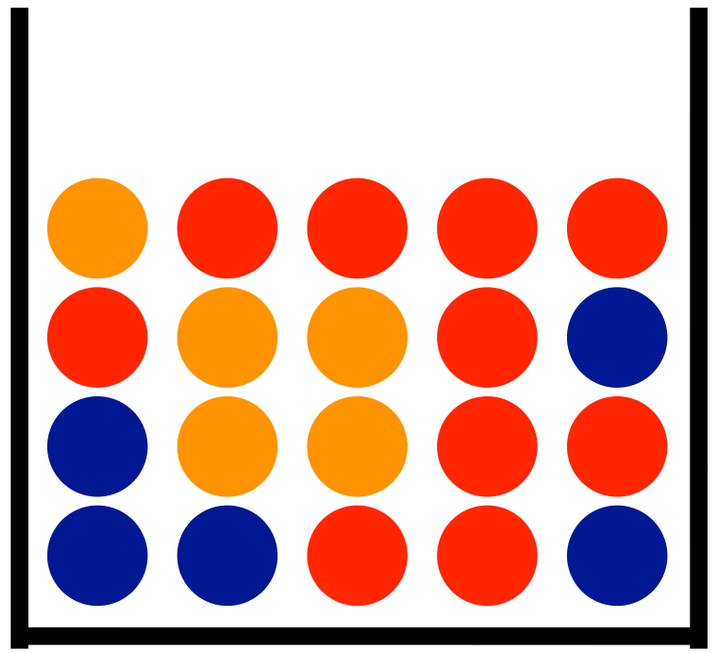
What can we do with a probability distribution?

- We can assign probabilities to different outcomes
- I reach into the bag and pull out an orange ball. What is the probability of that happening?
- I reach into the bag and pull out two balls: one red, one blue. What is the probability of that happening?
- What about three orange balls?

P(RED) = 0.5

P(BLUE) = 0.25

P(ORANGE) = 0.25

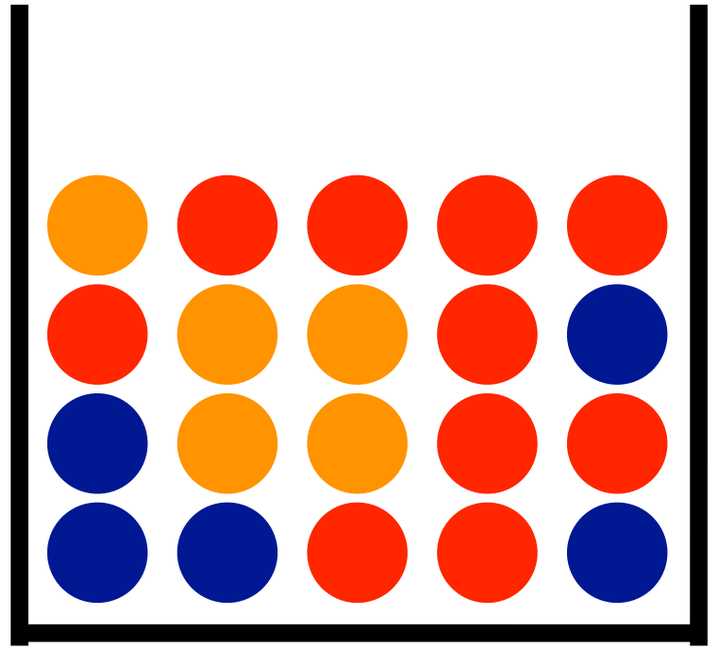


What can we do with a probability distribution?

- If we assume that each outcome is independent of previous outcomes, then the probability of a sequence of outcomes is calculated by multiplying the individual probabilities

- Assumption: when you take out a ball, you put it back in the bag before taking another one out

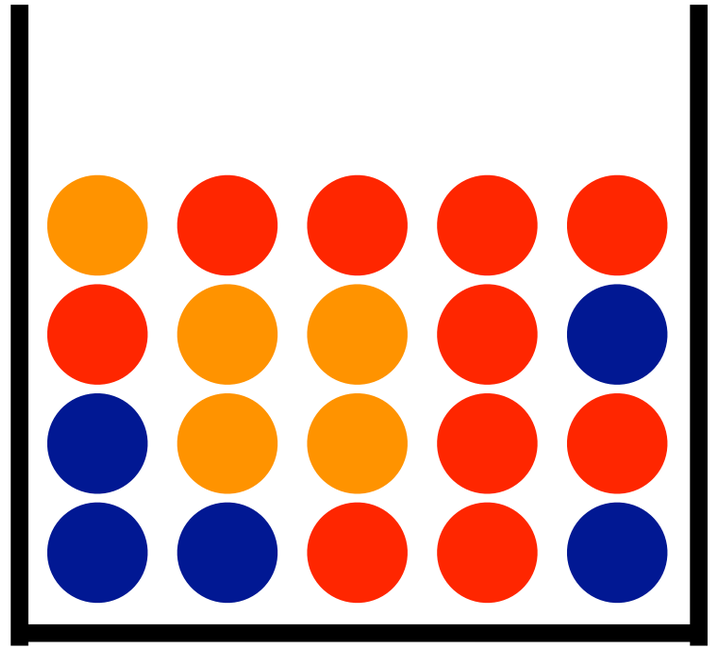
P(RED) = 0.5
P(BLUE) = 0.25
P(ORANGE) = 0.25



What can we do with a probability distribution?

- P(●) = 0.25
- P(●) = 0.5
- P(● ● ●) = 0.25 x 0.25 x 0.25
- P(● ● ●) = 0.25 x 0.25 x 0.25
- P(● ● ●) = 0.25 x 0.50 x 0.25
- P(● ● ● ●) = 0.25 x 0.50 x 0.25 x 0.50

P(RED) = 0.5
P(BLUE) = 0.25
P(ORANGE) = 0.25



Unigram Language Model

- Defines a probability distribution over individual words
  - P(university) = 2/20
  - P(of) = 4/20
  - P(north) = 2/20
  - P(carolina) = 1/20
  - P(at) = 4/20
  - P(chapel) = 3/20
  - P(hill) = 4/20

university university
of of of of
north north
carolina
at at at at
chapel chapel chapel
hill hill hill hill

Unigram Language Model

- It is called a unigram language model because we estimate (and predict) the likelihood of each word independent of any other word
- Assumes that words are independent!
  - The probability of seeing “tarheels” is the same, even if the preceding word is “carolina”
- Other language models take context into account
- Those work better for applications like speech recognition or automatic language translation
- Unigram models work well for information retrieval

Unigram Language Model

- Sequences of words can be assigned a probability by multiplying their individual probabilities:

P(university of north carolina) =

P(university) x P(of) x P(north) x P(carolina) =

(2/20) x (4/20) x (2/20) x (1/20) = 0.0001

P(chapel hill) =

P(chapel) x P(hill) =

(3/20) x (4/20) = 0.03

Document Language Models

- Estimating a document's language model:

  1. tokenize/split the document text into terms

  2. count the number of times each term occurs ($tf_{t,D}$)

  3. count the total number of term occurrences (N_D)

  4. assign term ***t*** a probability equal to:

$$\frac{tf_{t,D}}{N_D}$$

Document Language Models

- The language model estimated from document ***D*** is sometimes denoted as:

$$\theta_D$$

- The probability given to term ***t*** by the language model estimated from document ***D*** is sometimes denoted as:

$$P(t|D) = P(t|\theta_D) = \frac{tf_{t,D}}{N_D}$$

Document Language Models

language model estimation (top 20 terms)

term	$tf_{t,D}$	N_D	*P(term|D)*	*term*	$tf_{t,D}$	N_D	*P(term|D)*
a	22	420	0.05238	creed	5	420	0.01190
rocky	19	420	0.04524	philadelphia	5	420	0.01190
to	18	420	0.04286	has	4	420	0.00952
the	17	420	0.04048	pet	4	420	0.00952
is	11	420	0.02619	boxing	4	420	0.00952
and	10	420	0.02381	up	4	420	0.00952
in	10	420	0.02381	an	4	420	0.00952
for	7	420	0.01667	boxer	4	420	0.00952
his	7	420	0.01667	s	3	420	0.00714
he	6	420	0.01429	balboa	3	420	0.00714

Document Language Models

language model estimation (top 20 terms)

term	$tf_{t,D}$	N_D	*P(term|D)*	*term*	$tf_{t,D}$	N_D	*P(term|D)*
a	**22**	**420**	**0.05238**	creed	5	420	0.01190
rocky	**19**	**420**	**0.04524**	philadelphia	5	420	0.01190
to	18	420	0.04286	has	4	420	0.00952
the	17	420	0.04048	pet	4	420	0.00952
is	**11**	**420**	**0.02619**	boxing	4	420	0.00952
and	10	420	0.02381	up	4	420	0.00952
in	10	420	0.02381	an	4	420	0.00952
for	7	420	0.01667	**boxer**	**4**	**420**	**0.00952**
his	7	420	0.01667	s	3	420	0.00714
he	6	420	0.01429	balboa	3	420	0.00714

- What is the probability given by this language model to the query “rocky is a boxer”?

Document Language Models

language model estimation (top 20 terms)

term	$tf_{t,D}$	N_D	*P(term|D)*	*term*	$tf_{t,D}$	N_D	*P(term|D)*
a	**22**	**420**	**0.05238**	creed	5	420	0.01190
rocky	**19**	**420**	**0.04524**	philadelphia	5	420	0.01190
to	18	420	0.04286	has	4	420	0.00952
the	17	420	0.04048	**pet**	**4**	**420**	**0.00952**
is	**11**	**420**	**0.02619**	boxing	4	420	0.00952
and	10	420	0.02381	up	4	420	0.00952
in	10	420	0.02381	an	4	420	0.00952
for	7	420	0.01667	boxer	4	420	0.00952
his	7	420	0.01667	s	3	420	0.00714
he	6	420	0.01429	balboa	3	420	0.00714

- What is the probability given by this language model to the query "a boxer is a pet"?

Document Language Models

language model estimation (top 20 terms)

term	$tf_{t,D}$	N_D	*P(term|D)*	*term*	$tf_{t,D}$	N_D	*P(term|D)*
a	**22**	**420**	**0.05238**	creed	5	420	0.01190
rocky	**19**	**420**	**0.04524**	philadelphia	5	420	0.01190
to	18	420	0.04286	has	4	420	0.00952
the	17	420	0.04048	pet	4	420	0.00952
is	**11**	**420**	**0.02619**	boxing	4	420	0.00952
and	10	420	0.02381	up	4	420	0.00952
in	10	420	0.02381	an	4	420	0.00952
for	7	420	0.01667	boxer	4	420	0.00952
his	7	420	0.01667	s	3	420	0.00714
he	6	420	0.01429	balboa	3	420	0.00714

- What is the probability given by this language model to the query "a boxer is a dog"?

Query-Likelihood Retrieval Model

$$score(Q, D) = P(Q|\theta_D) = \prod_{i=1}^{n} P(q_i|\theta_D)$$

- There is one two issues with this scoring function
- What are they?

Query-Likelihood Retrieval Model

- A document with a single missing query-term will receive a score of zero (similar to boolean AND)
- Where is IDF?

- Don't we want to suppress the contribution of terms that are frequent in the document, but frequent in general (appear in many documents)?

Linear Interpolation Smoothing

- Let θ_D denote the language model associated with document *D*
- Let θ_C denote the language model associated with the entire collection
- Using linear interpolation, the probability given by the document language model to term ***t*** is:

$$P(t|D) = \lambda P(t|\theta_D) + (1 - \lambda)P(t|\theta_C)$$

Linear Interpolation Smoothing

$$P(t|D) = \lambda P(t|\theta_D) + (1-\lambda) P(t|\theta_C)$$

the probability given to the term by the document language model

the probability given to the term by the collection language model

Linear Interpolation Smoothing

$$P(t|D) = \lambda P(t|\theta_D) + (1-\lambda) P(t|\theta_C)$$

every one of these numbers
is between 0 and 1, so P(*t*|*D*)
is between 0 and 1

Query Likelihood Retrieval Model
with linear interpolation smoothing

- As before, a document's score is given by the probability that it "generated" the query
- As before, this is given by multiplying the individual query-term probabilities
- However, the probabilities are obtained using the linearly interpolated language model

$$score(Q, D) = P(Q|D) = \prod_{i=1}^{n} \left(\lambda P(t|\theta_D) + (1 - \lambda) P(t|\theta_C) \right)$$

Query Likelihood Retrieval Model
with linear interpolation smoothing

- Linear interpolation helps us avoid zero-probabilities
- Remember, because we're multiplying probabilities, if a document is missing a single query-term it will be given a score of zero!
- Linear interpolation smoothing has another added benefit, though it's not obvious
- Let's start with an example

Query Likelihood Retrieval Model
no smoothing

- Query: apple ipad
- Two documents (D_1 and D_2), each with 50 term occurrences

	D_1 $(N_{D1}=50)$	D_2 $(N_{D2}=50)$
apple	2/50 = 0.04	3/50 = 0.06
ipad	3/50 = 0.06	2/50 = 0.04
score	(0.04 x 0.06) = 0.0024	(0.06 x 0.04) = 0.0024

Query Likelihood Retrieval Model
no smoothing

- Query: apple ipad
- Two documents (D_1 and D_2), each with 50 term occurrences

	D_1 $(N_{D1}=50)$	D_2 $(N_{D2}=50)$
apple	2/50 = 0.04	3/50 = 0.06
ipad	3/50 = 0.06	2/50 = 0.04
score	(0.04 x 0.06) = 0.0024	(0.06 x 0.04) = 0.0024

- Which query-term is more important: apple or ipad?

Query Likelihood Retrieval Model
no smoothing

- A term is descriptive of the document if it occurs many times in the document
- But, not if it occurs many times in the document and also occurs frequently in the collection

Query Likelihood Retrieval Model
no smoothing

- Query: apple ipad
- Two documents (D_1 and D_2), each with 50 term occurrences

	D_1 $(N_{D1}=50)$	D_2 $(N_{D2}=50)$
apple	2/50 = 0.04	3/50 = 0.06
ipad	3/50 = 0.06	2/50 = 0.04
score	(0.04 x 0.06) = 0.0024	(0.06 x 0.04) = 0.0024

- Without smoothing, the query-likelihood model ignores how frequently the term occurs in general!

Query Likelihood Retrieval Model
with linear interpolation smoothing

- Suppose the corpus has 1,000,000 term-occurrences
- apple occurs 200 / 1,000,000 times
- ipad occurs 100 / 1,000,000 times
- Therefore:

$$P(\text{apple}|\theta_C) = \frac{200}{1000000} = 0.0002$$

$$P(\text{ipad}|\theta_C) = \frac{100}{1000000} = 0.0001$$

Query Likelihood Retrieval Model

with linear interpolation smoothing

$$score(Q, D) = \prod_{i=1}^{n} \left(\lambda P(t|\theta_D) + (1 - \lambda) P(t|\theta_C) \right)$$

	D_1 (N_{D1}=50)	D_2 (N_{D2}=50)
P(apple|D)	0.04	0.06
P(apple|C)	0.0002	0.0002
score(apple)	0.0201	0.0301
P(ipad|D)	0.06	0.04
P(ipad|C)	0.0001	0.0001
score(ipad)	0.03005	0.02005
total score	0.000604005	0.000603505

$$\lambda = 0.50$$

Outline

Information Retrieval

Search Engine Components

Document Representation

Retrieval Models

Evaluation

Federated Search and Cross-lingual IR

Open-source Toolkits

Information Retrieval Evaluation

- Evaluation is a fundamental issue of information retrieval
  - an area of IR research in its own right
- Evaluation methods:
  - test collection evaluation
  - user-study evaluation
  - online evaluation
- Each method has advantages and disadvantages

Test Collection Evaluation

Test Collection Evaluation
motivation

- Many factors affect search engine effectiveness:
  - Queries: some queries are easier than others
  - Corpus: the number of documents that are relevant to a query will vary across corpora
  - Relevance judgements: relevance is subjective
  - The IR system: the document representation and retrieval model

Test Collection Evaluation

motivation

- Comparing different IR systems requires a controlled experimental setting
- Test collection evaluation: vary the IR system, but hold everything else constant
- Evaluate systems using metrics that measure the quality of a system's output ranking
- Known as the Cranfield Methodology
  - developed in the 60's and still widely used

Test Collection Evaluation

overview

- Collect a set of queries (e.g., 50)
- For each query, describe a hypothetical information need
- For each information need, have a human assessor determine which documents are relevant/non-relevant
- Evaluate systems based on the quality of their rankings
  - evaluation metric: describes the quality of a ranking with known relevant/non-relevant docs

Test Collection Evaluation

queries

- QUERY: parenting
- DESCRIPTION: Relevant blogs include those from parents, grandparents, or others involved in parenting, raising, or caring for children. Blogs can include those provided by health care providers if the focus is on children. Blogs that serve primarily as links to other sites or market products related to children and their caregivers are not relevant.

(TREC Blog Track 2009)

Test Collection Evaluation

relevance judgements

- Which documents should be judged for relevance?
  - Only the ones that contain all query-terms?
  - Only the ones that contain at least one query-term?
  - All the documents in the collection?

Test Collection Evaluation

relevance judgements

- Which documents should be judged for relevance?
  - Only the ones that contain all query-terms?
  - Only the ones that contain at least one query-term?
  - All the documents in the collection?
- The best solution is to judge all of them
  - A document can be relevant without having a single query-term
  - But, there's a problem...

Test Collection Evaluation

relevance judgements

- GOV2 (2004)
  - 25,000,000 Web-pages
  - Crawl of entire ".gov" Web domain
- BLOG08 (January 2008 - February 2009)
  - 1,303,520 blogs "polled" once a week for new posts
  - 28,488,766 posts
- ClueWeb09 (2009)
  - 1,040,809,705 Web-pages

Test Collection Evaluation

relevance judgements

- Which documents should be judged for relevance?
  - Only the ones that contain all query-terms?
  - Only the ones that contain at least one query-term?
  - All the documents in the collection?
- The best solution is to judge all of them
  - A document can be relevance without having a single query-term
- Is this feasible?

Test Collection Evaluation
pooling

- Given any query, the overwhelming majority of documents are not relevant
- General Idea:
  - Identify the documents that are most likely to be relevant
  - Have assessors judge only those documents
  - Assume the remaining ones are non-relevant

Test Collection Evaluation

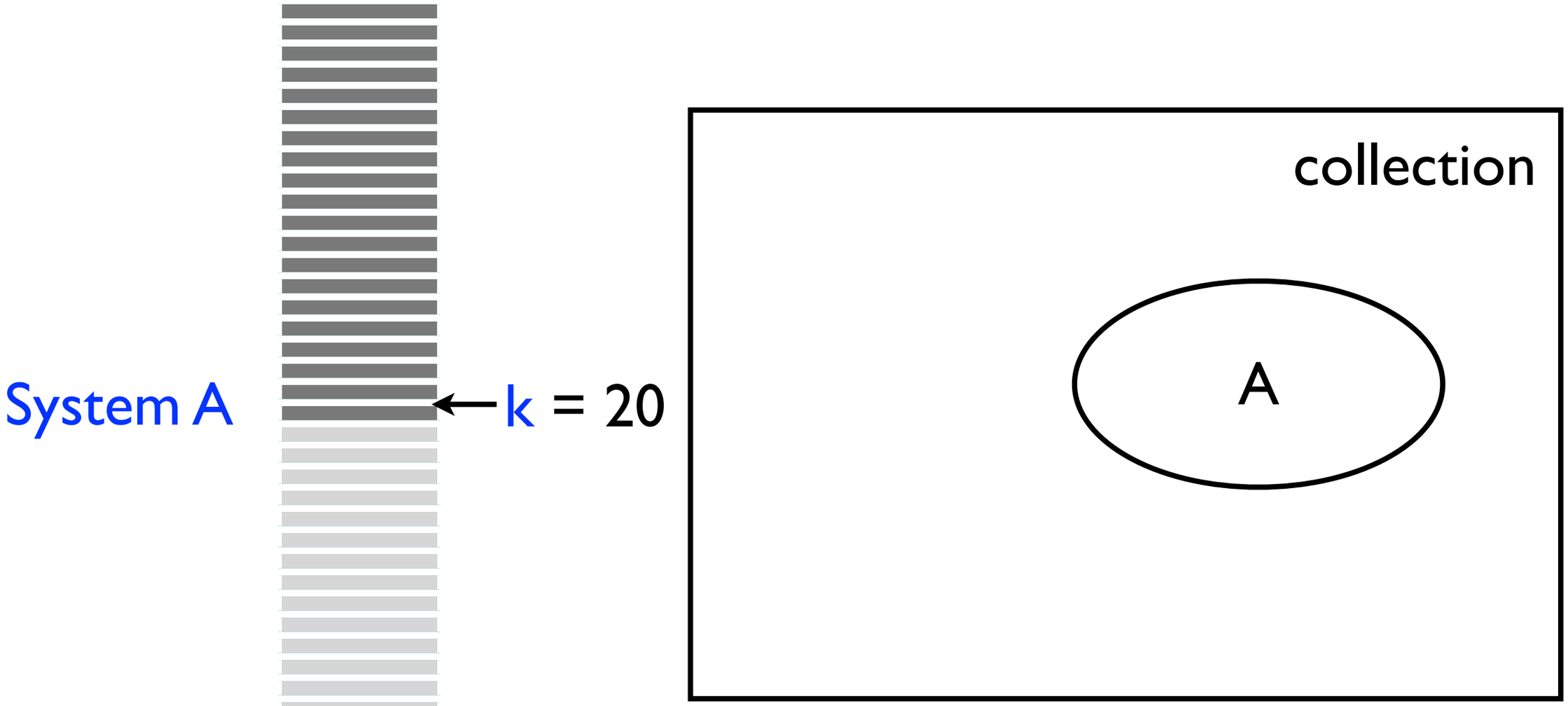
pooling

System A

collection

Test Collection Evaluation

pooling

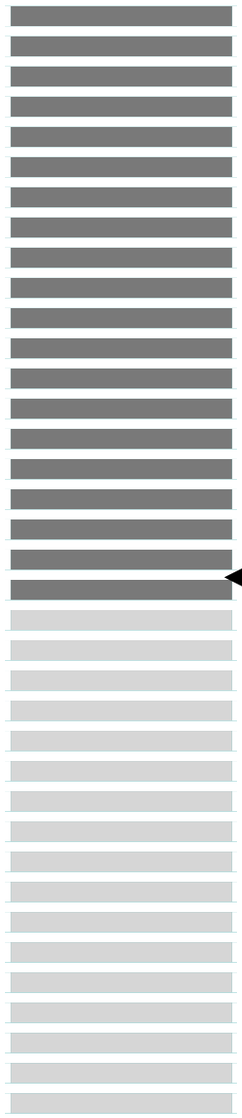


k = depth of the pool

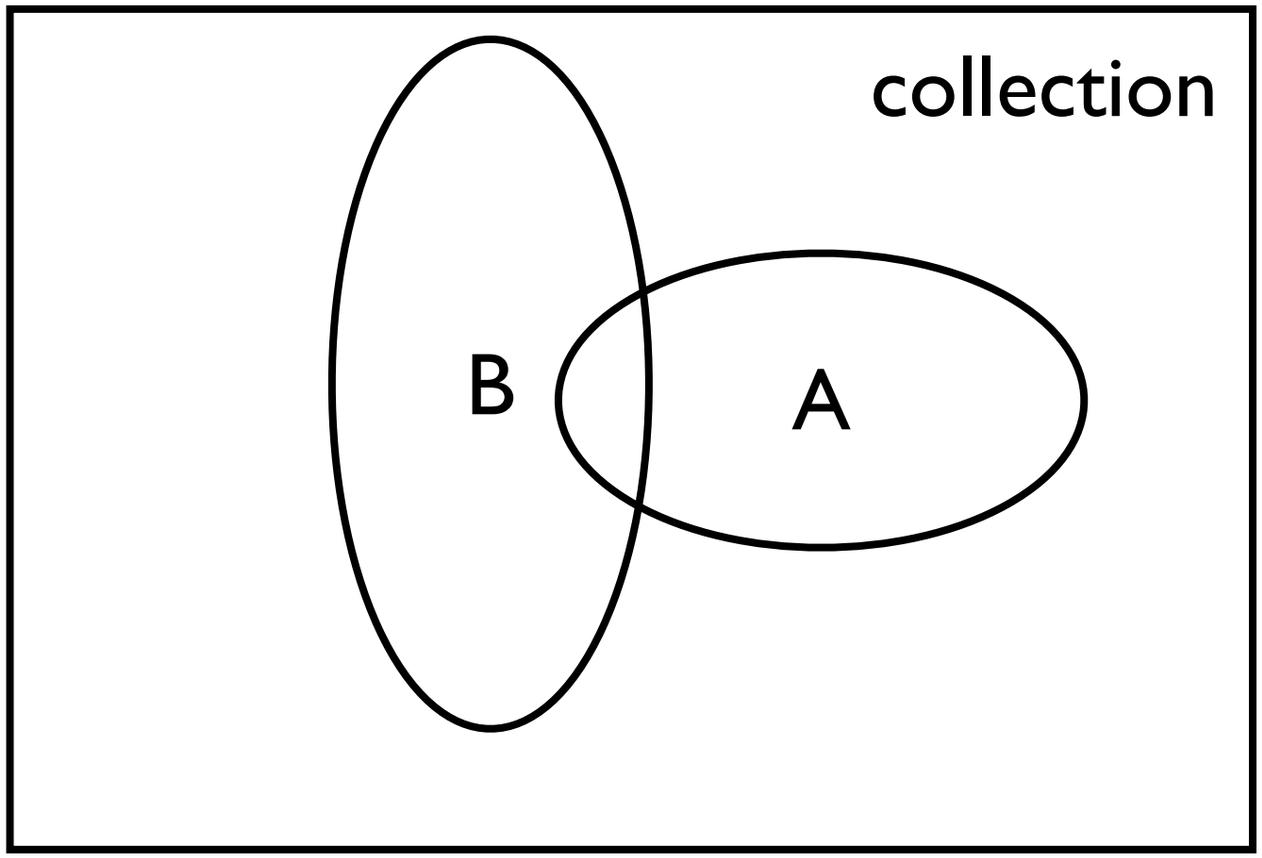
Test Collection Evaluation

pooling

System B



← k = 20

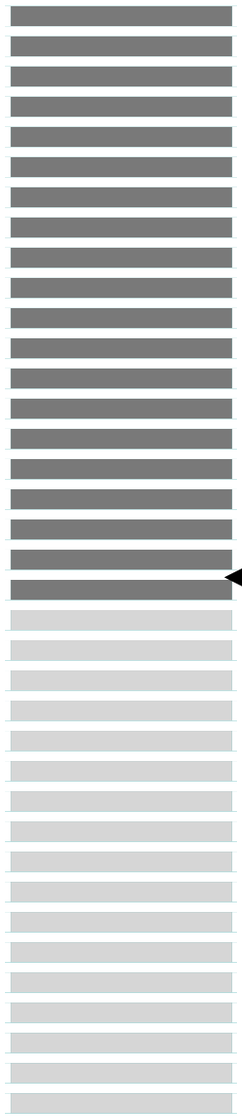


k = depth of the pool

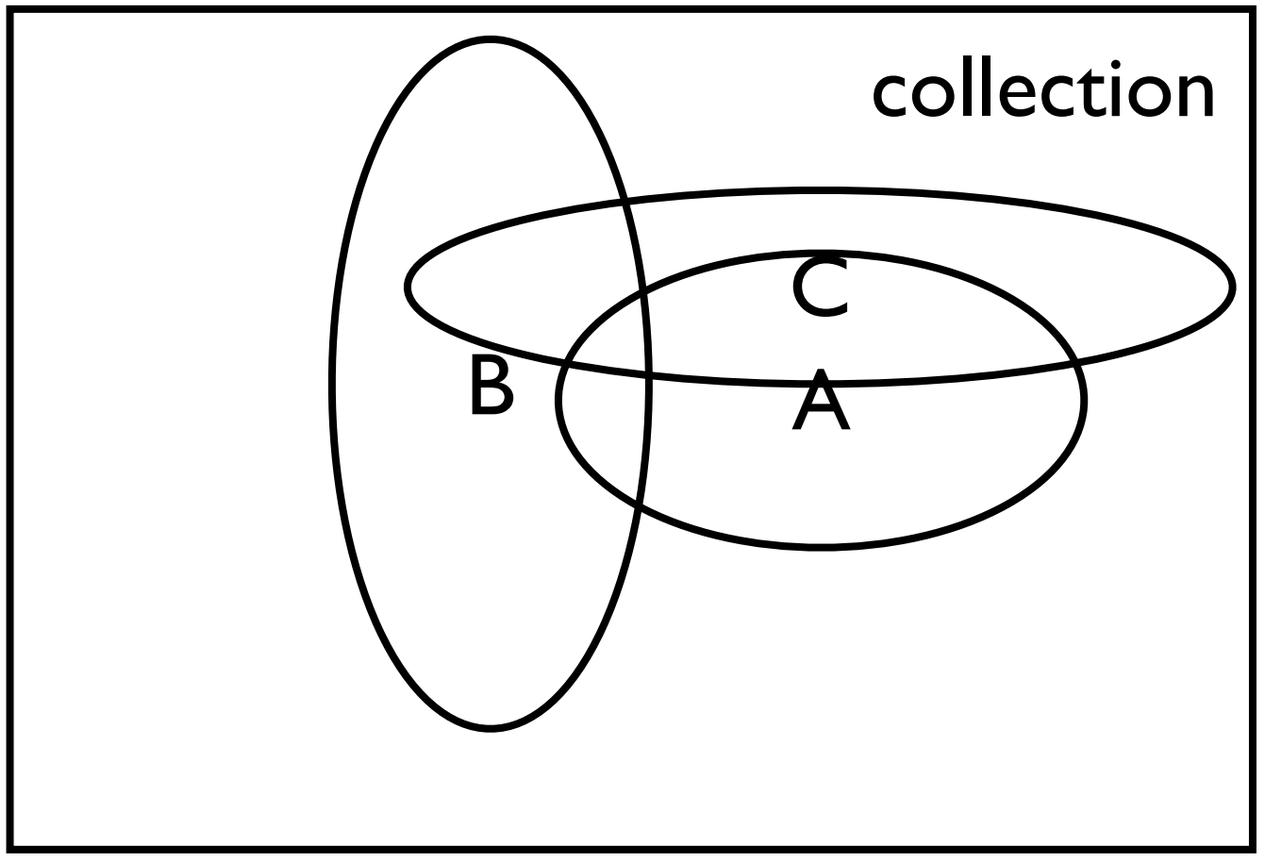
Test Collection Evaluation

pooling

System C



k = 20

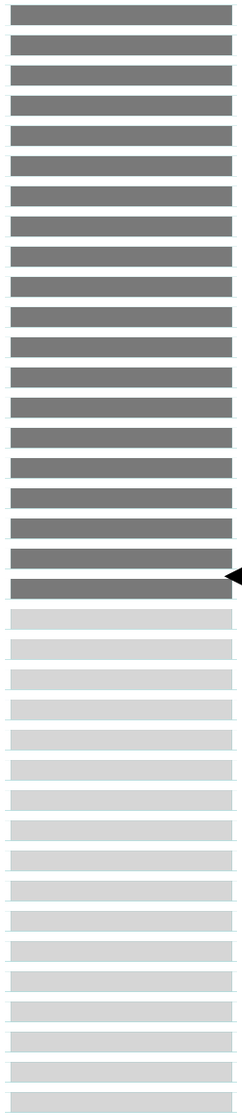


k = depth of the pool

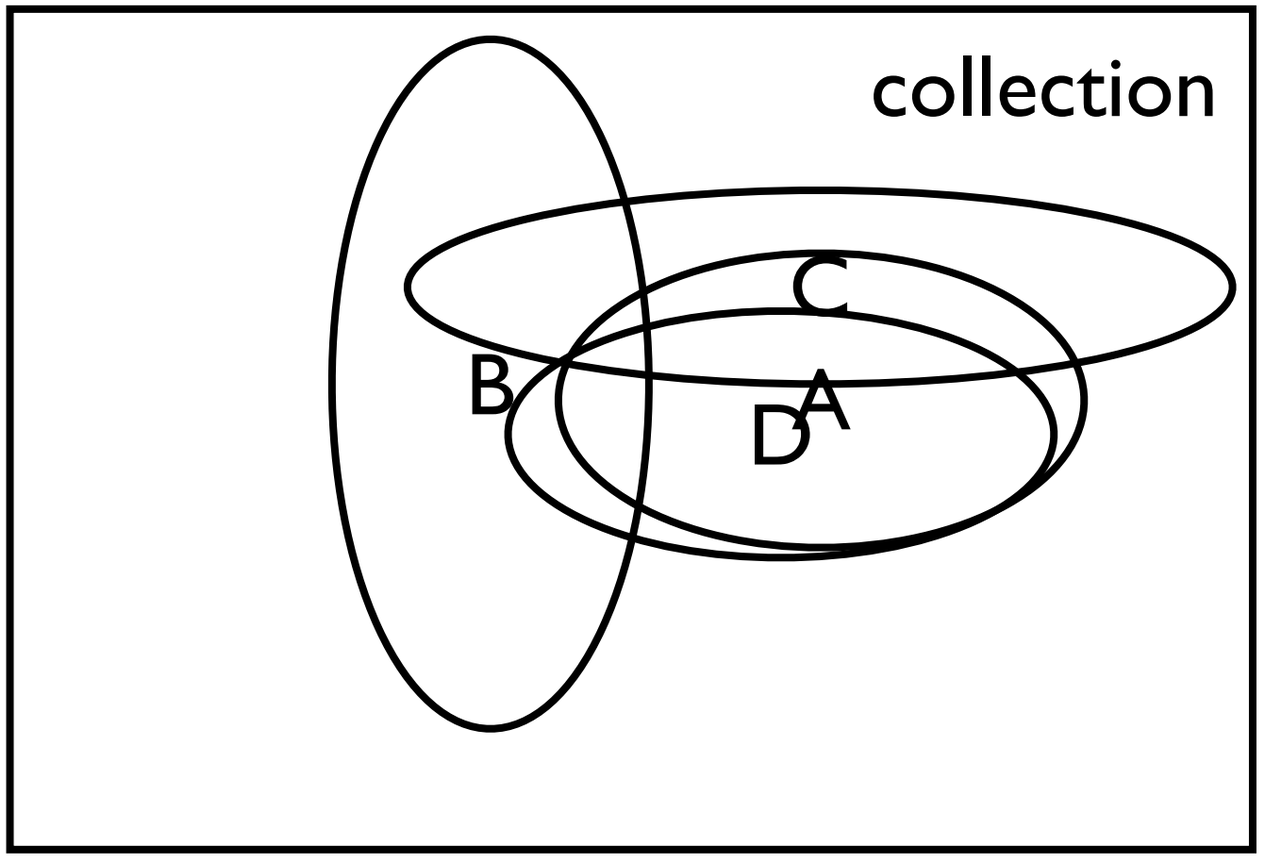
Test Collection Evaluation

pooling

System D



← k = 20

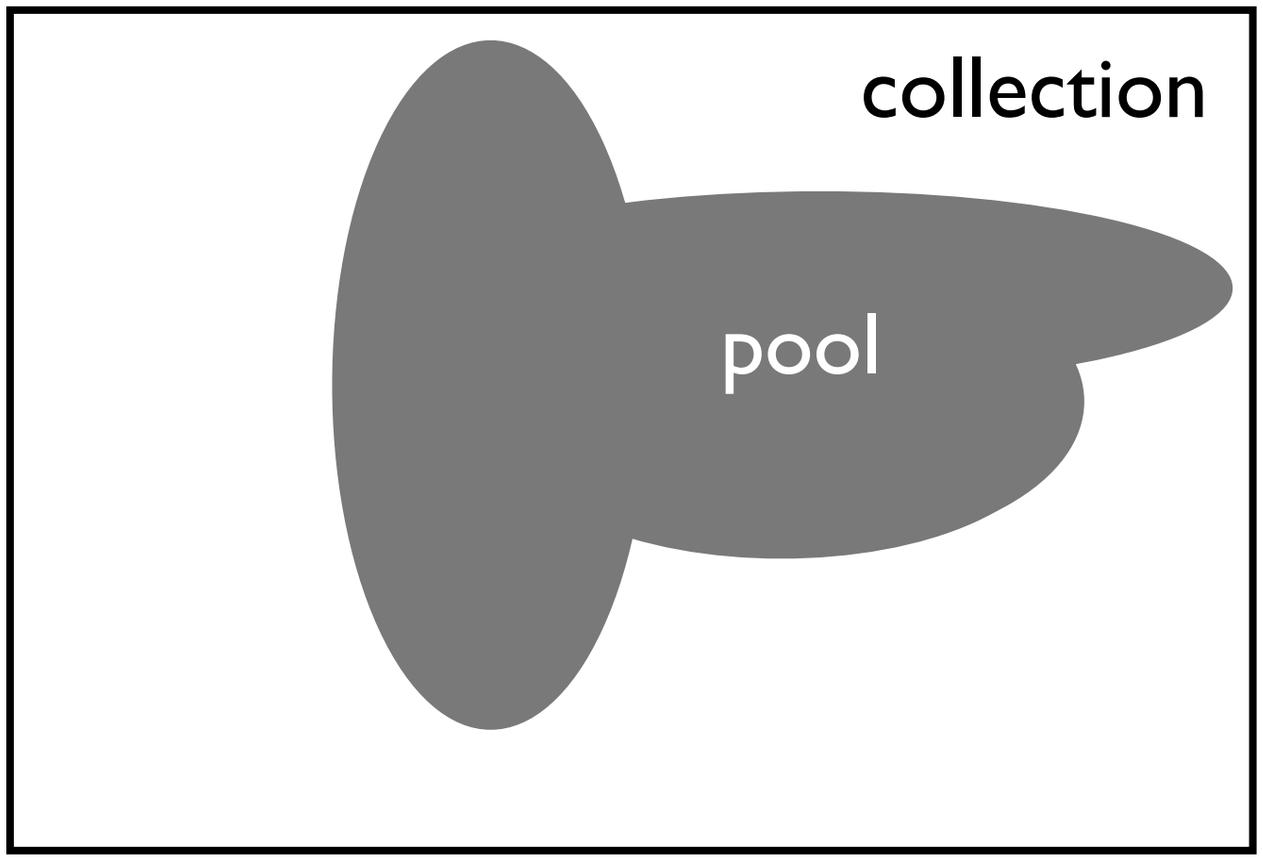


•
•
•

k = depth of the pool

Test Collection Evaluation

pooling



Test Collection Evaluation
pooling

- Take the top-k documents retrieved by various systems
- Remove duplicates
- Show to the assessor in random order (along with the information need description)
- Assume that documents outside the pool are non-relevant

Test Collection Evaluation
pooling

- Usually the depth (k) of the pool is between 50 and 200 and the number of systems included in the pool is between 10 and 20
- A test-collection constructed using pooling can be used to evaluate systems that were not in the original pool
- However, what is the risk?

Evaluation Metrics

Test Collection Evaluation

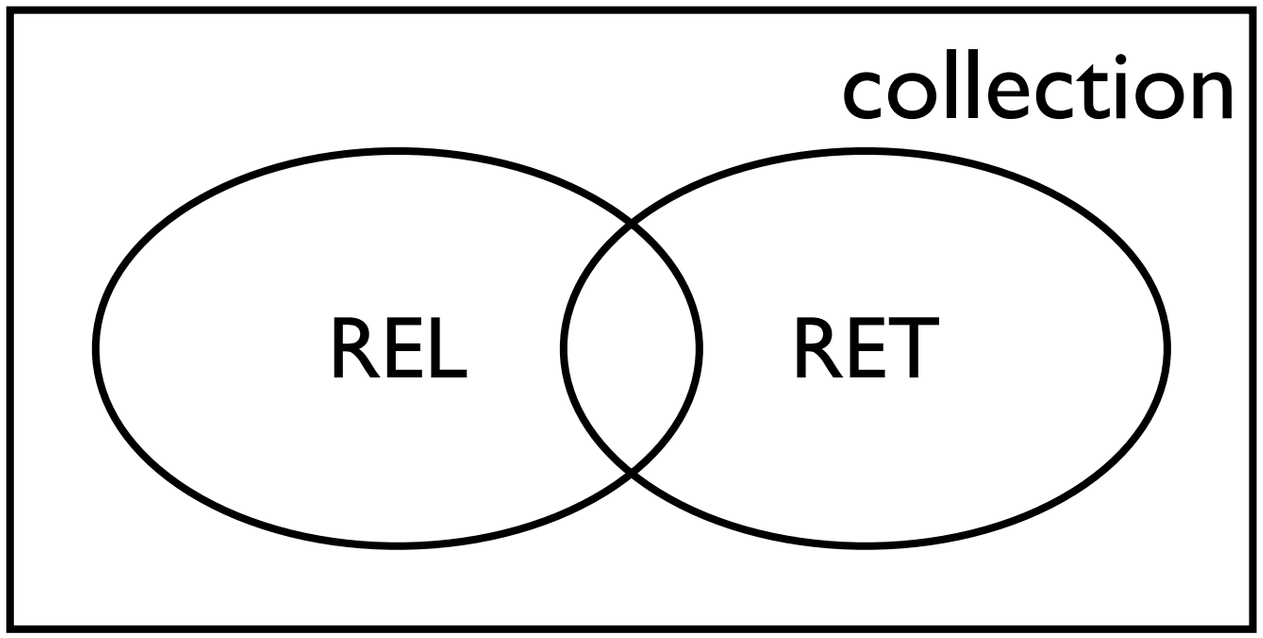
evaluation metrics

- At this point, we have a set of queries, with identified relevant and non-relevant documents
- The goal of an evaluation metric is to measure the quality of a set or ranking of known relevant/non-relevant documents

Unranked Boolean Retrieval

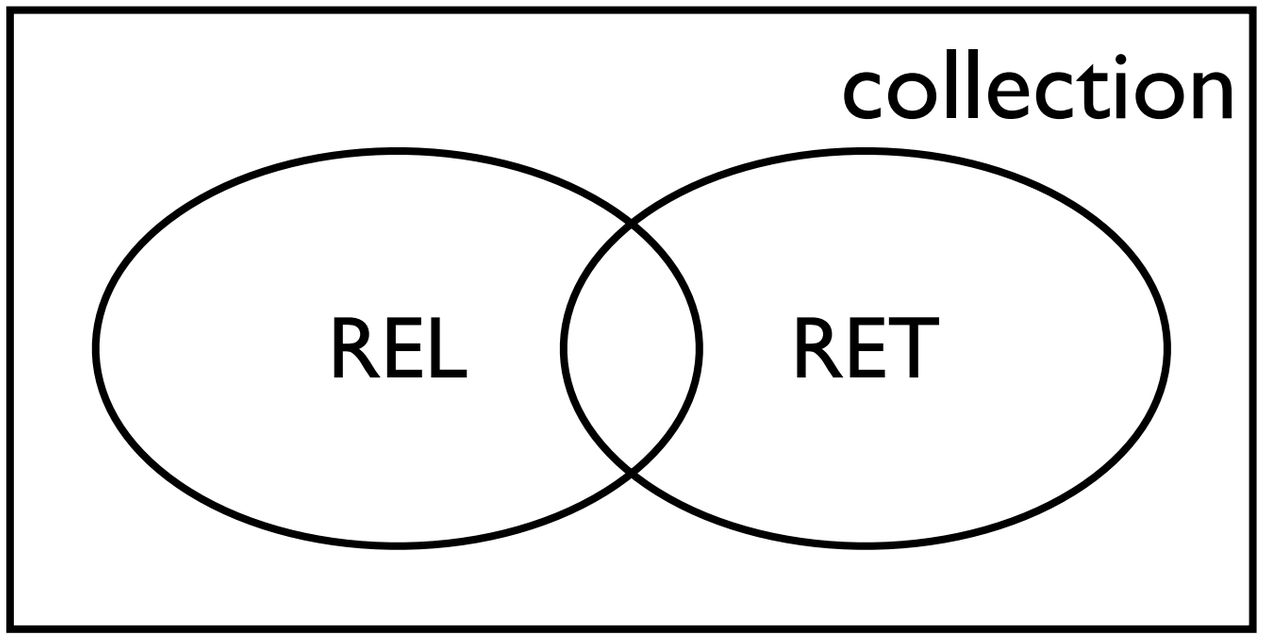
precision and recall

- A set of relevant documents (REL) and a set of retrieved documents (RET)



Unranked Boolean Retrieval
precision and recall

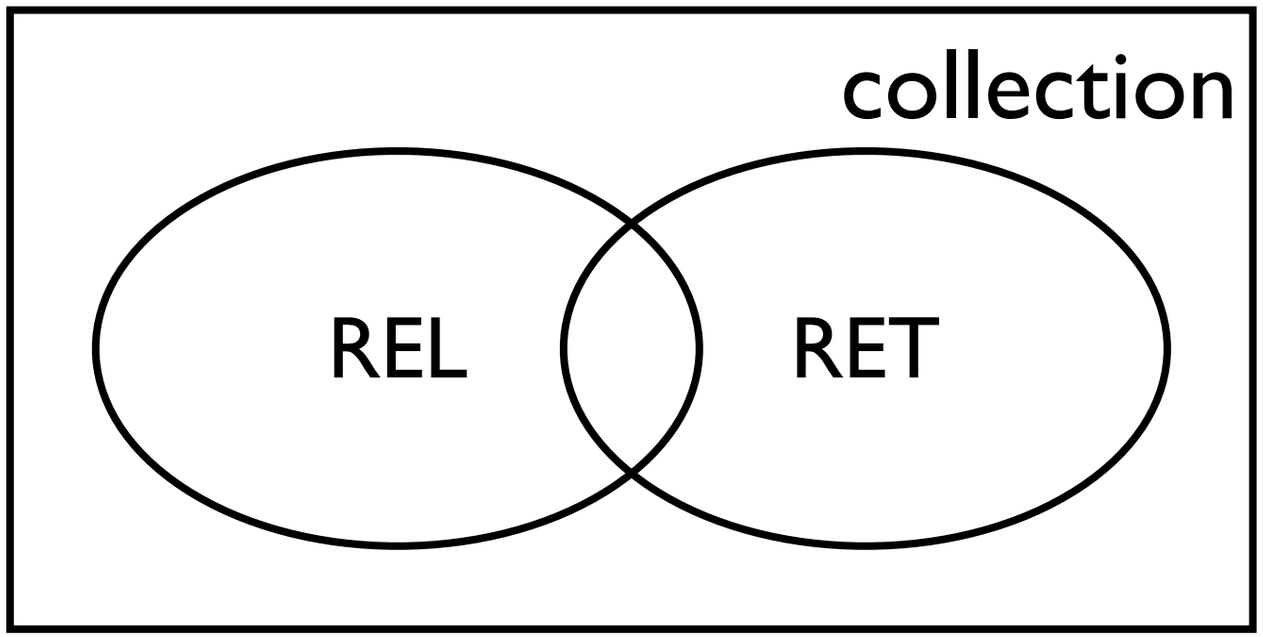
- Precision (P): the proportion of retrieved documents that are relevant



$$\mathcal{P} = \frac{|RET \cap REL|}{|RET|}$$

Unranked Boolean Retrieval
precision and recall

- Recall (R): the proportion of relevant documents that are retrieved



$$\mathcal{R} = \frac{|RET \cap REL|}{|REL|}$$

Unranked Boolean Retrieval
precision and recall

- Recall measures the system's ability to find all the relevant documents
- Precision measures the system's ability to reject any non-relevant documents

Unranked Boolean Retrieval
precision and recall

- A system can make two types of errors:
  - a false positive error: the system retrieves a document that is not relevant (should not have been retrieved)
  - a false negative error: the system fails to retrieve a document that is relevant (should have been retrieved)
- How do these types of errors affect precision and recall?

Unranked Boolean Retrieval
precision and recall

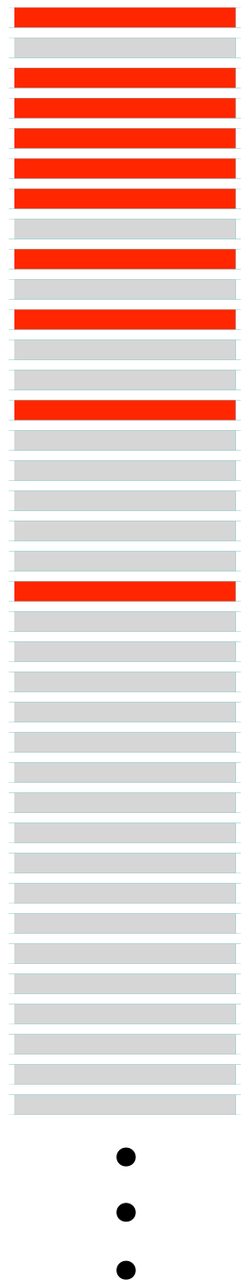
- A system can make two types of errors:
  - a false positive error: the system retrieves a document that is not relevant (should not have been retrieved)
  - a false negative error: the system fails to retrieve a document that is relevant (should have been retrieved)
- How do these types of errors affect precision and recall?
- Precision is affected by the number of false positive errors
- Recall is affected by the number of false negative errors

Ranked Retrieval
precision and recall

- In most situations, the system outputs a ranked list of documents rather than an unordered set
- User-behavior assumption:
  - The user examines the output ranking from top-to-bottom until he/she is satisfied or gives up
- Precision/Recall @ rank cut-off *K*

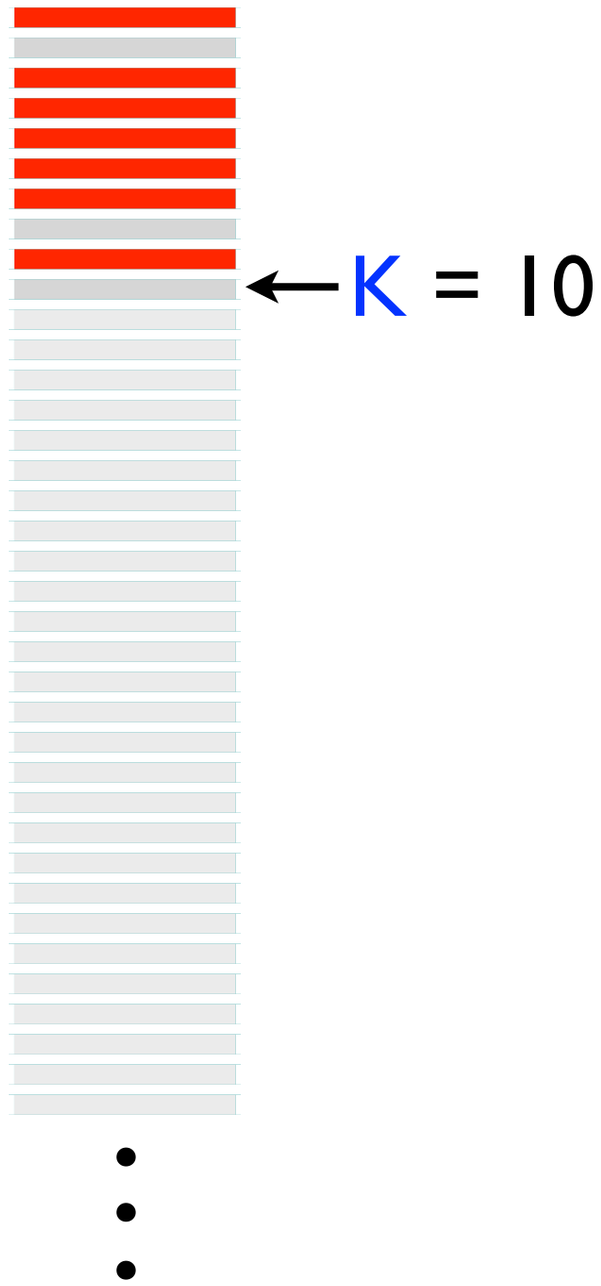
Ranked Retrieval
precision and recall

- Precision: proportion of retrieved documents that are relevant
- Recall: proportion of relevant documents that are retrieved



Ranked Retrieval
precision and recall

- P@K: proportion of ~~retrieved~~ top-K documents that are relevant
- R@K: proportion of relevant documents that are ~~retrieved~~ in the top-K
- Assumption: the user will only examine the top-K results



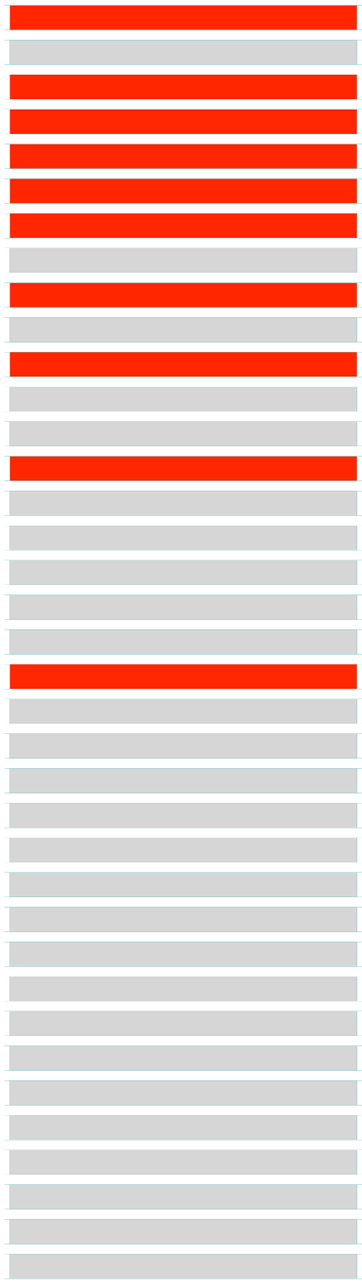
Ranked Retrieval

precision and recall

- Assume 20 relevant documents

K	P@K	R@K
1	(1/1) = 1.0	(1/20) = 0.05
2	(1/2) = 0.5	(1/20) = 0.05
3	(2/3) = 0.67	(2/20) = 0.10
4	(3/4) = 0.75	(3/20) = 0.15
5	(4/5) = 0.80	(4/20) = 0.20
6	(5/6) = 0.83	(5/20) = 0.25
7	(6/7) = 0.86	(6/20) = 0.30
8	(6/8) = 0.75	(6/20) = 0.30
9	(7/9) = 0.78	(7/20) = 0.35
10	(7/10) = 0.70	(7/20) = 0.35

K = 10



Ranked Retrieval
motivation: average precision

- Ideally, we want the system to achieve high precision for varying values of K

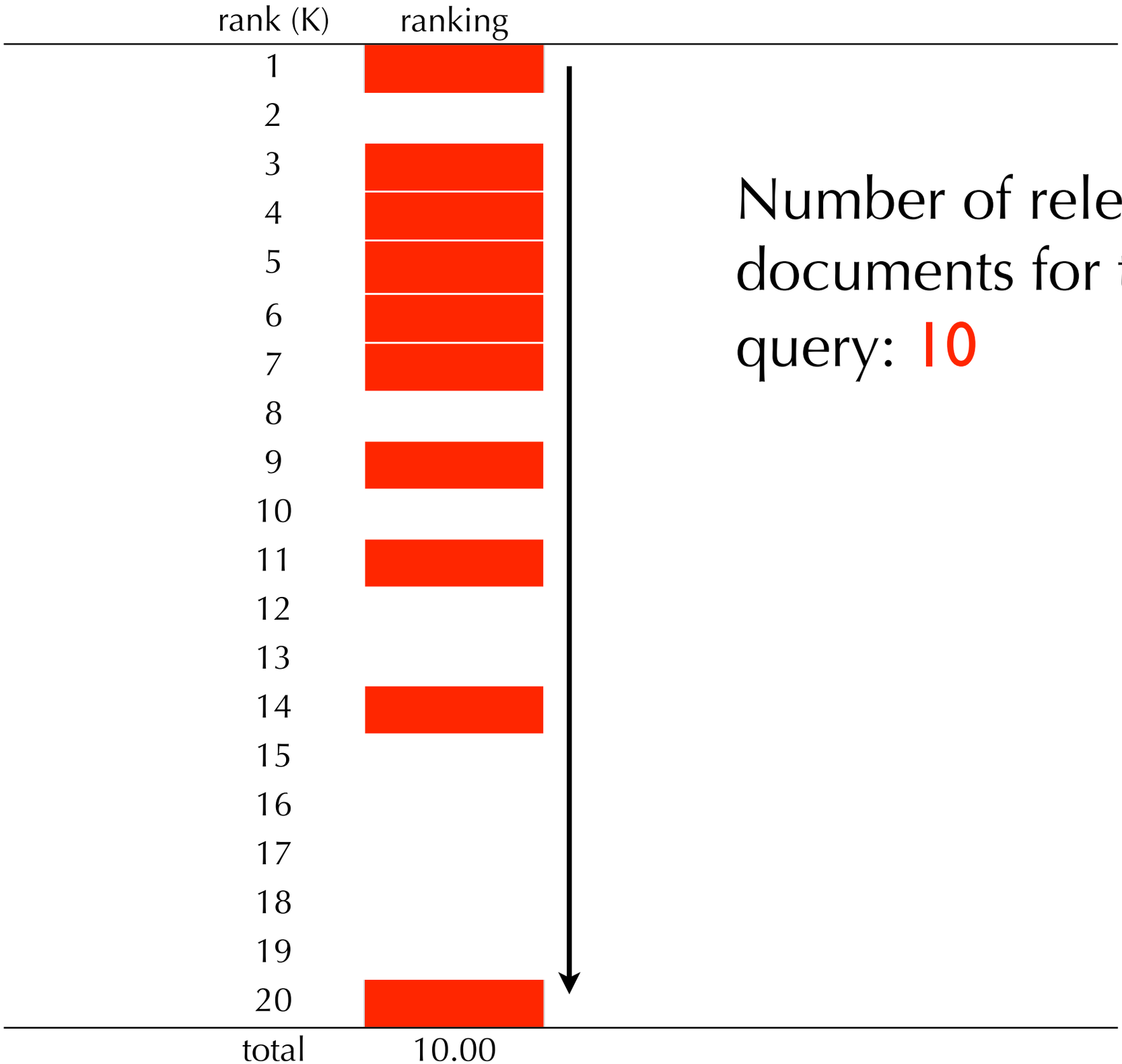
- The metric average precision accounts for precision and recall without having to set K

Ranked Retrieval
average precision

1. Go down the ranking one-rank-at-a-time
2. If the document at rank K is relevant, measure P@K
   - proportion of top-K documents that are relevant
3. Finally, take the average of all P@K values
   - the number of P@K values will equal the number of relevant documents

Ranked Retrieval

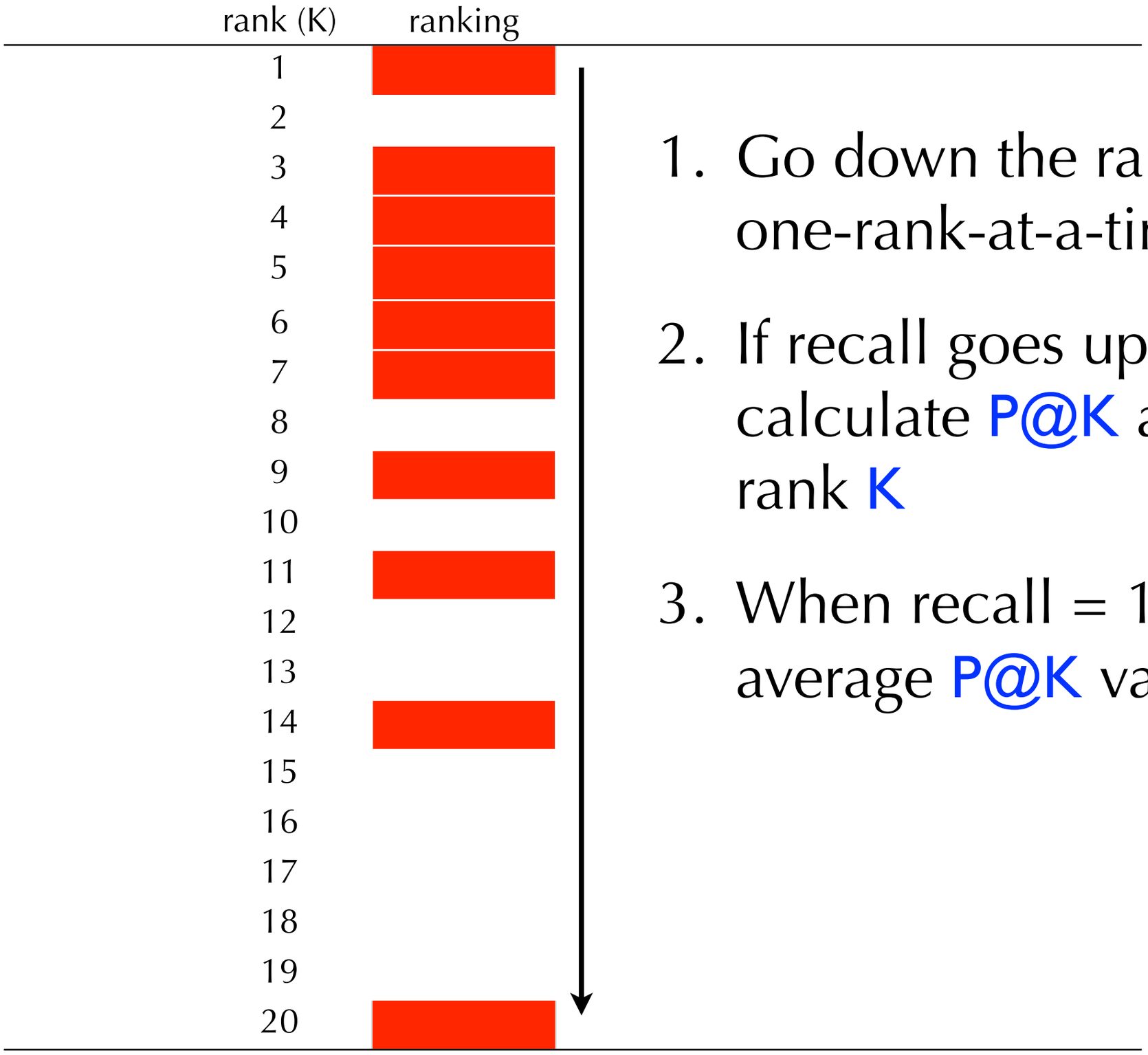
average-precision



Number of relevant documents for this query: 10

Ranked Retrieval

average-precision



1. Go down the ranking one-rank-at-a-time
2. If recall goes up, calculate P@K at that rank K
3. When recall = 1.0, average P@K values

Ranked Retrieval

average-precision

rank (K)	ranking	R@K	P@K
1		0.10	1.00
2		0.10	0.50
3		0.20	0.67
4		0.30	0.75
5		0.40	0.80
6		0.50	0.83
7		0.60	0.86
8		0.60	0.75
9		0.70	0.78
10		0.70	0.70
11		0.80	0.73
12		0.80	0.67
13		0.80	0.62
14		0.90	0.64
15		0.90	0.60
16		0.90	0.56
17		0.90	0.53
18		0.90	0.50
19		0.90	0.47
20		1.00	0.50
total	10.00	average-precision	0.76

Ranked Retrieval

average-precision

swapped ranks 2 and 3

rank (K)	ranking	R@K	P@K
1		0.10	1.00
2		0.20	1.00
3		0.20	0.67
4		0.30	0.75
5		0.40	0.80
6		0.50	0.83
7		0.60	0.86
8		0.60	0.75
9		0.70	0.78
10		0.70	0.70
11		0.80	0.73
12		0.80	0.67
13		0.80	0.62
14		0.90	0.64
15		0.90	0.60
16		0.90	0.56
17		0.90	0.53
18		0.90	0.50
19		0.90	0.47
20		1.00	0.50
total	10.00	average-precision	0.79

Ranked Retrieval

average-precision

rank (K)	ranking	R@K	P@K
1		0.10	1.00
2		0.10	0.50
3		0.20	0.67
4		0.30	0.75
5		0.40	0.80
6		0.50	0.83
7		0.60	0.86
8		0.60	0.75
9		0.70	0.78
10		0.70	0.70
11		0.80	0.73
12		0.80	0.67
13		0.80	0.62
14		0.90	0.64
15		0.90	0.60
16		0.90	0.56
17		0.90	0.53
18		0.90	0.50
19		0.90	0.47
20		1.00	0.50
total	10.00	average-precision	0.76

Ranked Retrieval

average-precision

swapped ranks 8 and 9

rank (K)	ranking	R@K	P@K
1		0.10	1.00
2		0.10	0.50
3		0.20	0.67
4		0.30	0.75
5		0.40	0.80
6		0.50	0.83
7		0.60	0.86
8		0.70	0.88
9		0.70	0.78
10		0.70	0.70
11		0.80	0.73
12		0.80	0.67
13		0.80	0.62
14		0.90	0.64
15		0.90	0.60
16		0.90	0.56
17		0.90	0.53
18		0.90	0.50
19		0.90	0.47
20		1.00	0.50
total	10.00	average-precision	0.77

Ranked Retrieval
average precision

- Advantages:
  - no need to choose K
  - accounts for both precision and recall
  - mistakes at the top are more influential
  - mistakes at the bottom are still accounted for
- Disadvantages
  - not quite as easy to interpret as P/R@K

Ranked Retrieval

MAP: mean average precision

- We've talked about average precision for a single query
- Mean Average Precision (MAP): average precision averaged across a set of queries
  - yes, confusing. but, better than calling it "average average precision"!
  - one of the most common metrics in IR evaluation

Ranked Retrieval
precision-recall curves

- In some situations, we want to understand the trade-off between precision and recall

- A precision-recall (PR) curve expresses precision as a function of recall

Ranked Retrieval

precision-recall curves: general idea

- Different tasks require different levels of recall
- Sometimes, the user wants a few relevant documents
- Other times, the user wants most of them

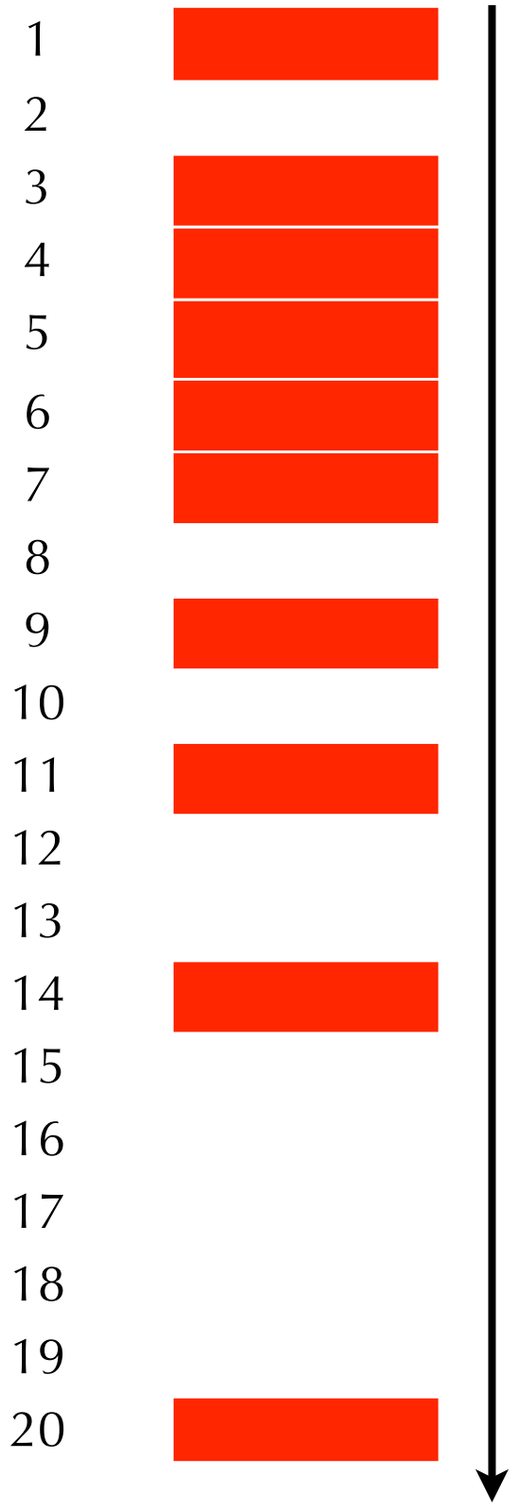
- Suppose a user wants some level of recall R
- The goal for the system is to minimize the number of false negatives the user must look at in order to achieve a level of recall R

Ranked Retrieval
precision-recall curves: general idea

- False negative error: not retrieving a relevant document
  - false negative errors affects recall
- False positive errors: retrieving a non-relevant document
  - false positives errors affects precision
- If a user wants to avoid a certain level of false-negatives, what is the level of false-positives he/she must filter through?

Ranked Retrieval

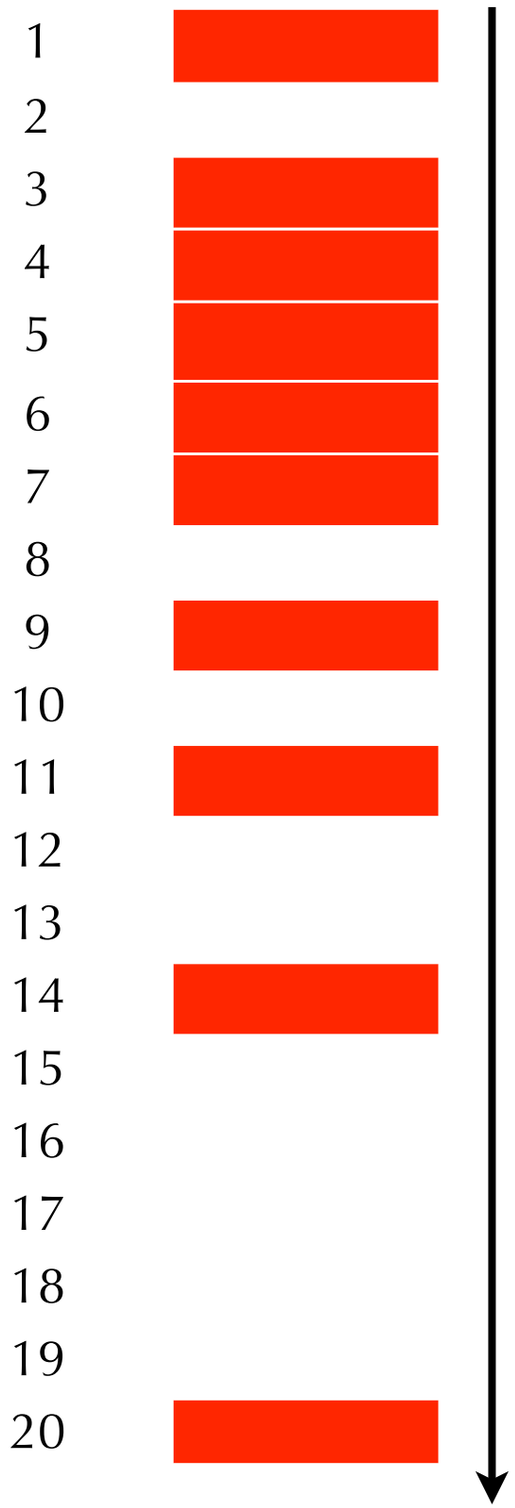
precision-recall curves



- Assume 10 relevant documents for this query
- Suppose the user wants R = (1/10)
- What level of precision will the user observe?

Ranked Retrieval

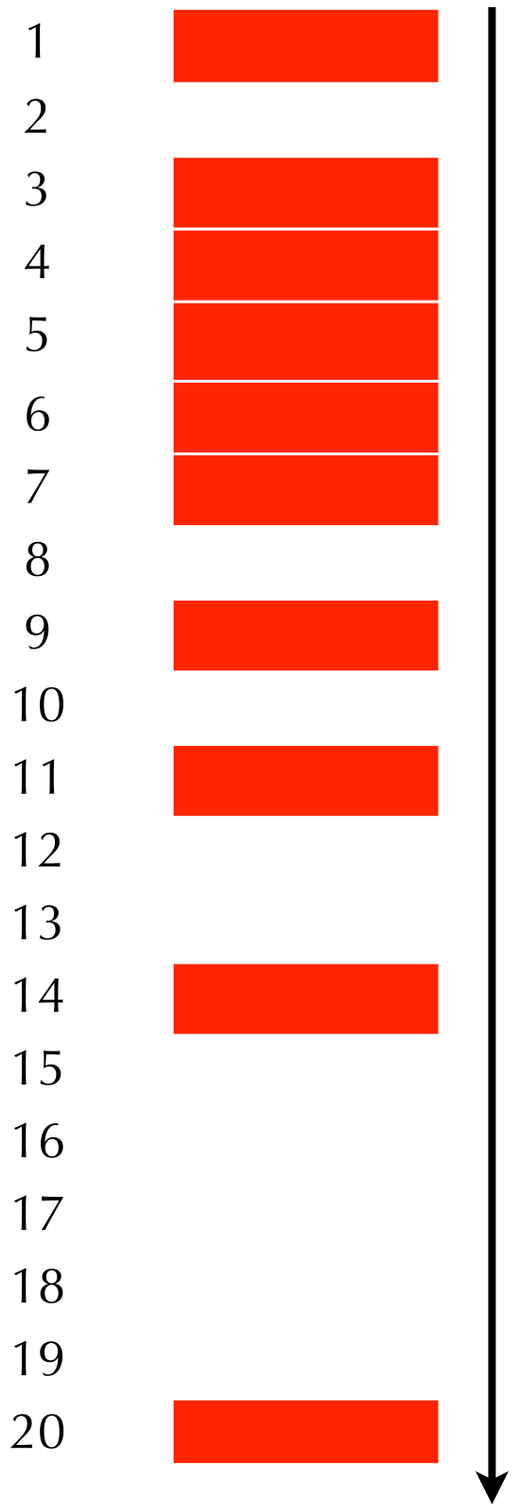
precision-recall curves



- Assume 10 relevant documents for this query
- Suppose the user wants R = (2/10)
- What level of precision will the user observe?

Ranked Retrieval

precision-recall curves



- Assume 10 relevant documents for this query
- Suppose the user wants R = (10/10)
- What level of precision will the user observe?

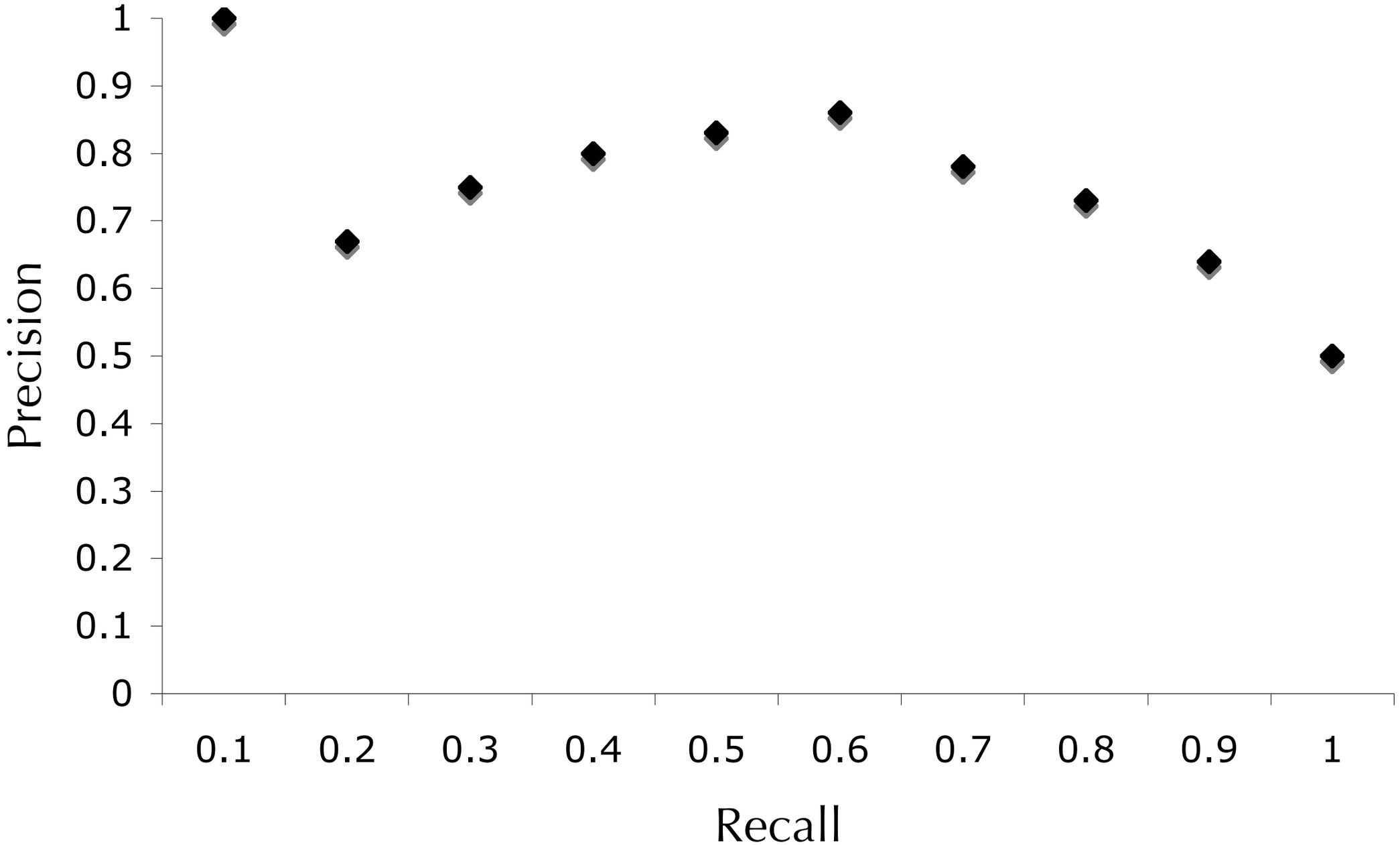
Ranked Retrieval

precision-recall curves

rank (K)	ranking	R@K	P@K
1		0.10	1.00
2		0.10	0.50
3		0.20	0.67
4		0.30	0.75
5		0.40	0.80
6		0.50	0.83
7		0.60	0.86
8		0.60	0.75
9		0.70	0.78
10		0.70	0.70
11		0.80	0.73
12		0.80	0.67
13		0.80	0.62
14		0.90	0.64
15		0.90	0.60
16		0.90	0.56
17		0.90	0.53
18		0.90	0.50
19		0.90	0.47
20		1.00	0.50

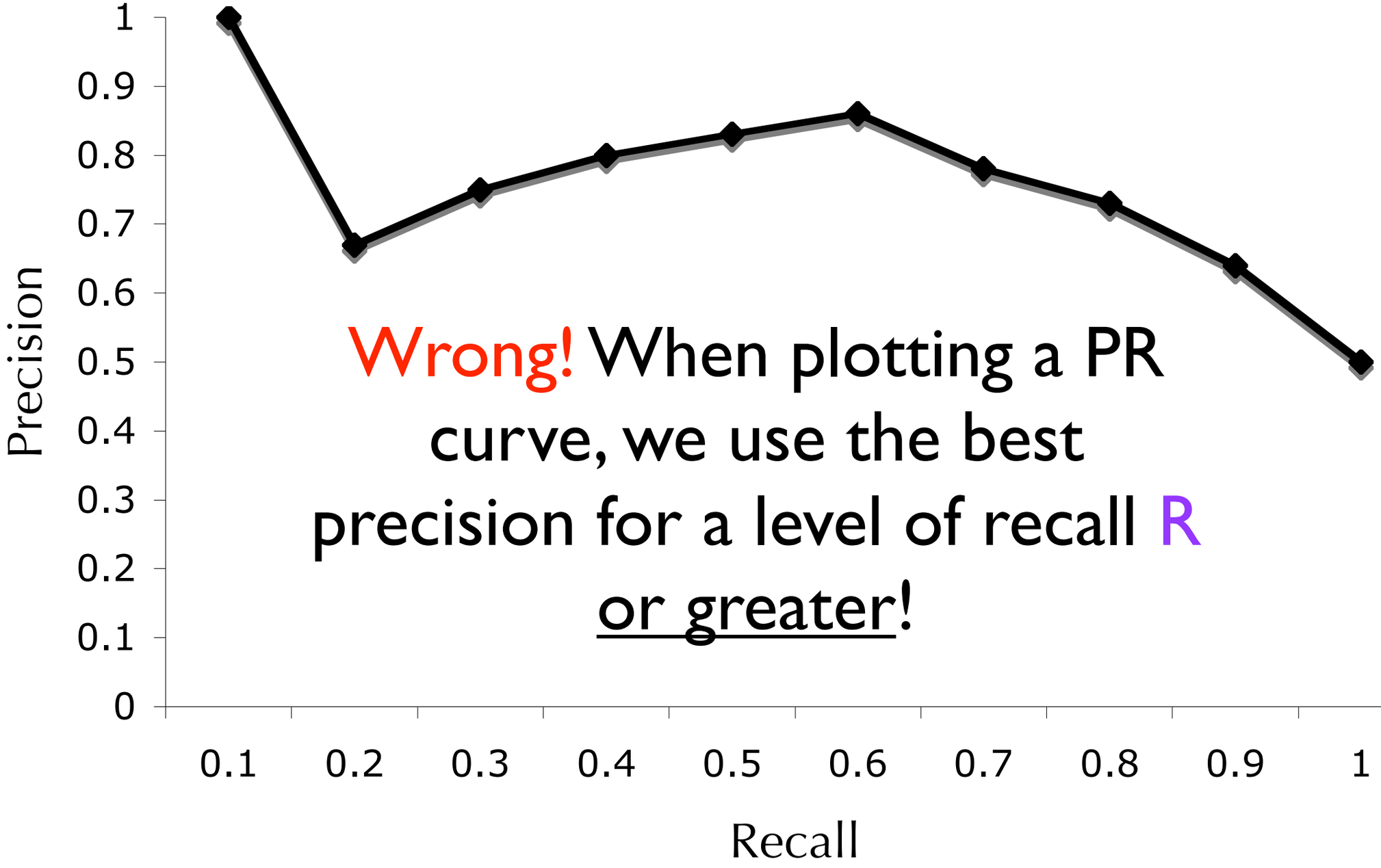
Ranked Retrieval

precision-recall curves



Ranked Retrieval

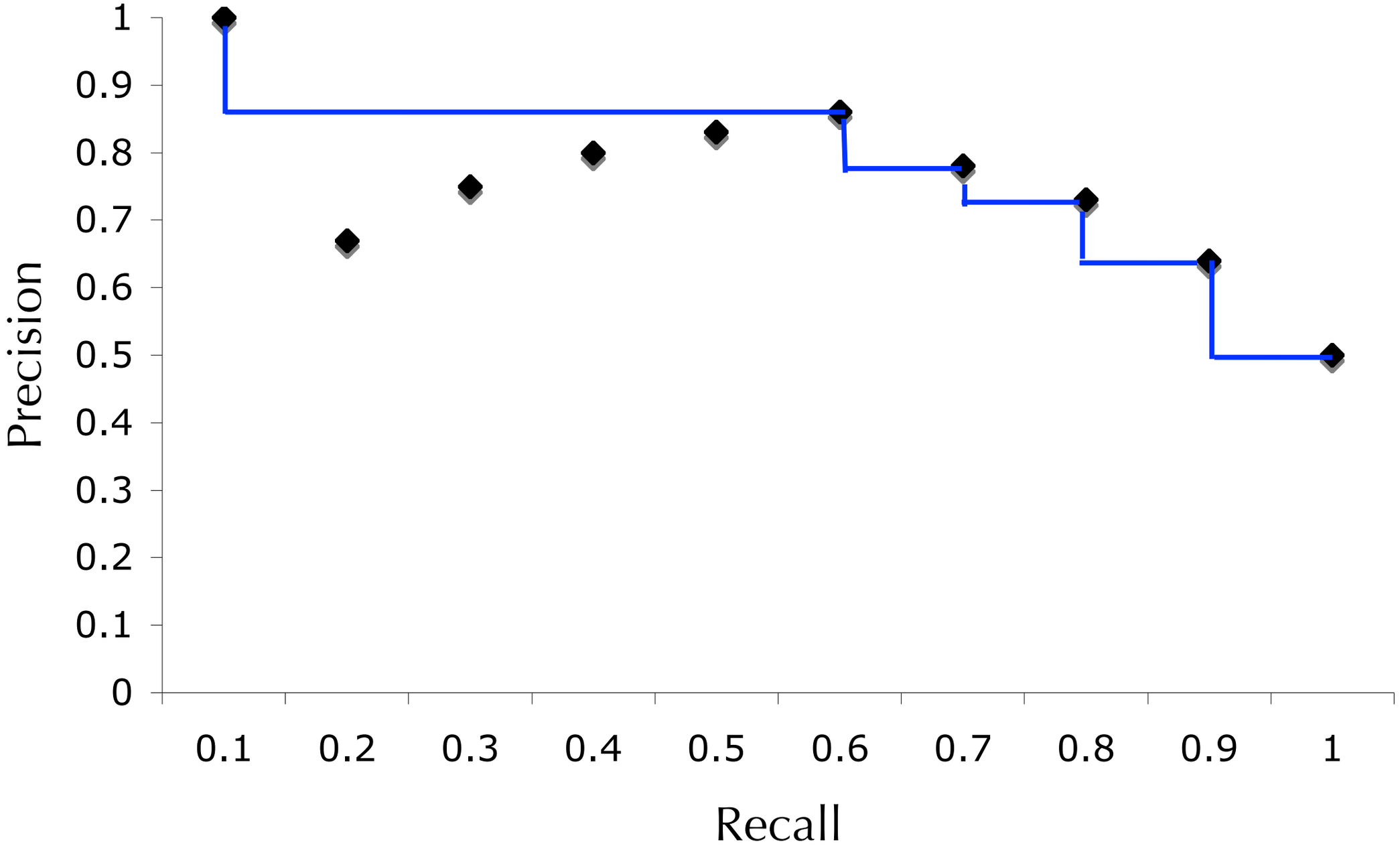
precision-recall curves



Wrong! When plotting a PR curve, we use the best precision for a level of recall R or greater!

Ranked Retrieval

precision-recall curves



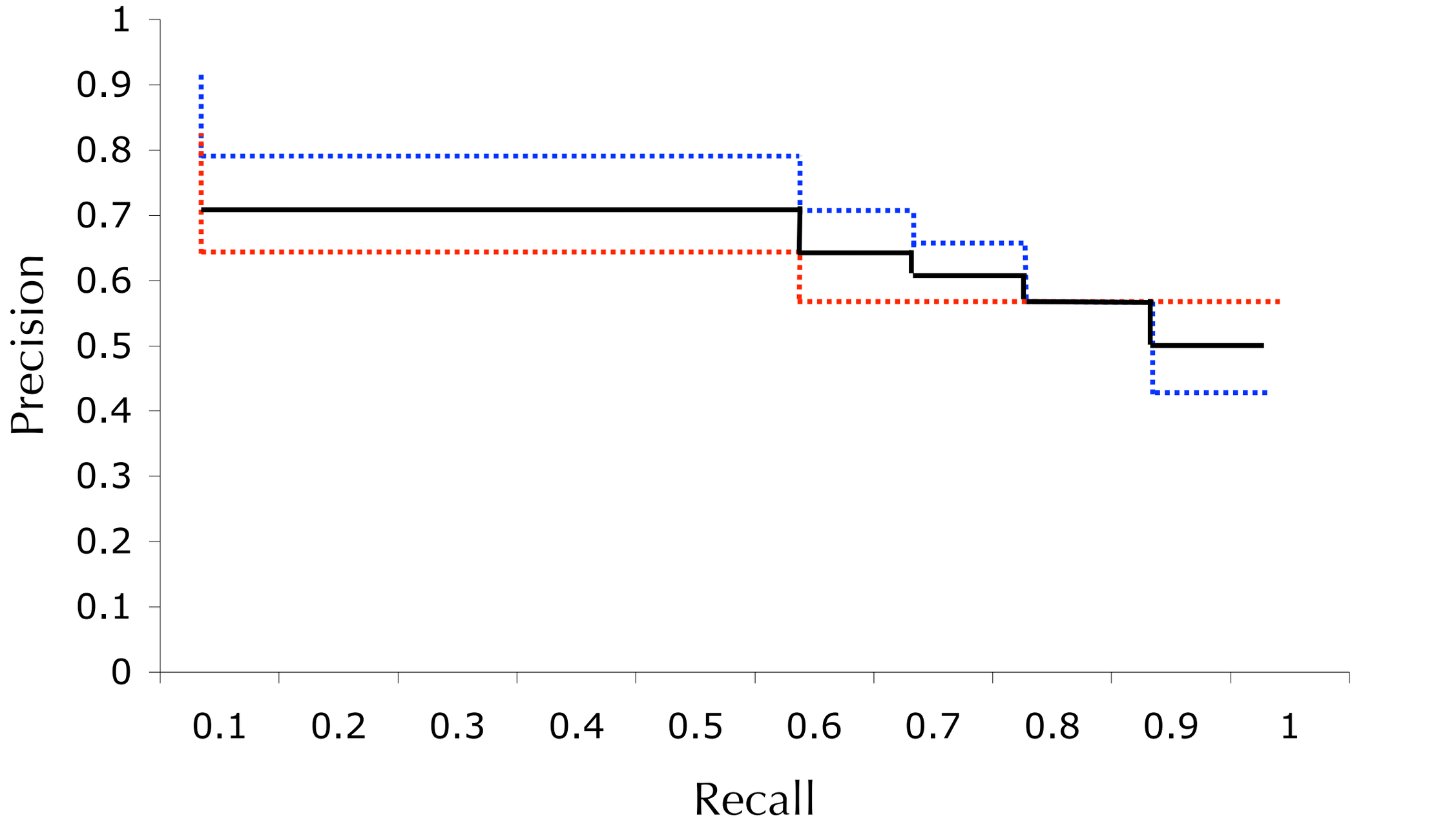
Ranked Retrieval
precision-recall curves

- For a single query, a PR curve looks like a step-function
- For multiple queries, we can average these curves
  - Average the precision values for different values of recall (e.g., from 0.01 to 1.0 in increments of 0.01)
- This forms a smoother function

Ranked Retrieval

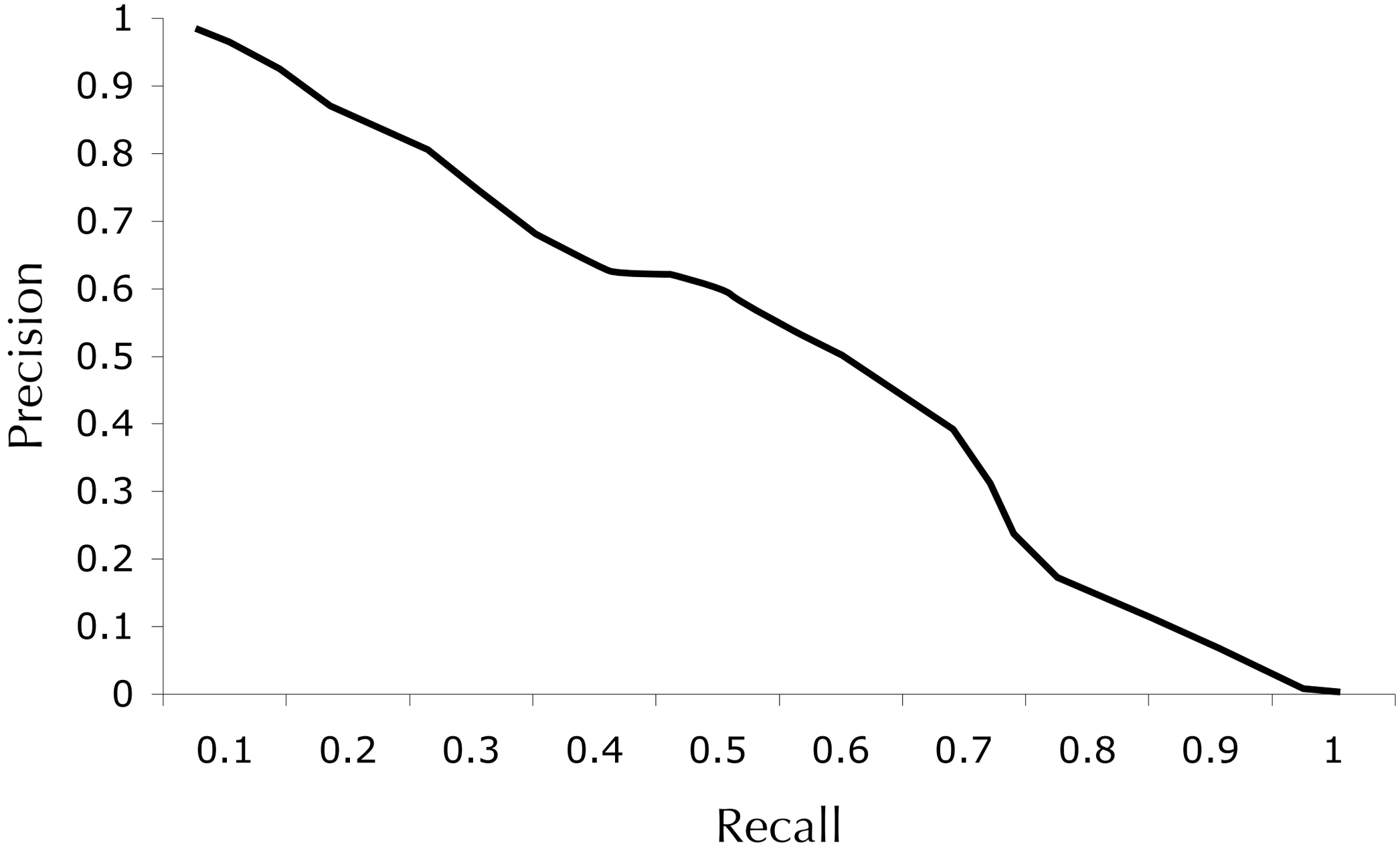
precision-recall curves

- PR curves can be averaged across multiple queries



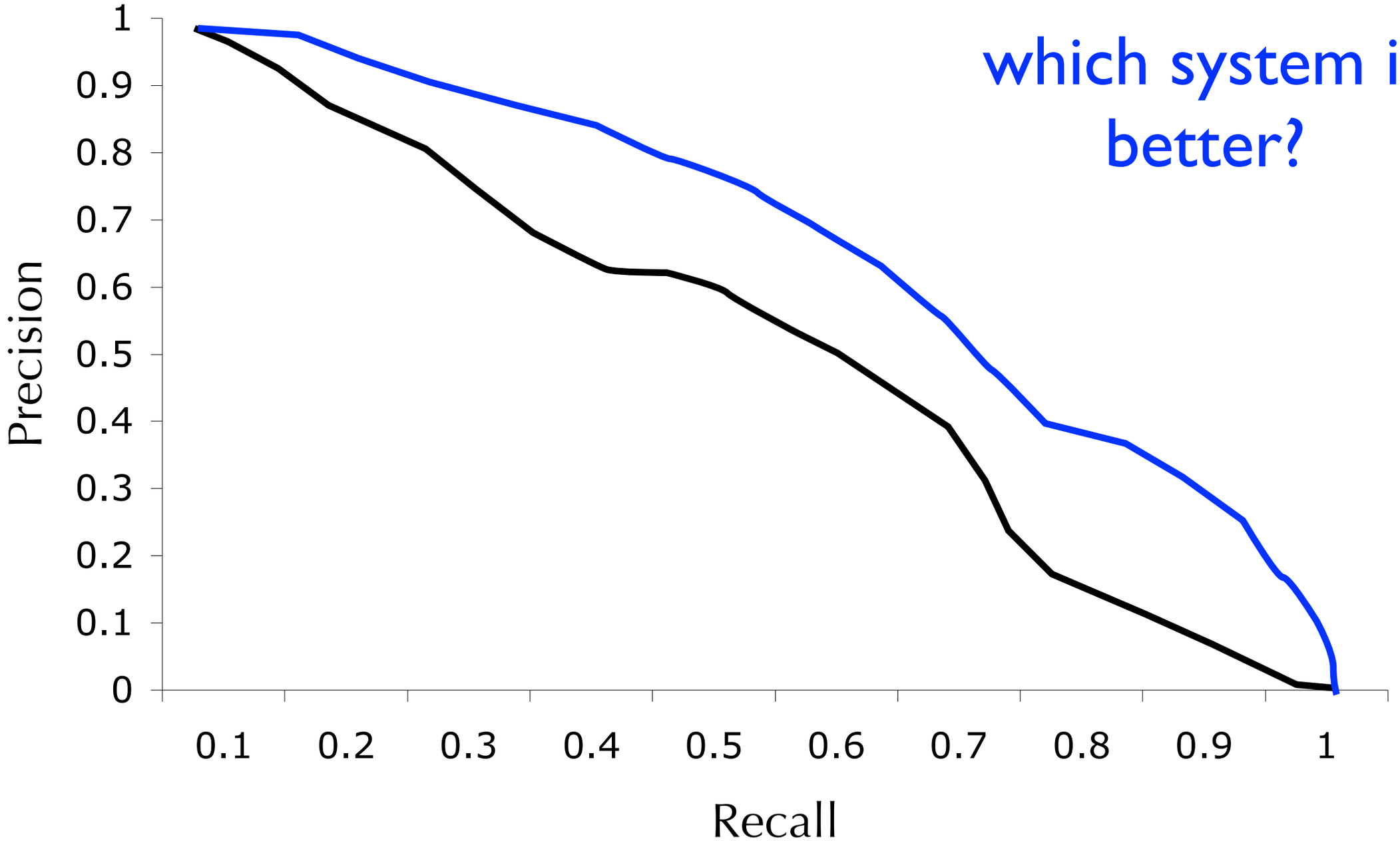
Ranked Retrieval

precision-recall curves



Ranked Retrieval

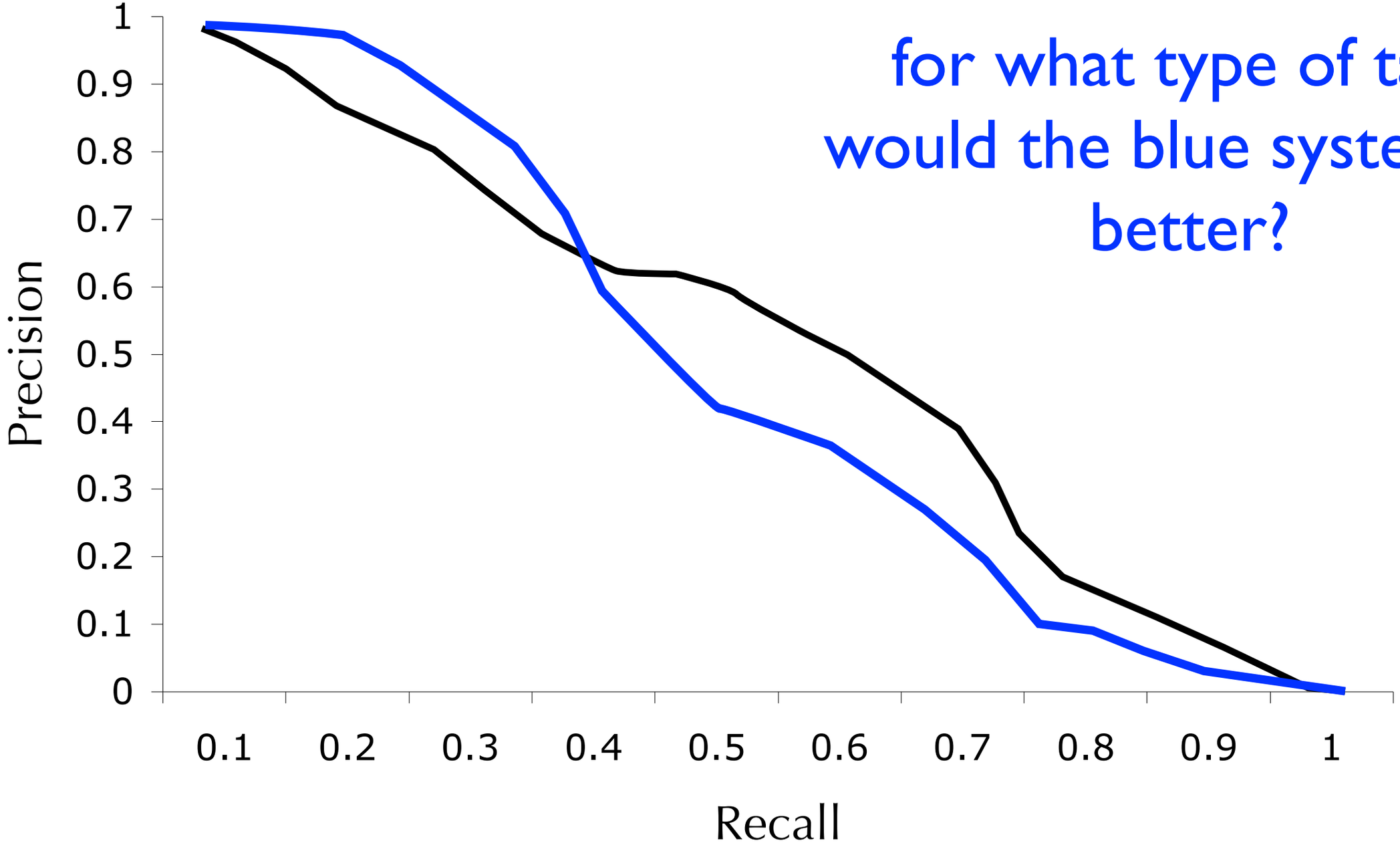
precision-recall curves



which system is better?

Ranked Retrieval

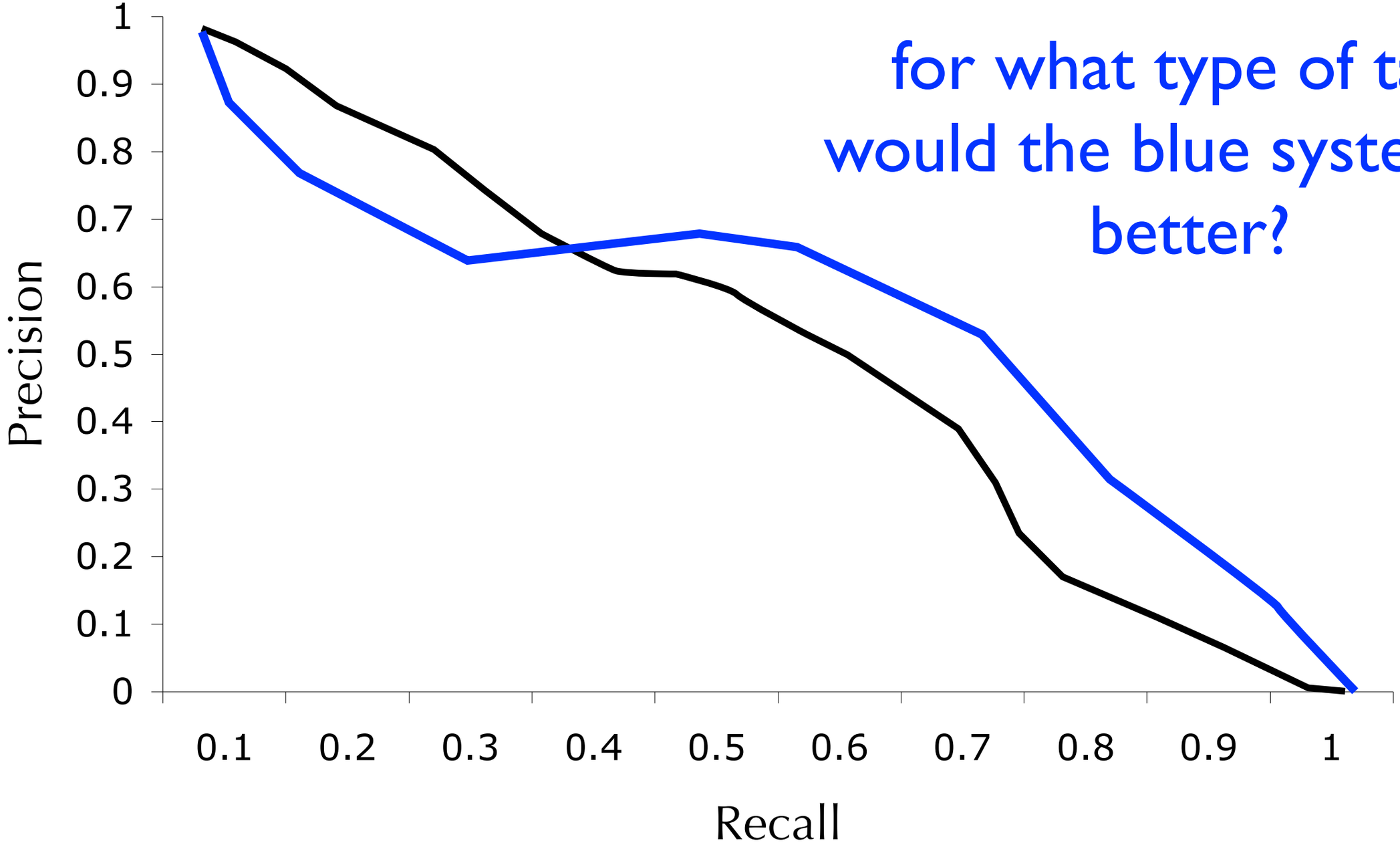
precision-recall curves



for what type of task would the blue system be better?

Ranked Retrieval

precision-recall curves



for what type of task would the blue system be better?

Metric Review
set-retrieval evaluation

- Precision: the proportion of retrieved documents that are relevant
- Recall: the proportion of relevant documents that are retrieved

Metric Review
ranked-retrieval evaluation

- P@K: precision under the assumption that the top-K results is the 'set' retrieved
- R@K: recall under the assumption that the top-K results is the 'set' retrieved
- Average-precision: considers precision and recall and focuses on the top results
- Precision-recall curves: describes precision for different levels of recall

Which Metric Would You Use?

bing
twitter
f
okcupid
e Harmony
PANDORA
Google
Y!
match.com
mapquest mQ
ICON FINDER
YAHOO! ANSWERS
flickr
Picasa
acm
Linked in
Westlaw
The New York Times
JSTOR
yelp
You Tube
Broadcast Yourself™
LexisNexis

Outline

Information Retrieval

Search Engine Components

Document Representation

Retrieval Models

Evaluation

Federated Search and Cross-lingual IR

Open-source Toolkits

Federated Search

Up to this point...

- Classic information retrieval
  - search from a single centralized index
  - all queries processed the same way

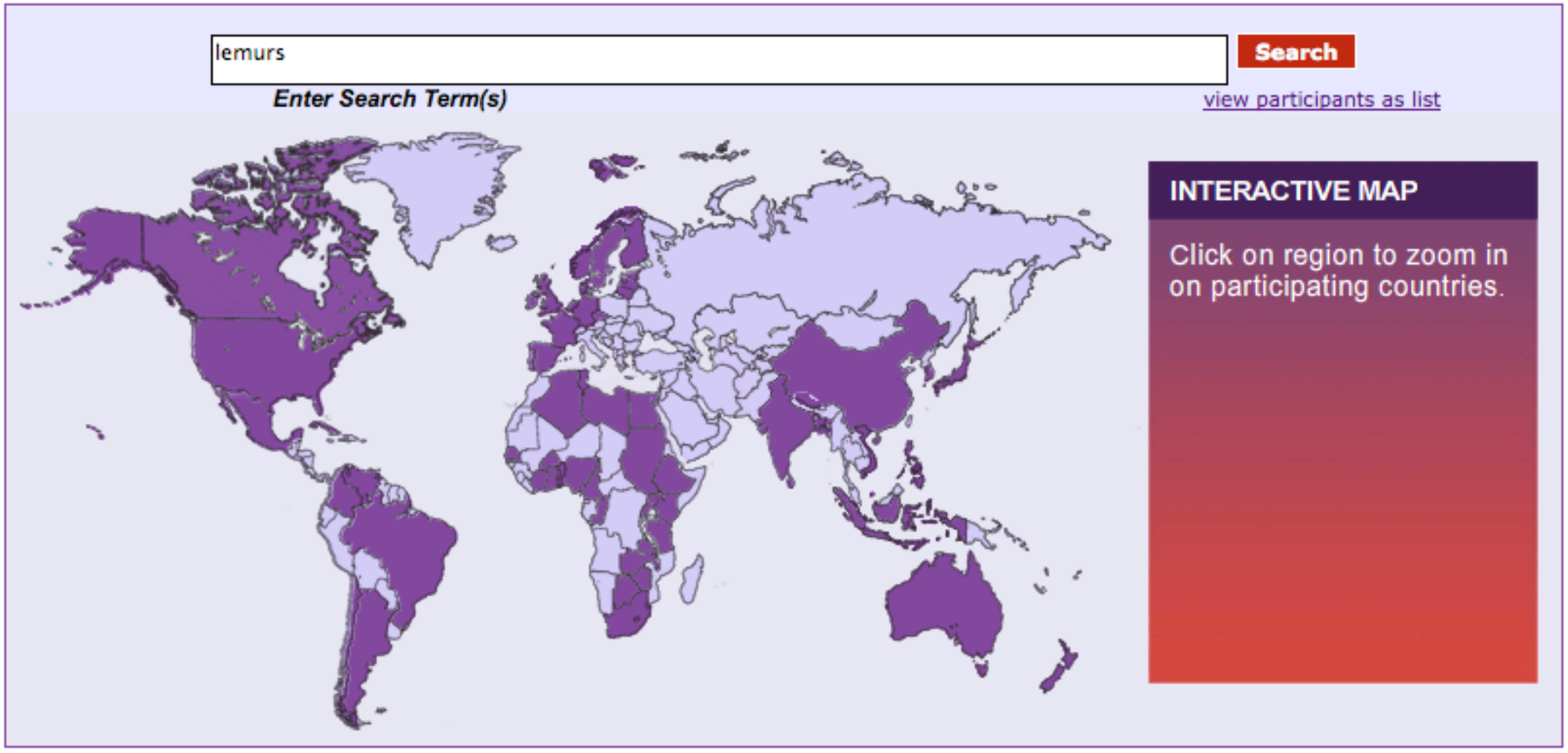
- Federated search
  - search across multiple distributed collections
  - a.k.a: resources, search engines, search services, etc.

Motivation

- Some content cannot be crawled and centrally indexed (exposed only via a search interface)
  - also referred to as “the hidden web”
- Even if crawl-able, we may prefer searchable access to this content via the third-party search engine. why?
  - content updated locally
  - unique document representation (e.g., metadata)
  - customized retrieval

Federated Search Examples
(World Wide Science)

- Exhaustive search (across <u>all</u> collections)



Federated Search Examples
(World Wide Science)

Refine Search | lemurs | New Search | Advanced Search | by Deep Web TECHNOLOGIES

Search: **Full Record: lemurs**
301 ranked results of 1,625 available

Create an alert from this search | Summary of All Results

62 of **62** sources complete

Results **1** – **10** of 301 Sort by: Rank Limit to: All Sources 1 2 3 4 5

1 **Lemurs - Ambassadors for Madagascar**
★★★★☆ *Thalmann, Urs*
Madagascar Conservation & Development 2006-01-01
Directory of Open Access Journals (Sweden)

2 **The dental comb of lemurs**
★★★★☆ *Roberts, D.*
UK PubMed Central
Digital Repository Infrastructure Vision for European Research (DRIVER)

3 **The Placentation of Lemurs**
★★★★☆ *Turner*
UK PubMed Central
Digital Repository Infrastructure Vision for European Research (DRIVER)

4 **Object permanence in lemurs.**
★★★★☆ *Deppe, Anja M.*
Vascoda (Germany)

Federated Search Examples

(World Wide Science)

Refine Search | lemurs | New Search | Advanced Search

Search: **Full Record: lemurs**
301 ranked results of 1,625 available

Create an alert from this search | Summary of All Results

Results **1** – **10** of 301 Sort by: Rank Limit to: All Sources 1 2 3

1 **Lemurs - Ambassadors for Madagascar**
★★★★☆ *Thalmann, Urs*
Madagascar Conservation & Development 2006-01-01
Directory of Open Access Journals (Sweden)

2 **The dental comb of lemurs**
★★★★☆ *Roberts, D.*
UK PubMed Central
Digital Repository Infrastructure Vision for European Research (DRIVER)

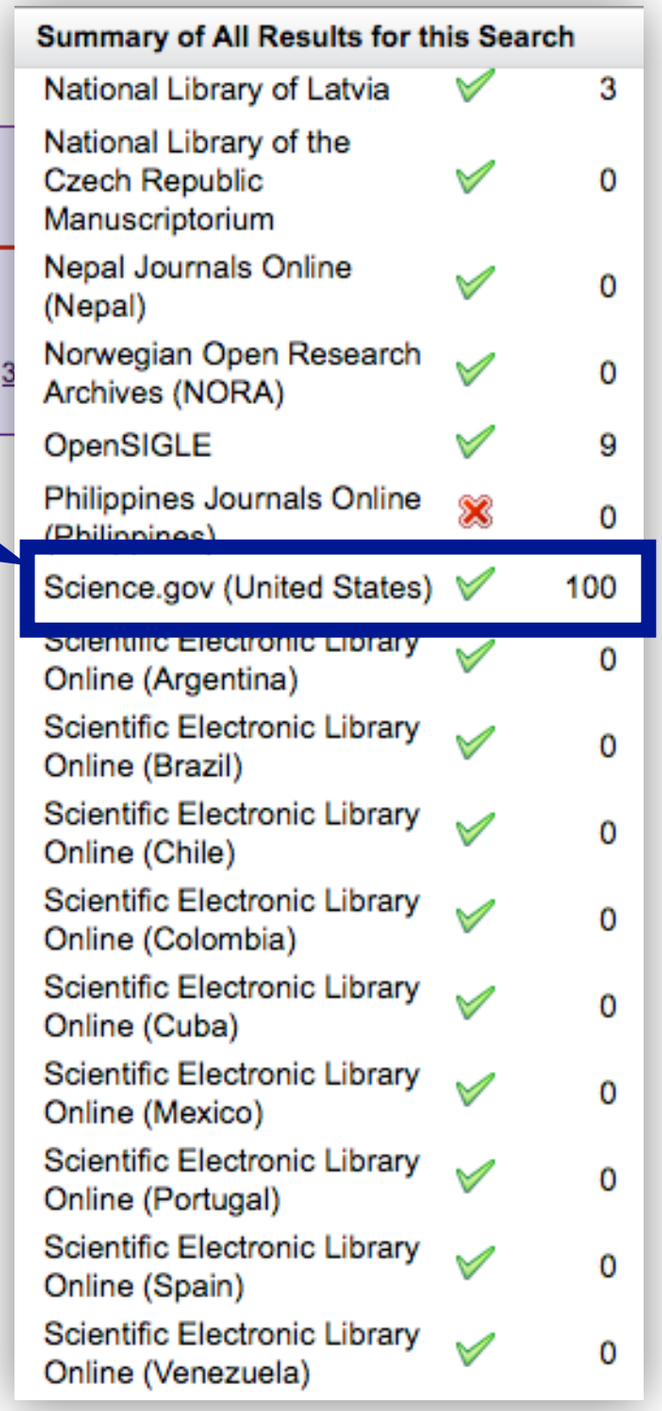
3 **The Placentation of Lemurs**
★★★★☆ *Turner*
UK PubMed Central
Digital Repository Infrastructure Vision for European Research (DRIVER)

4 **Object permanence in lemurs.**
★★★★☆ *Deppe, Anja M.*
Vascoda (Germany)

Summary of All Results for this Search

National Library of Latvia	✓	3
National Library of the Czech Republic Manuscriptorium	✓	0
Nepal Journals Online (Nepal)	✓	0
Norwegian Open Research Archives (NORA)	✓	0
OpenSIGLE	✓	9
Philippines Journals Online (Philippines)	✗	0
Science.gov (United States)	✓	100
Scientific Electronic Library Online (Argentina)	✓	0
Scientific Electronic Library Online (Brazil)	✓	0
Scientific Electronic Library Online (Chile)	✓	0
Scientific Electronic Library Online (Colombia)	✓	0
Scientific Electronic Library Online (Cuba)	✓	0
Scientific Electronic Library Online (Mexico)	✓	0
Scientific Electronic Library Online (Portugal)	✓	0
Scientific Electronic Library Online (Spain)	✓	0
Scientific Electronic Library Online (Venezuela)	✓	0

Federated Search Examples
(World Wide Science)



Federated Search Examples

(Vertical Aggregation in Web Search)

maps

web

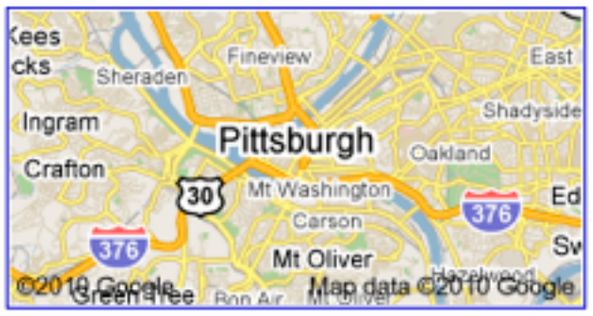
images

web

books

pittsburgh Search

Pittsburgh, PA maps.google.com



City of **Pittsburgh**, Pennsylvania - Pghgov.com

Official city site including information on economic development, resident information, links, tourism and contact information.

www.city.**pittsburgh**.pa.us/ - Cached - Similar

Images for **pittsburgh** - Report images

Pittsburgh - Wikipedia, the free encyclopedia

Pittsburgh is the second-largest city in the U.S. Commonwealth of Pennsylvania and the county seat of Allegheny County. Regionally, it anchors the largest ...

History of Pittsburgh - Neighborhoods - List of people from the Pittsburgh ... - 1936

en.wikipedia.org/wiki/**Pittsburgh** - Cached - Similar

Books for **pittsburgh**

Pittsburgh: a sketch of its early social life - Charles William Dahlinger - 1916 - 216 pages

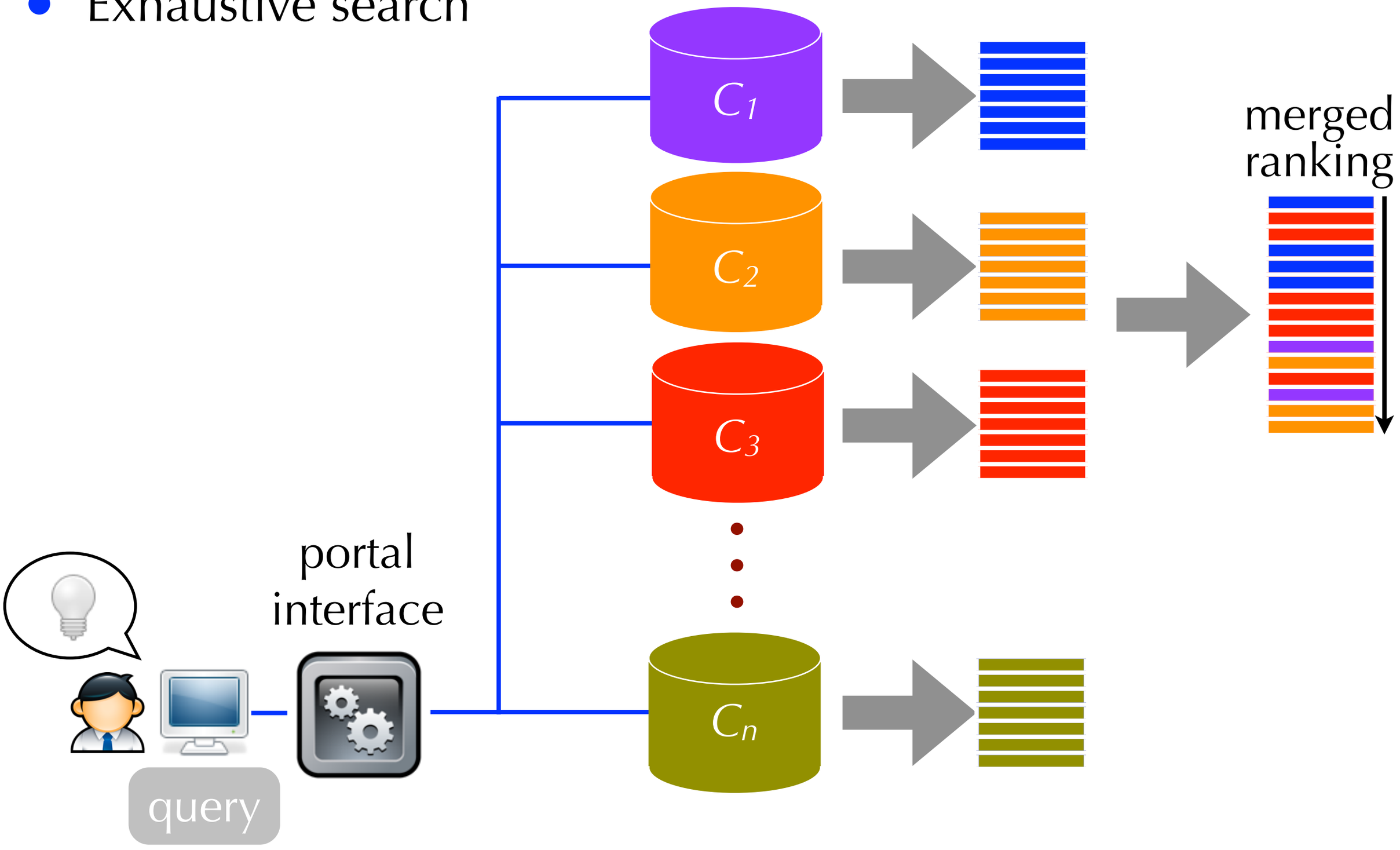
Pittsburgh:: 1758-2008 - Pittsburgh Post-Gazette, Carnegie Library of Pittsburgh - 2008 - 128 pages

Pittsburgh: 17582008 surveys the citys evolution from strategic fort in the wilderness ...

books.google.com

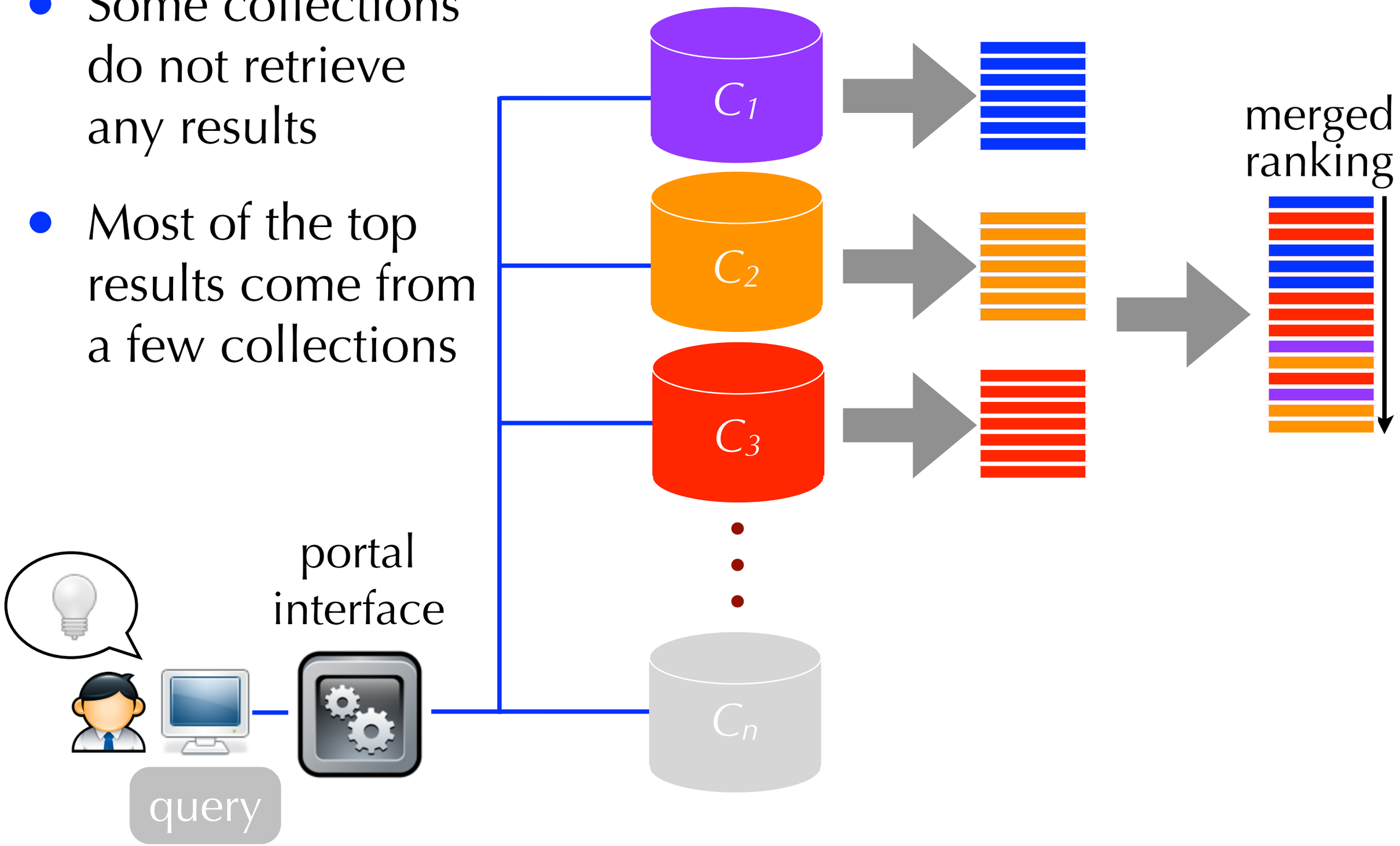
Federated Search

- Exhaustive search



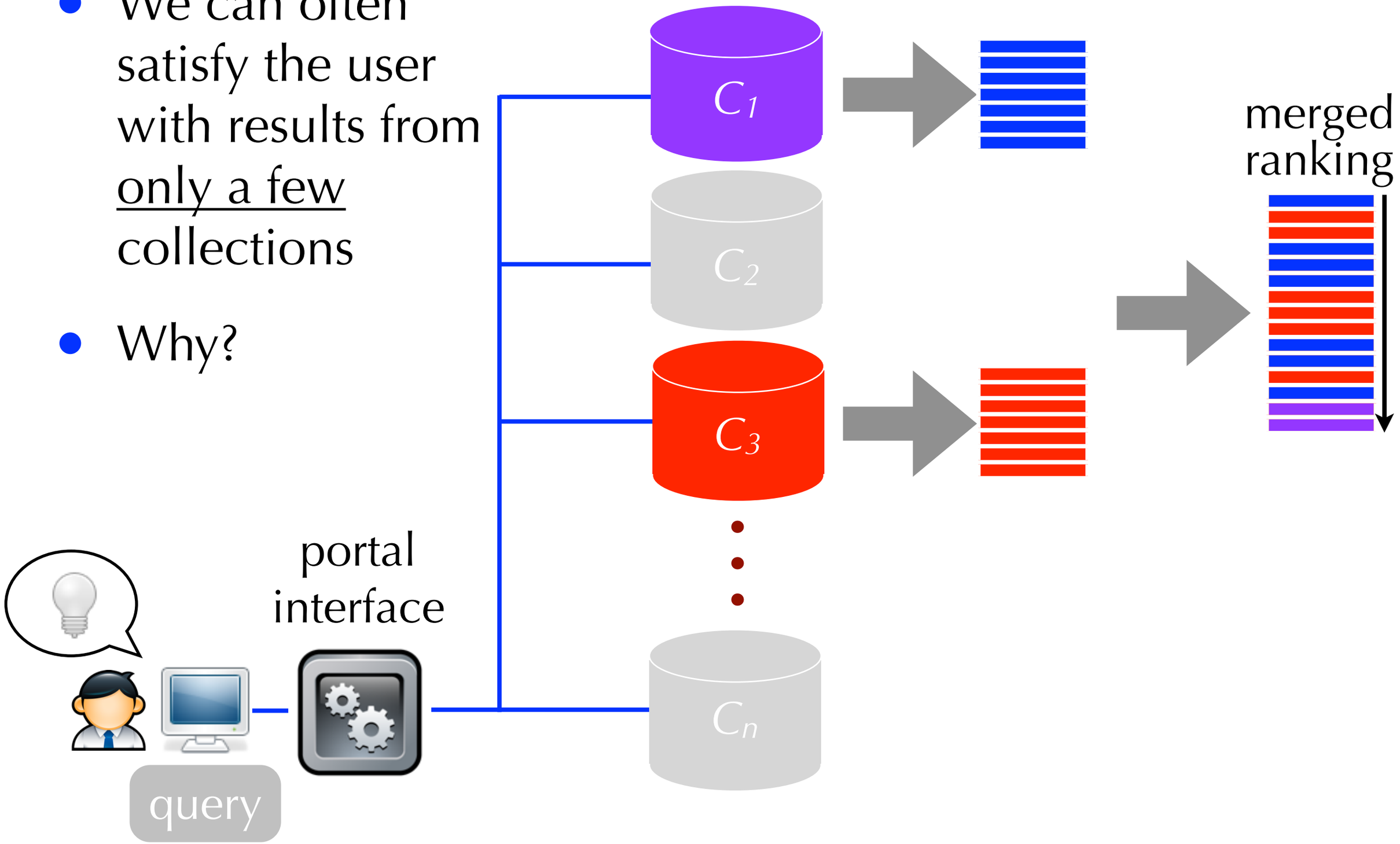
Federated Search

- Some collections do not retrieve any results
- Most of the top results come from a few collections



Federated Search

- We can often satisfy the user with results from <u>only a few</u> collections
- Why?



Federated Search

- Objective: given a query, predict which few collections have relevant documents and combine their results into a single document ranking

Federated Search

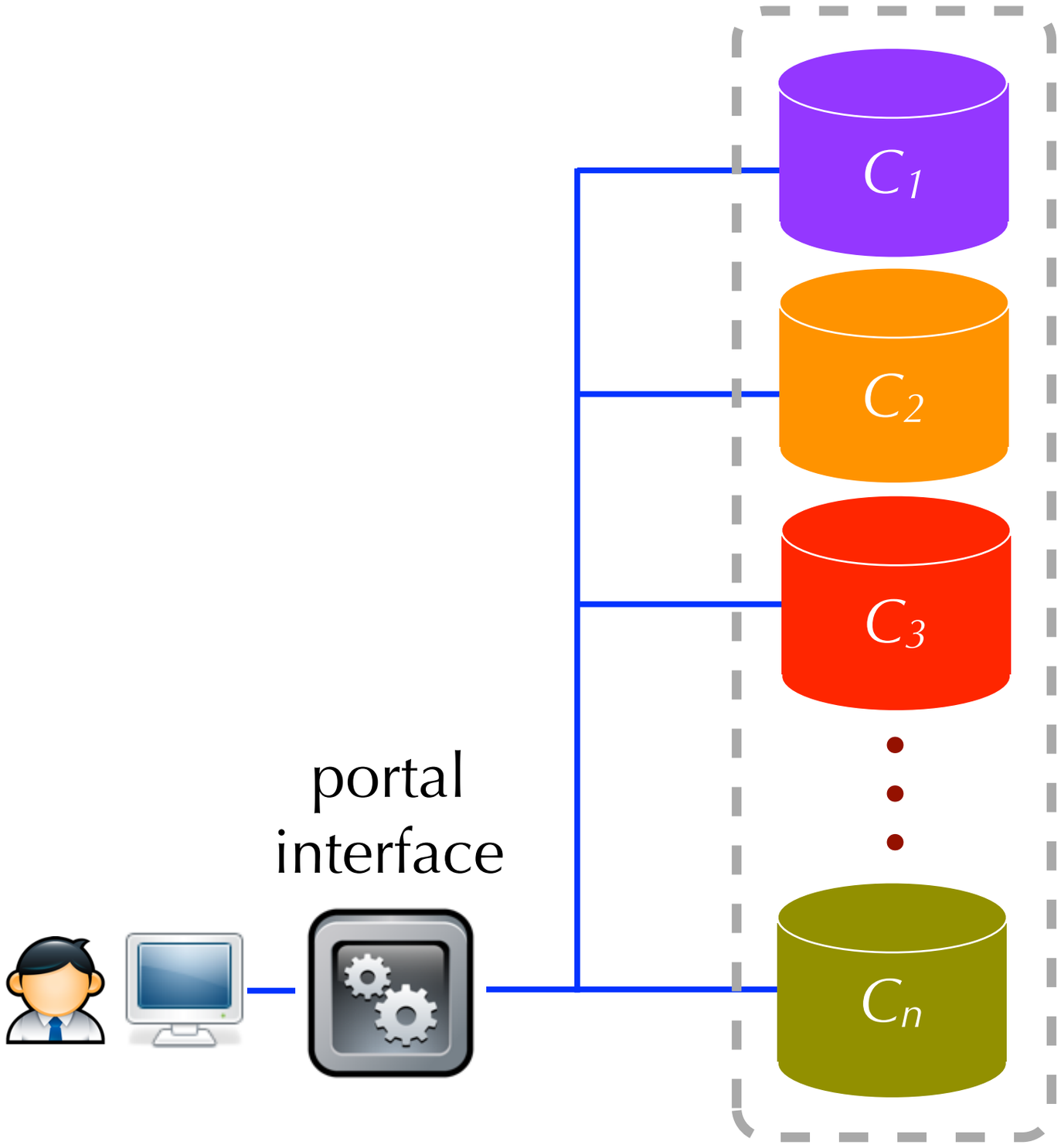
Resource representation

Resource selection

Results merging

Federated Search

Resource Representation

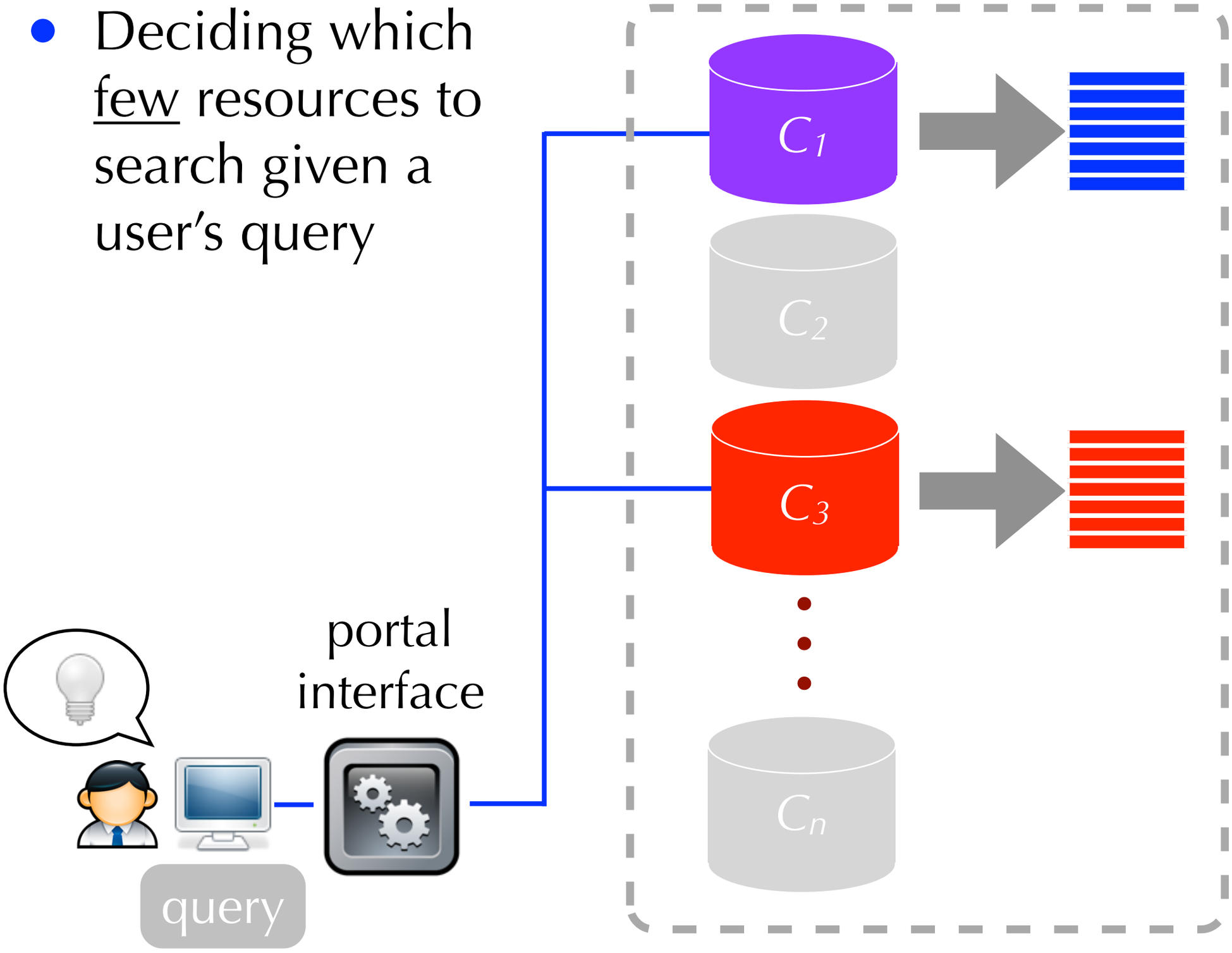


- Gathering information about what each resource contains
- What types of information needs does each resource satisfy?

Federated Search

Resource Selection

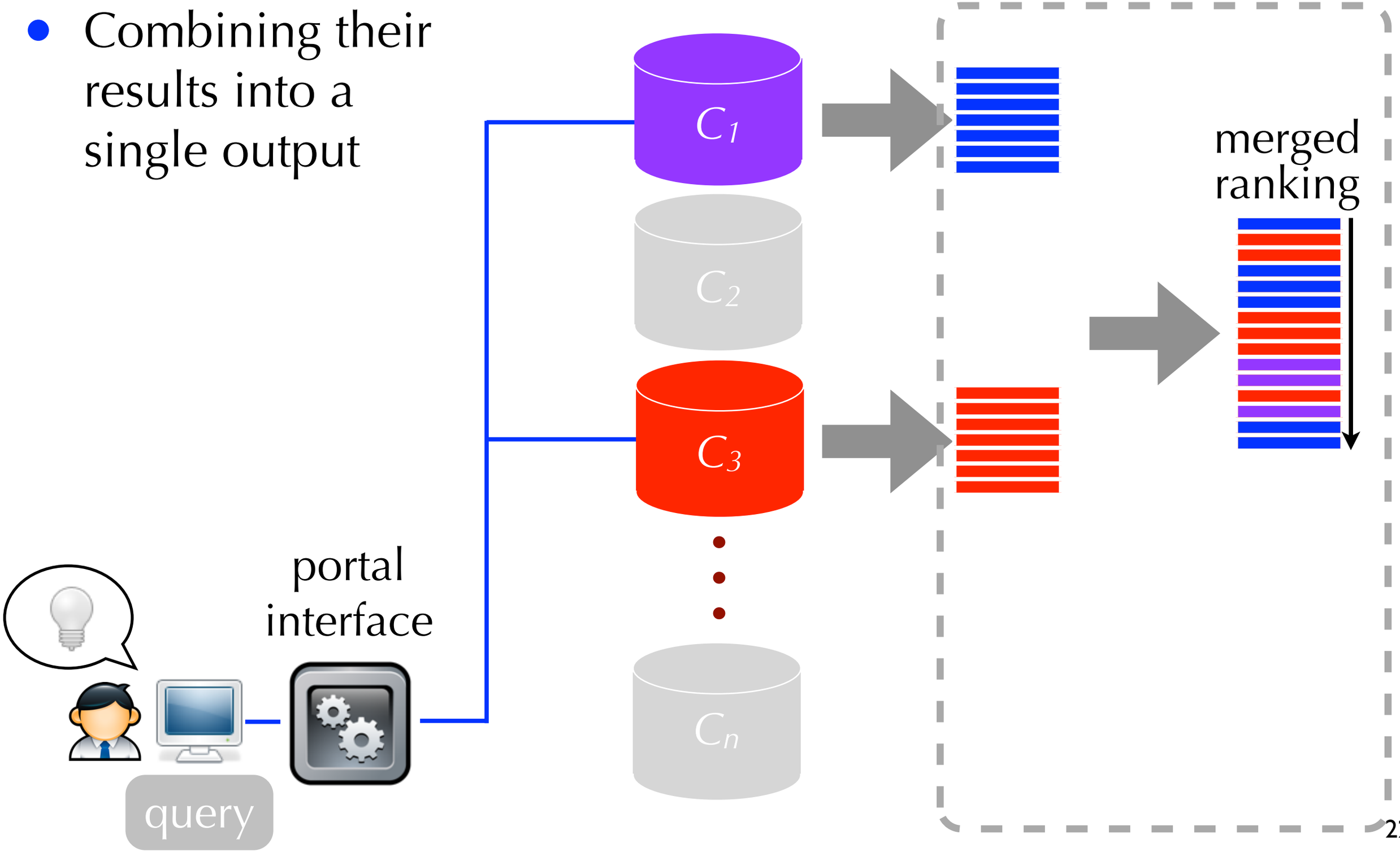
- Deciding which <u>few</u> resources to search given a user's query



Federated Search

Results Merging

- Combining their results into a single output



Federated Search

off-line

resource representation

---

resource selection

results merging

at query-time

Cooperative vs. Uncooperative

- Cooperative environment
  - assumption: resources provide accurate and complete information to facilitate selection and merging
  - centrally designed protocols and APIs
- Uncooperative environment
  - assumption: resources provide no special support for federated search
  - only a search interface
- Different environments require different solutions

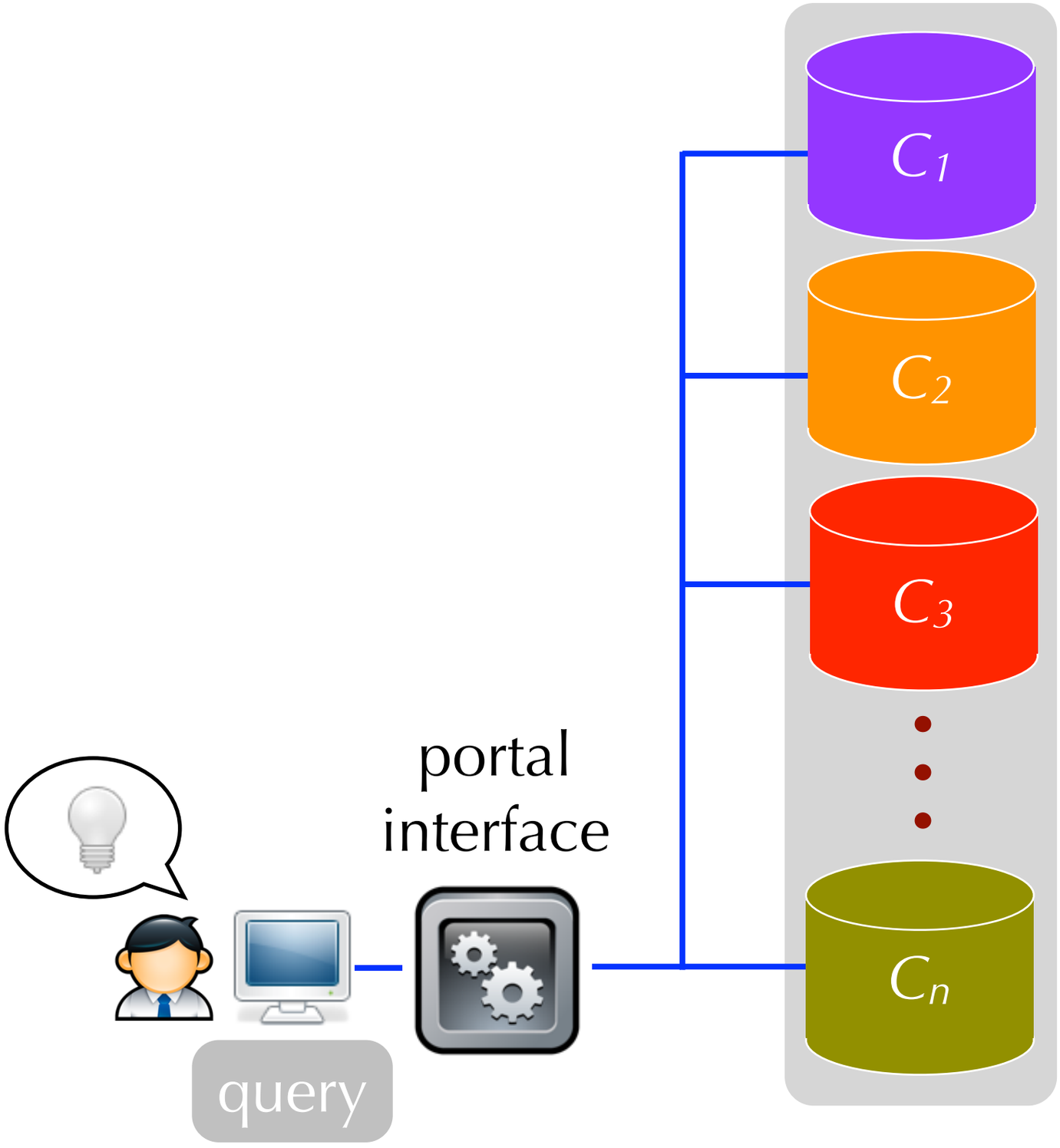
Resource Representation

Resource Representation

- Objective: to gather information about what each resource contains
  - but, ultimately to inform resource selection
- Discussion: what sources of evidence could we use to do this?

Resource Representation

using content

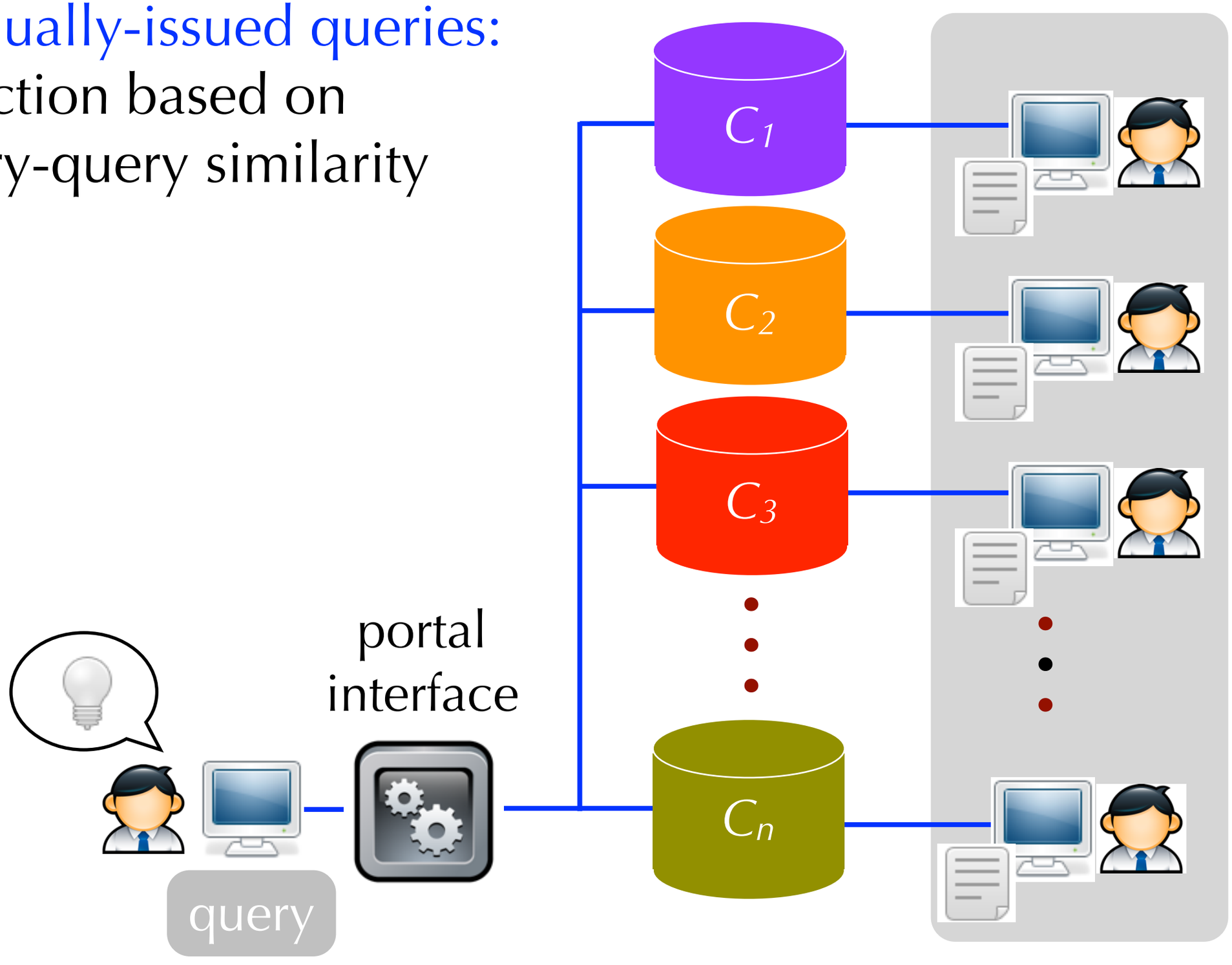


- Term frequencies: selection based on the query-collection similarity

- A set of "typical" docs: selection based on the predicted relevance of sampled documents

Resource Representation
using manually-issued queries

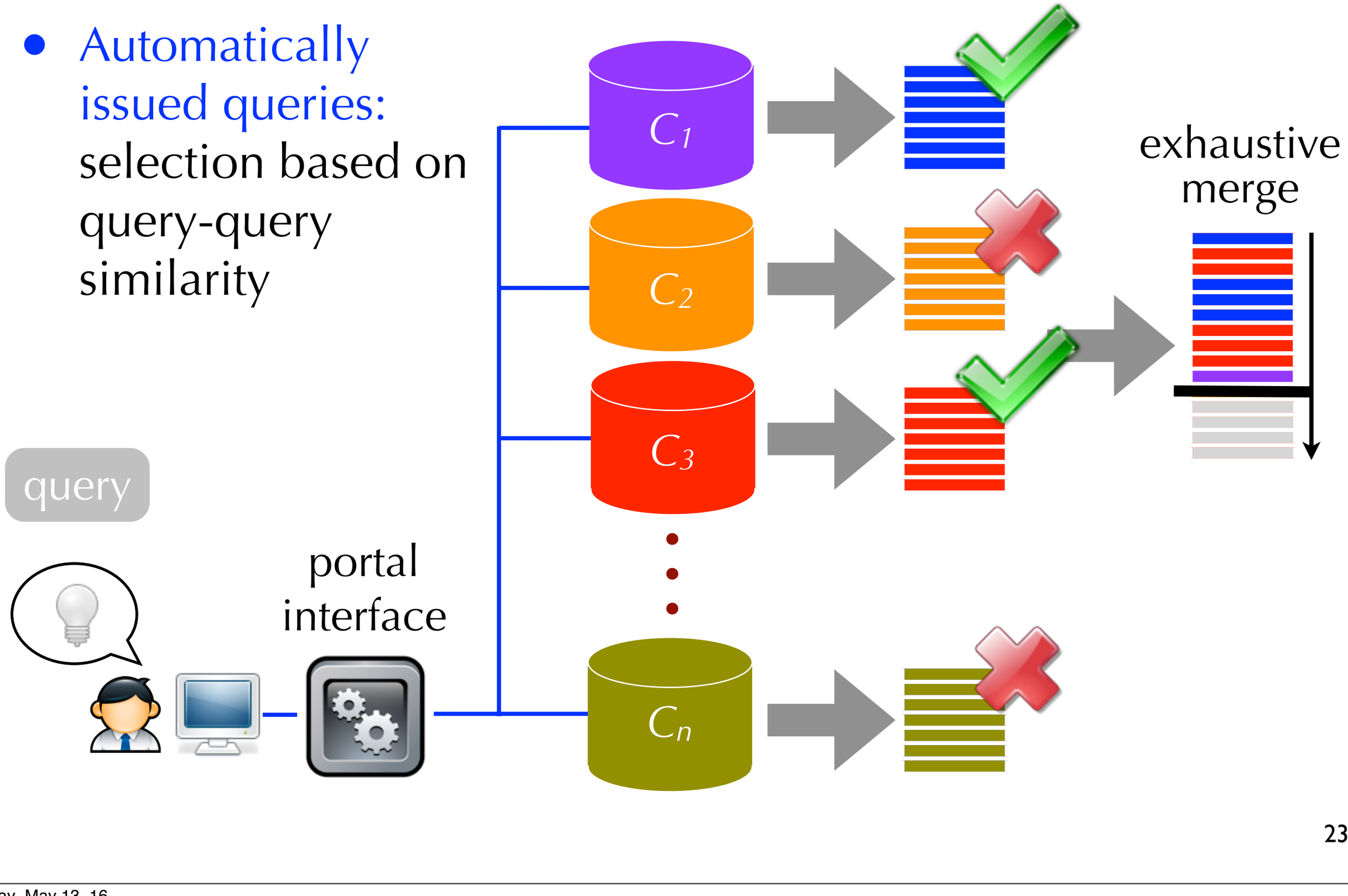
- Manually-issued queries: selection based on query-query similarity



Resource Representation

using previous retrievals

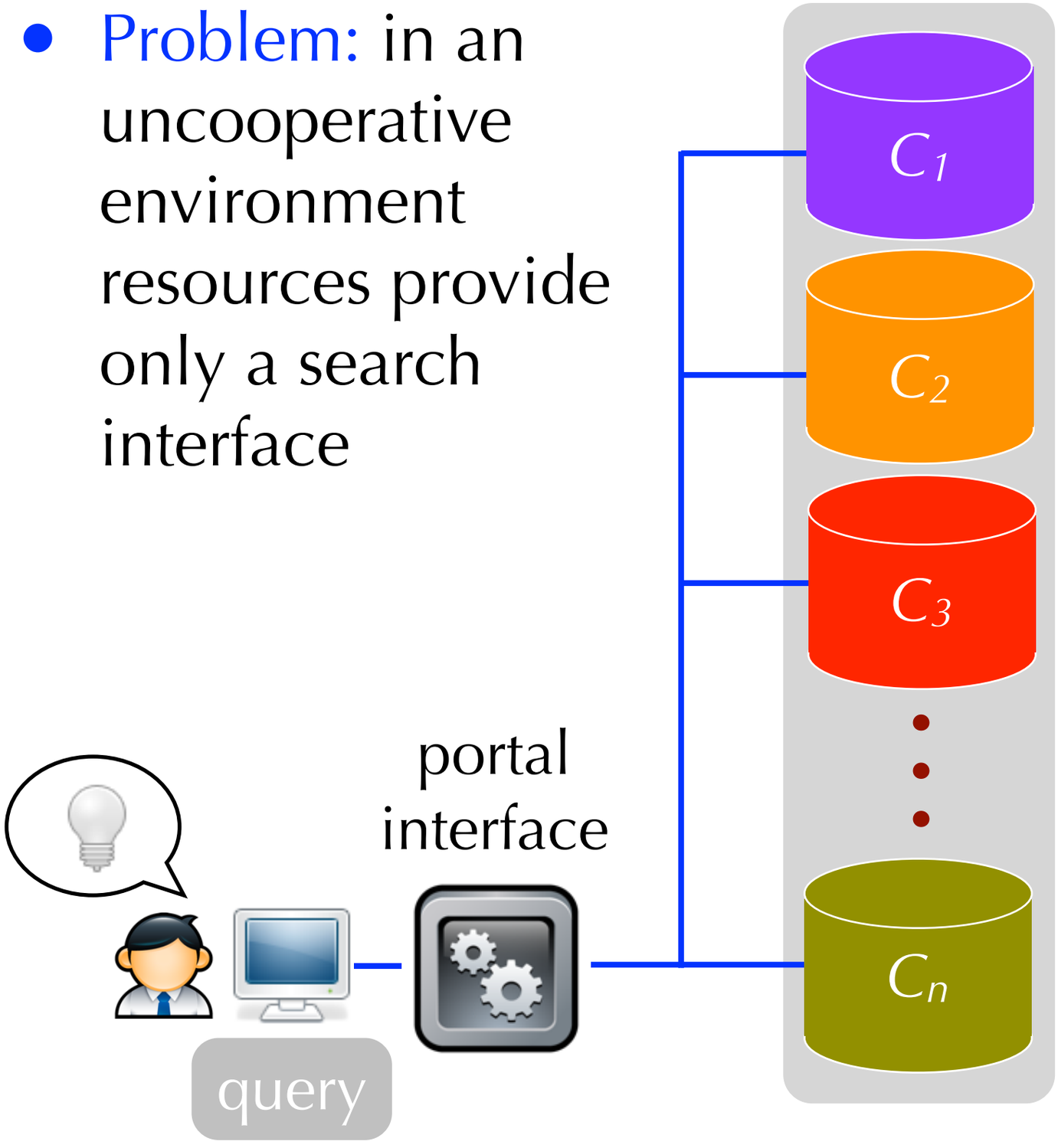
- Automatically issued queries: selection based on query-query similarity



Resource Representation

using content

- Problem: in an uncooperative environment resources provide only a search interface



- Term frequencies: selection based on the query-collection similarity
- A set of 'typical' docs: selection based on the predicted relevance of sampled documents

Query-based Sampling
(Callan and Connell, 2001)

- Repeat *N* times (e.g., *N*=100),
  1. submit a query to the search engine
  2. download a few results (e.g., 4)
  3. update the collection representation (e.g., term frequencies)
  4. select a new query for sampling (e.g., from the emerging representation)

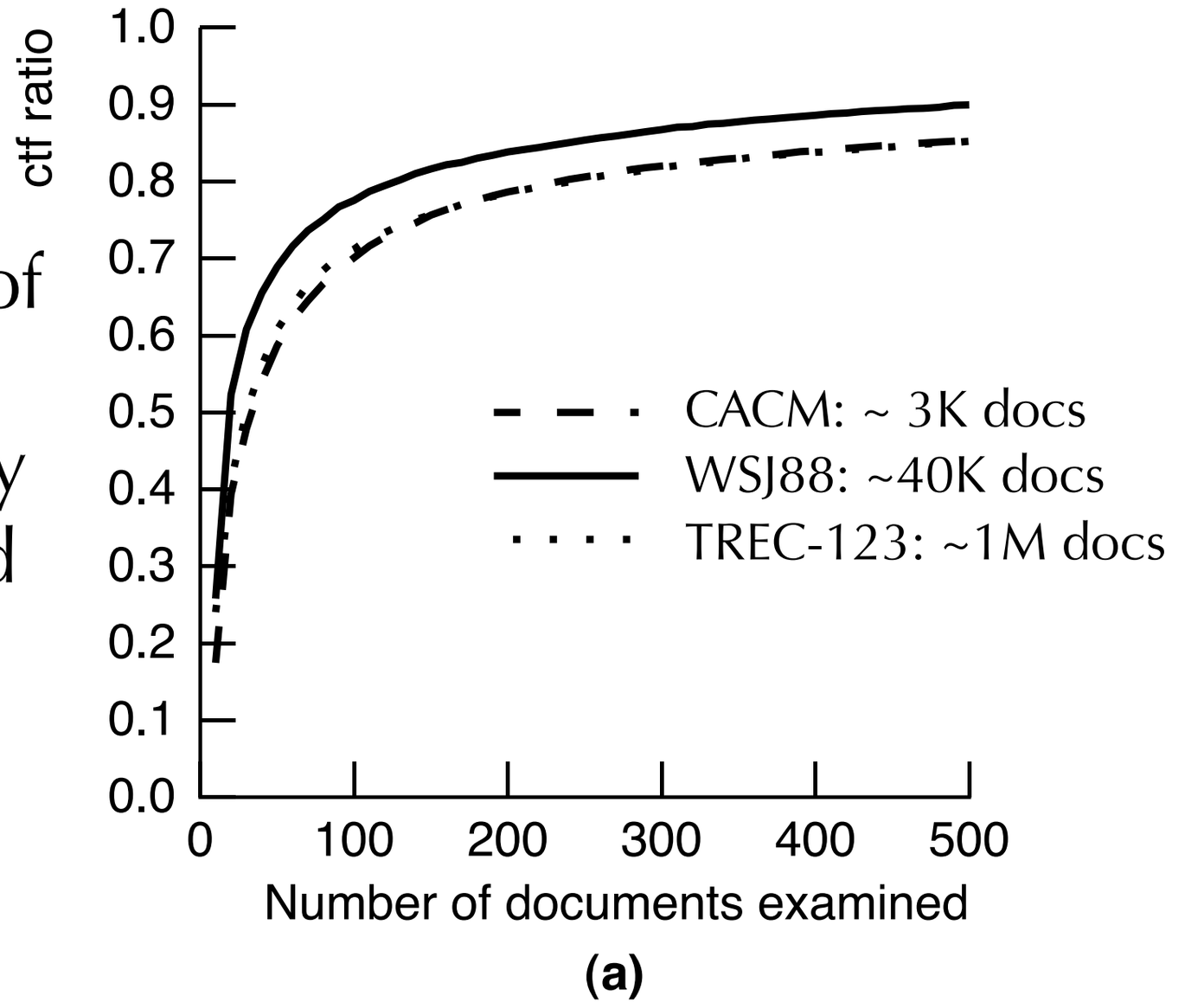
Query-based Sampling

- Discussion: suppose we want to represent resources using term frequency information, how many samples do we need?

- Hint: zipf's law states that the number of new terms seen in each additional document decreases exponentially

Query-based Sampling
(Callan and Connell, 2001)

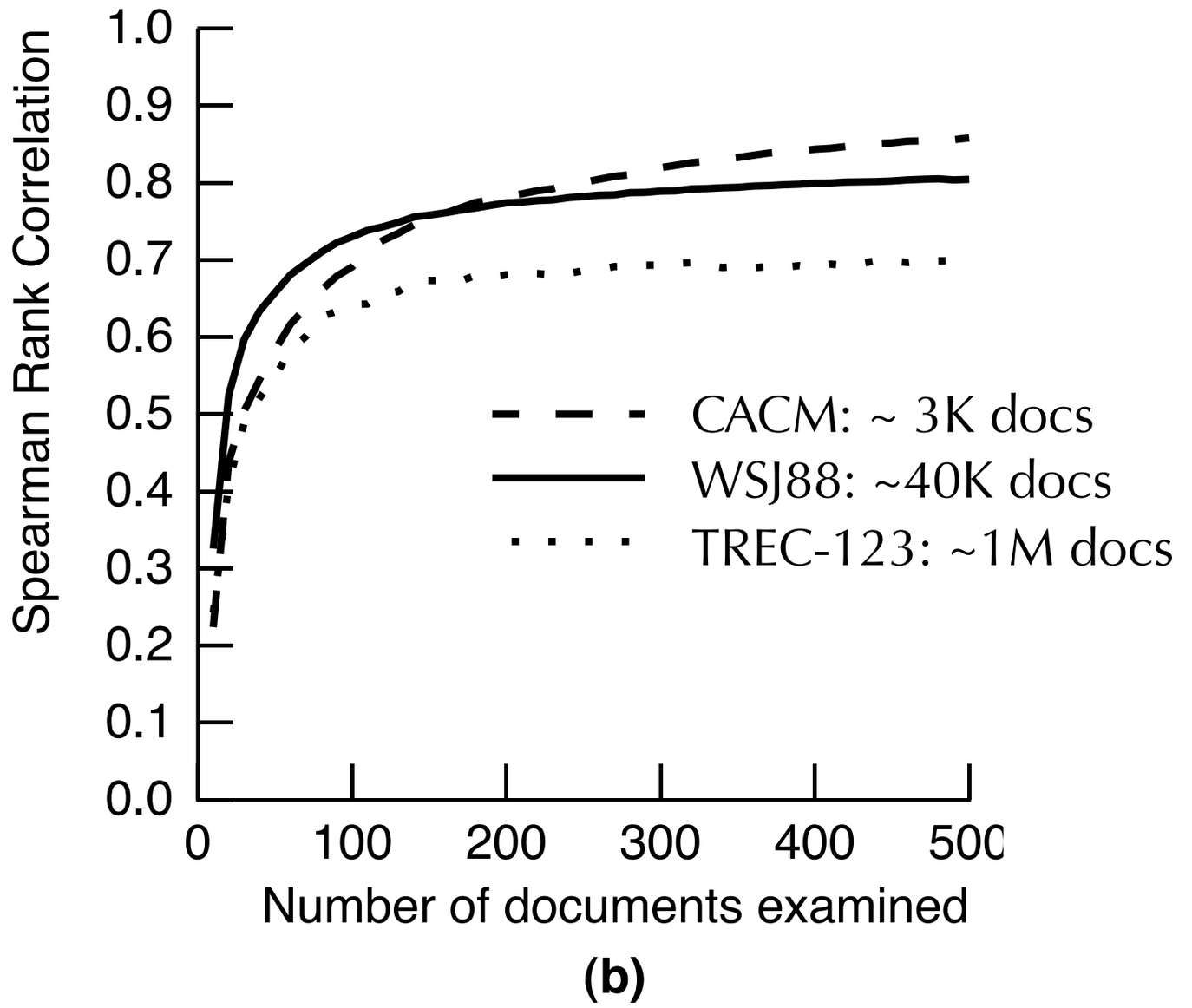
ctf ratio: % of collection "covered" by the observed terms



(a)

- After 500 docs we've seen enough vocabulary to account for about 80-90% all term occurrences

Query-based Sampling
(Callan and Connell, 2001)



- The ordering of terms (by frequency) based on sample set statistics approximates the actual one

Resource Selection

Resource Selection

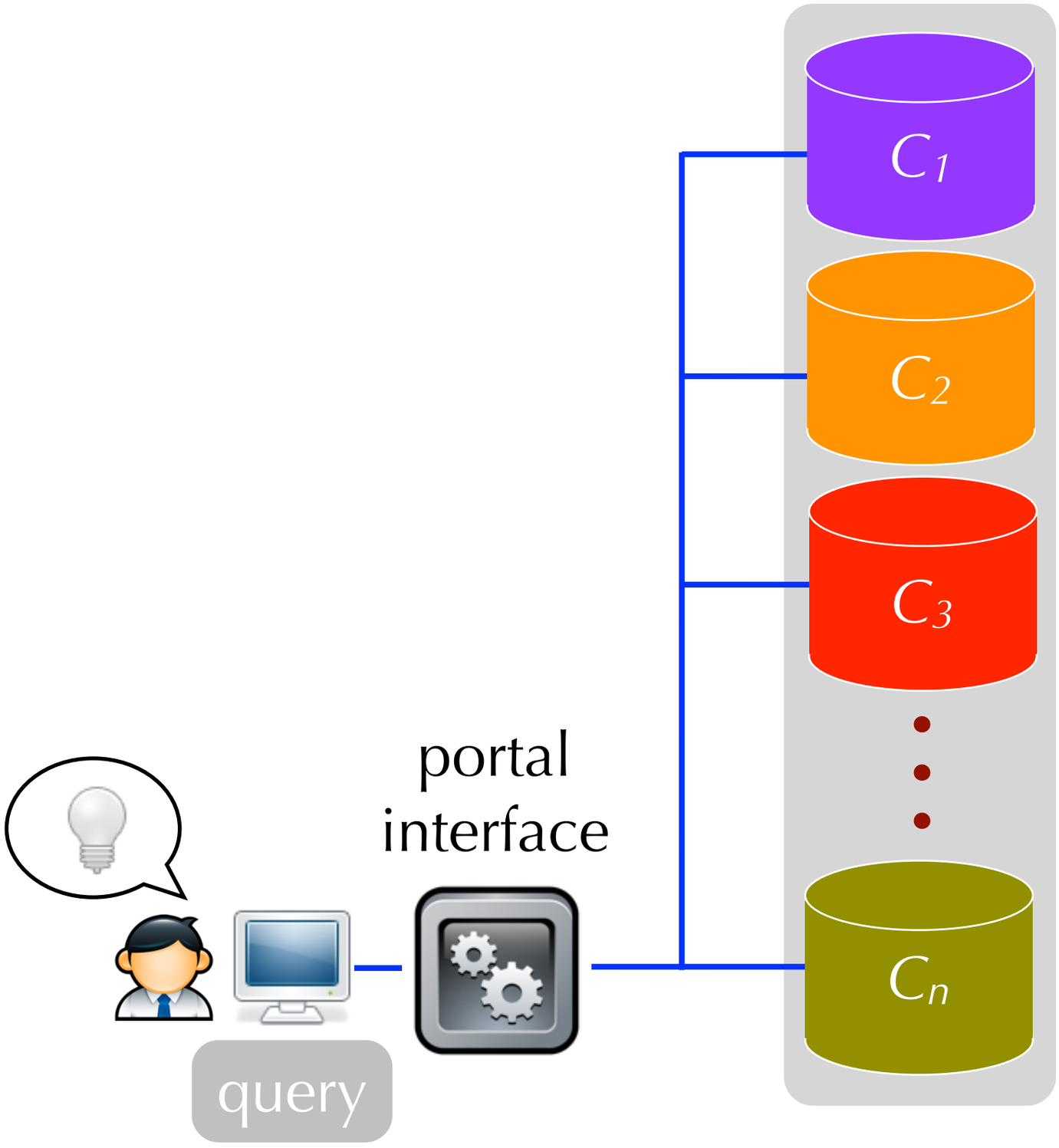
- Objective: deciding which resources to search given a user's query
- Most prior work casts the problem as resource ranking
  - given a query, select the $k \ll n$ collections that produce good merged results
  - k is given (an interesting research problem)

Resource Selection

- Content-based methods: score resources based on the similarity between the query and content from the resource
  - large vs. small document models
- Query-similarity methods: score resources based on the effectiveness of previously issued queries that are similar to the query (will be covered at high level)

Resource Representation

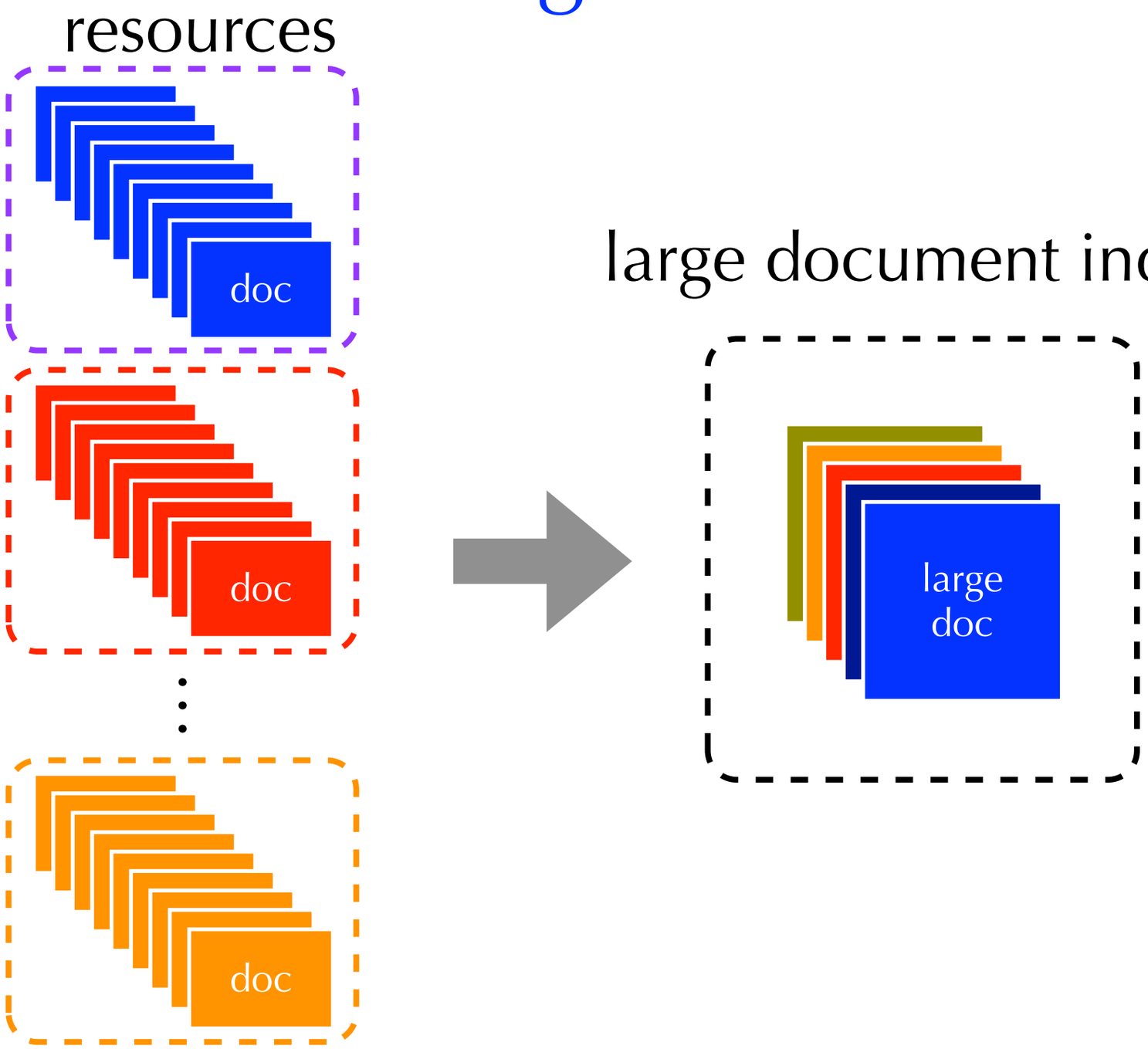
using content



- Term frequencies: selection based on the query-collection similarity

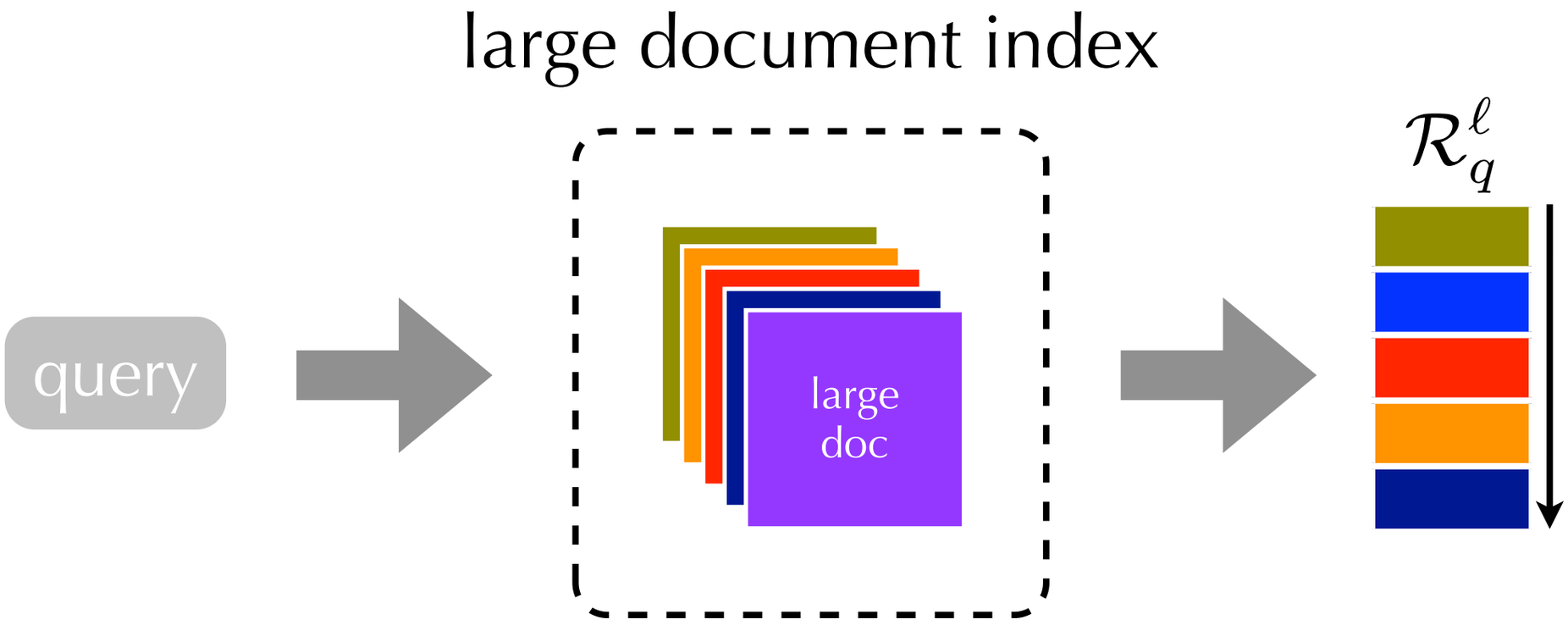
- A set of 'typical' docs: selection based on the predicted relevance of sampled documents

Large Document Models



- Represent each resource (or its samples) as a single "large document"

Large Document Models



1. Given the query, rank "large documents" using functions adapted from document retrieval
2. Select the top *k*

Large Document Models

- CORI (Callan, 1995)

$$\text{CORI}_w(C_i) = b + (1-b) \times \frac{df_{w,i}}{df_{w,i} + 50 + 150 \times \frac{col\_len}{avg\_col\_len}} \times \frac{\log\left(\frac{|\mathcal{C}|+0.5}{cf_w}\right)}{\log(|\mathcal{C}|+1.0)}$$

- adapted from BM25

$$P(w|d) = b + (1-b) \times \frac{tf}{tf + 0.5 + 1.5 \times \frac{doc\_len}{avg\_doc\_len}} \times \frac{\log\left(\frac{N+0.5}{df}\right)}{\log(N+1.0)}$$

Large Document Models

- KL-Divergence (Xu and Croft 1999)

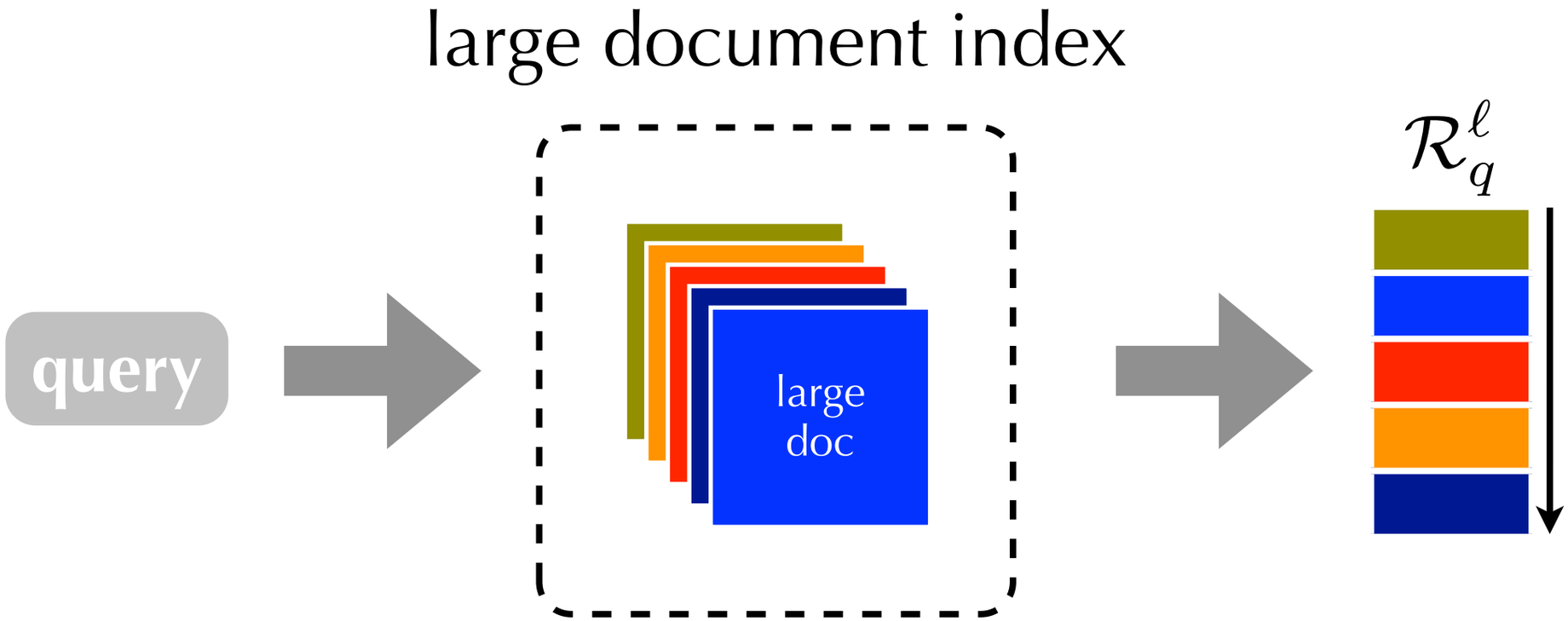
$$KL_q(C_i) = \sum_{w \in q} P(w|q) \log \left(\frac{P(w|q)}{P(w|C_i)} \right)$$

- Query Likelihood (Si *et al.*, 2002)

$$P(q|C_i) = \prod_{w \in q} \lambda P(w|C_i) + (1 - \lambda) P(w|G)$$

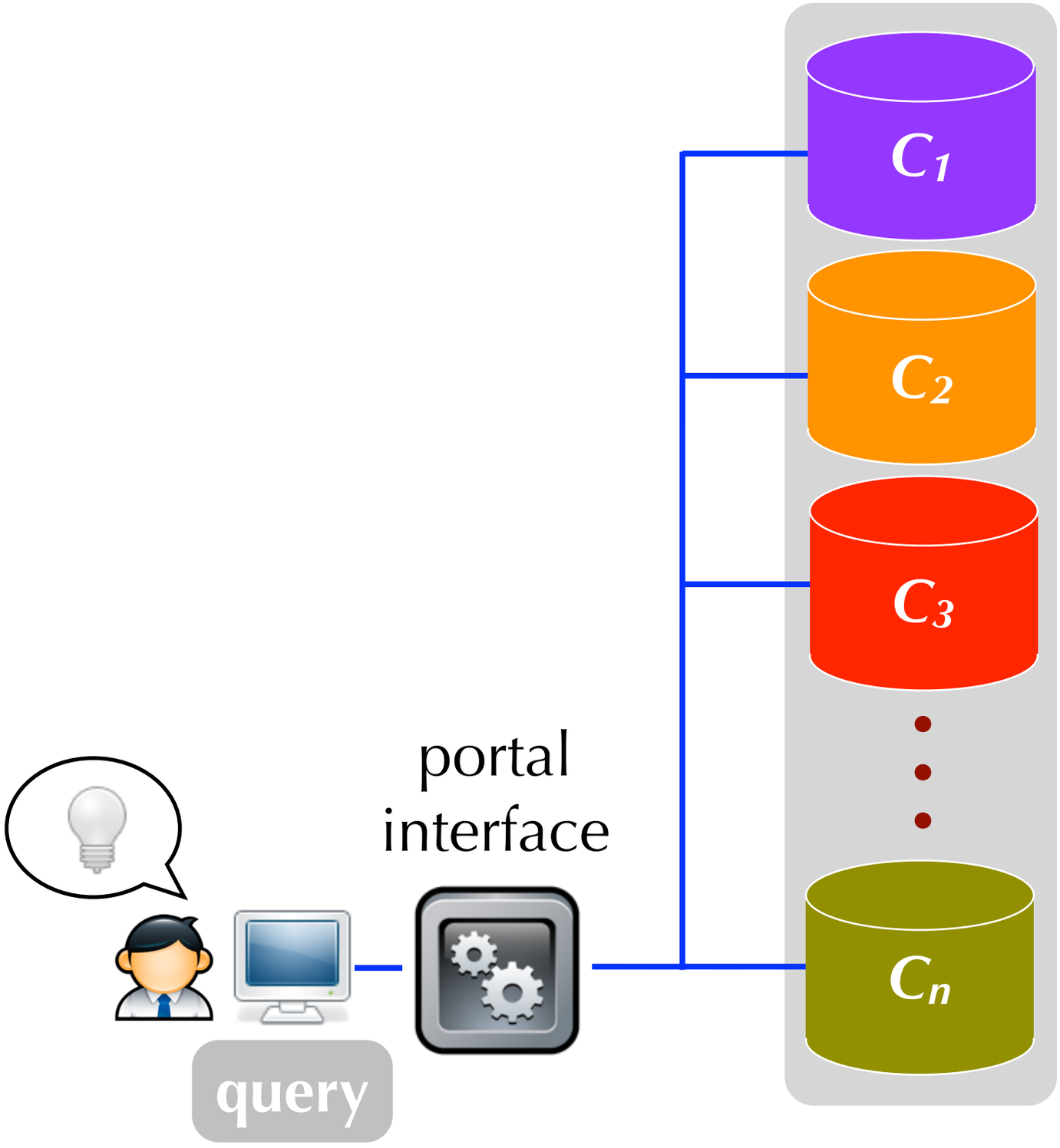
Large Document Models

- Discussion: potential limitations?



Resource Representation

using content

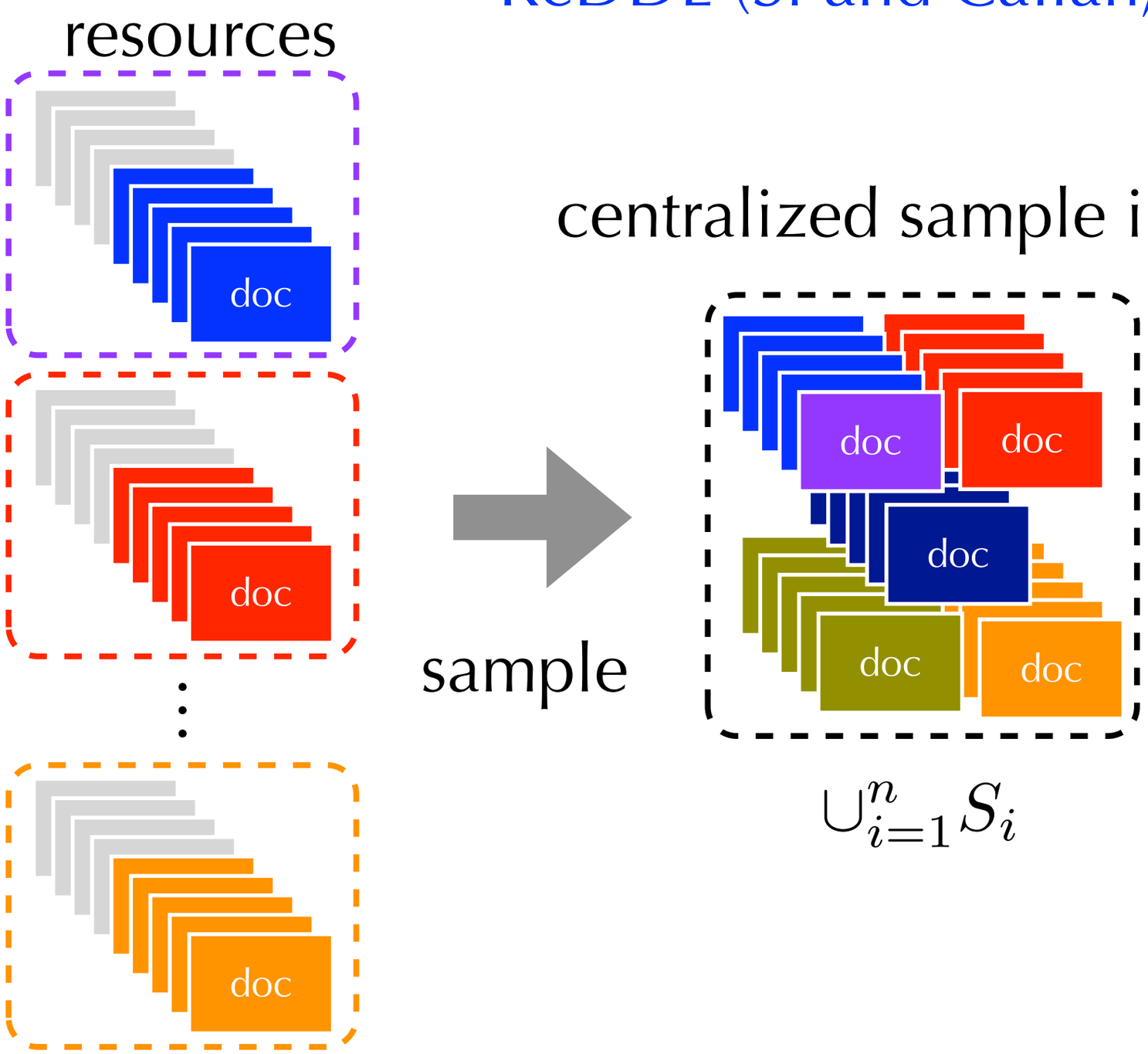


- Term frequencies: selection based on the query-collection similarity

- A set of 'typical' docs: selection based on the predicted relevance of sampled documents

Small Document Models

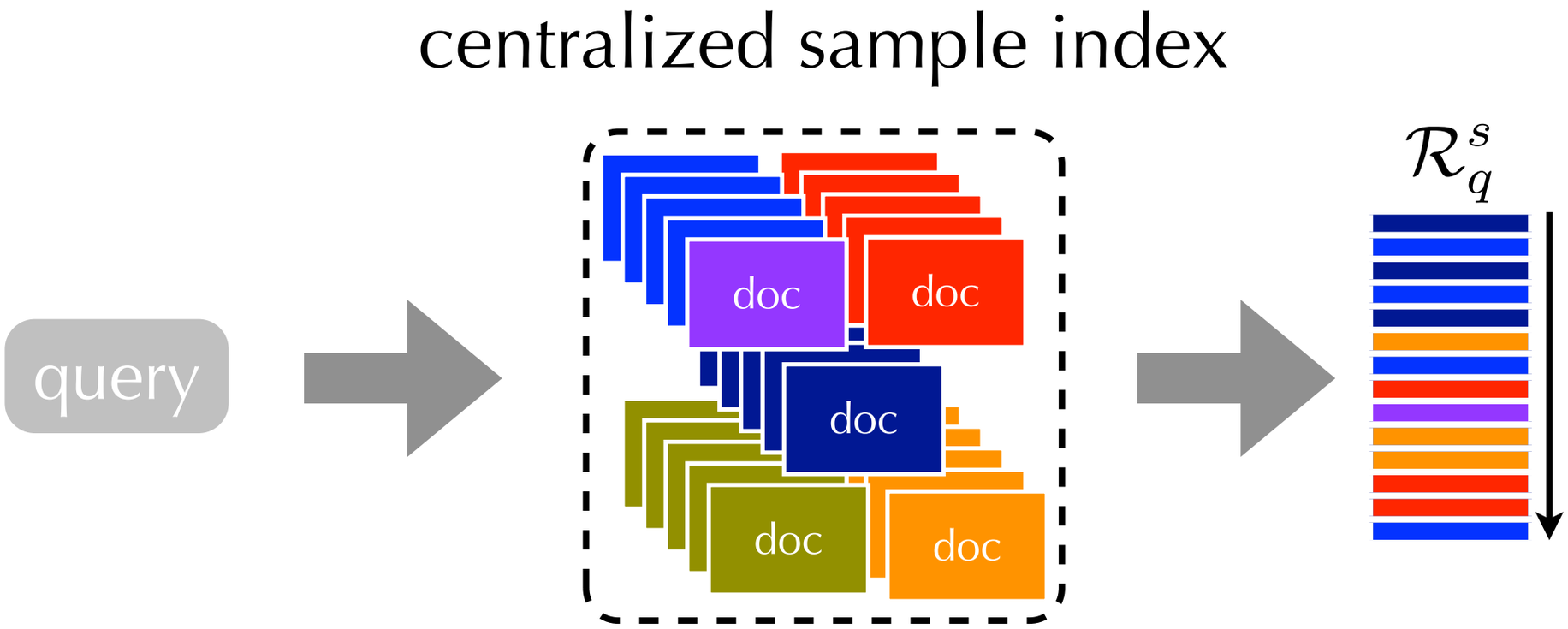
ReDDE (Si and Callan, 2003)



- Combine samples in a centralized index, keeping track of which collection each sample came from

Small Document Models

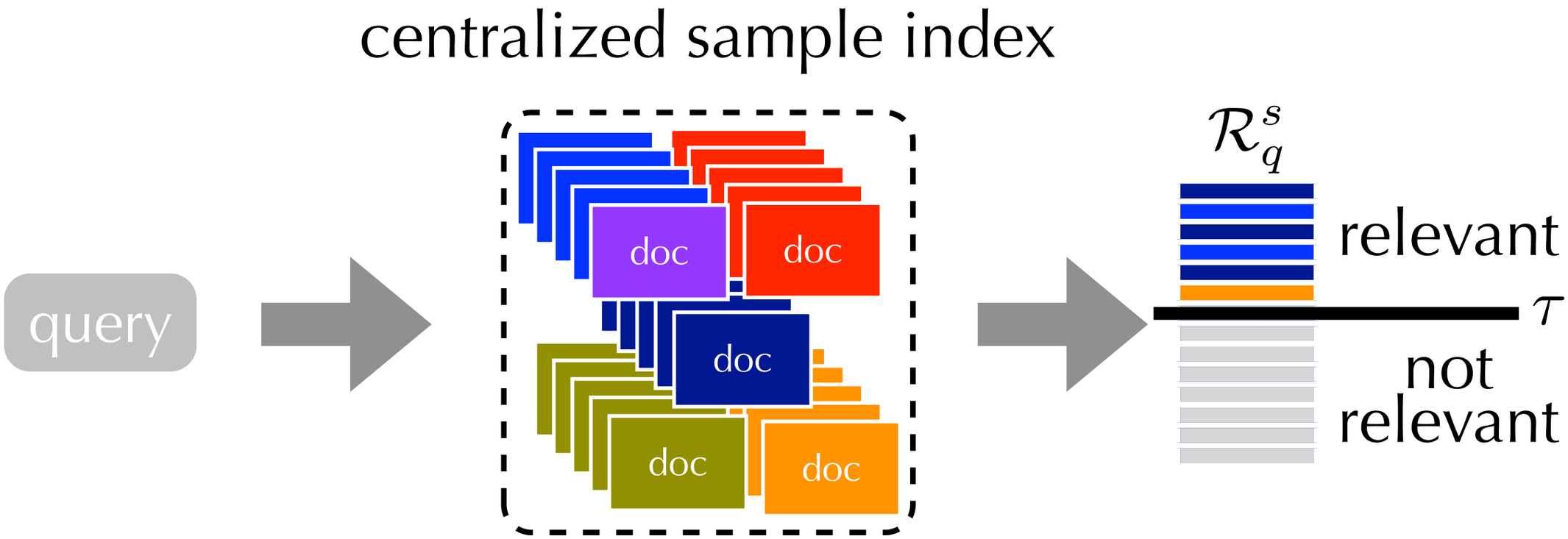
ReDDE (Si and Callan, 2003)



- Given a query, conduct a retrieval from the centralized sample index

Small Document Models

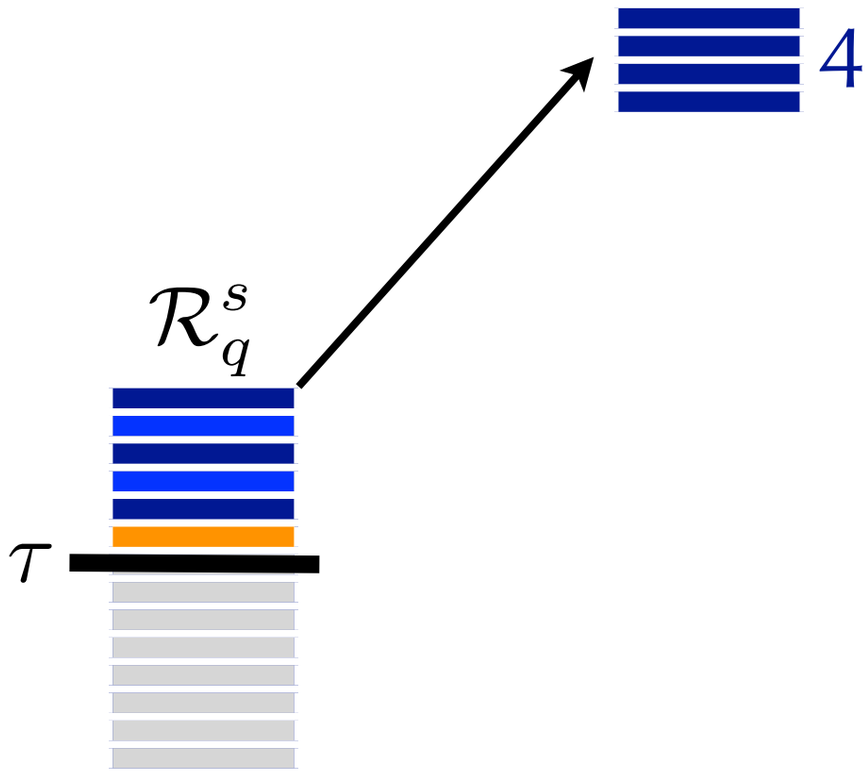
ReDDE (Si and Callan, 2003)

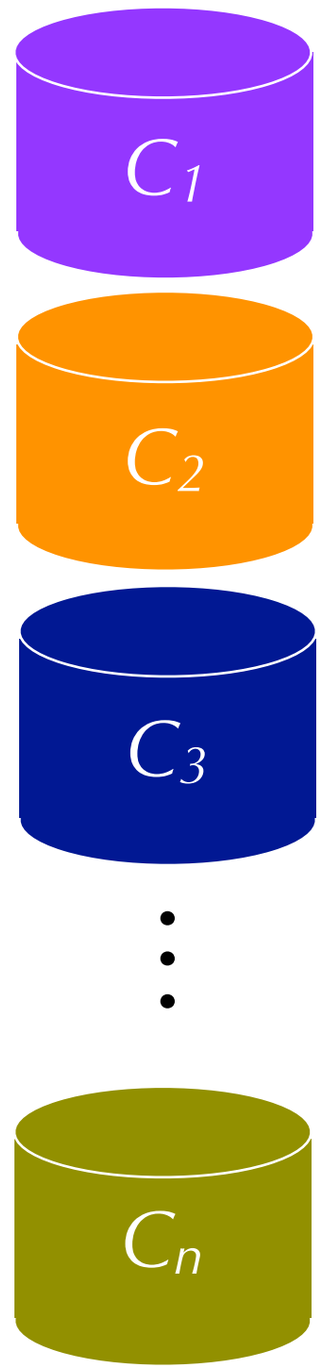


- Use a rank-based threshold to predict a set of relevant samples

Small Document Models

ReDDE (Si and Callan, 2003)



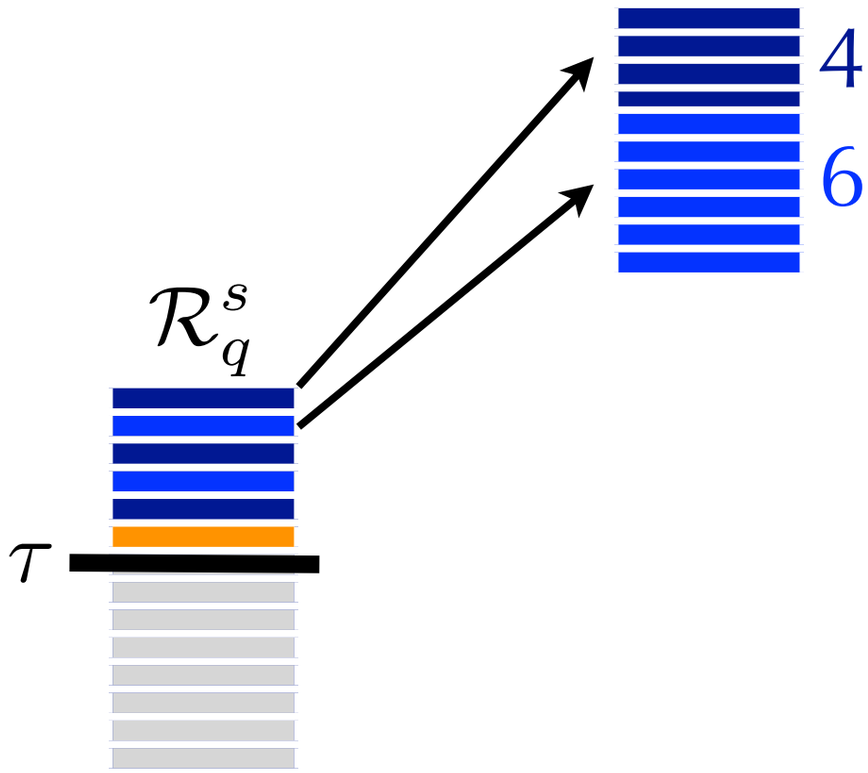


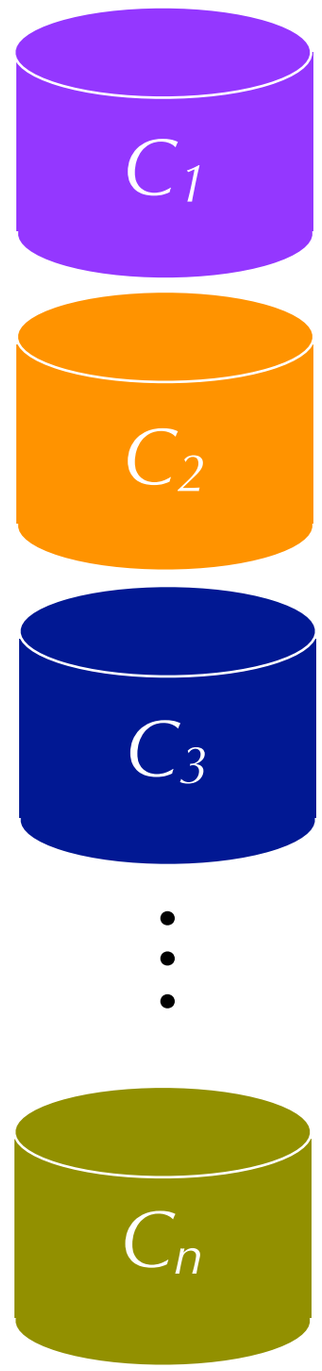
$$\text{scale factor}(C_i) = \frac{|C_i|}{|S_i|}$$

- Assume that each relevant sample represents some number of relevant documents in its original collection

Small Document Models

ReDDE (Si and Callan, 2003)



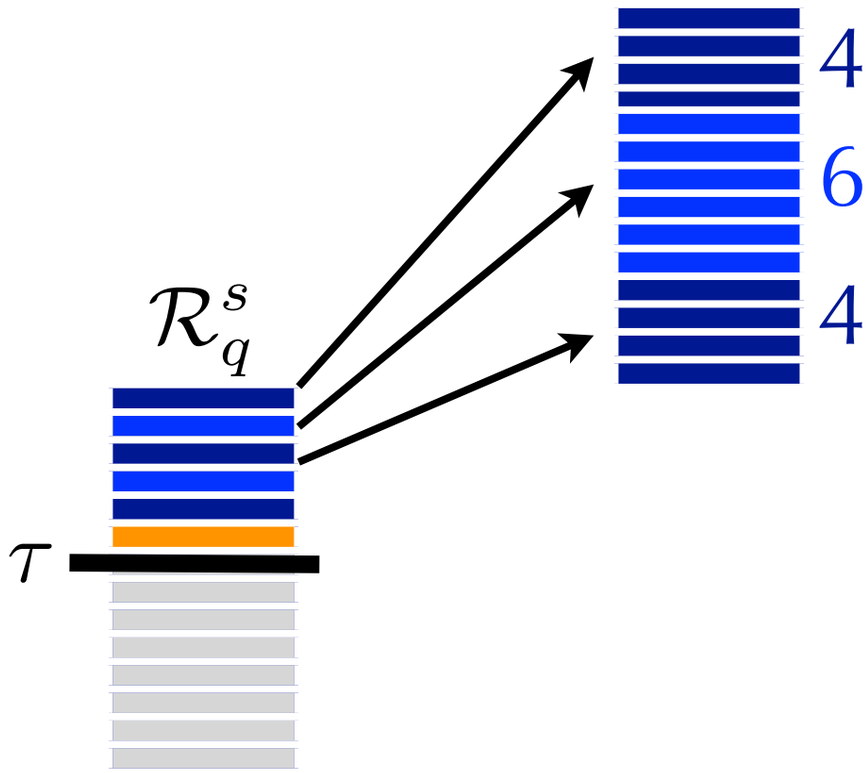


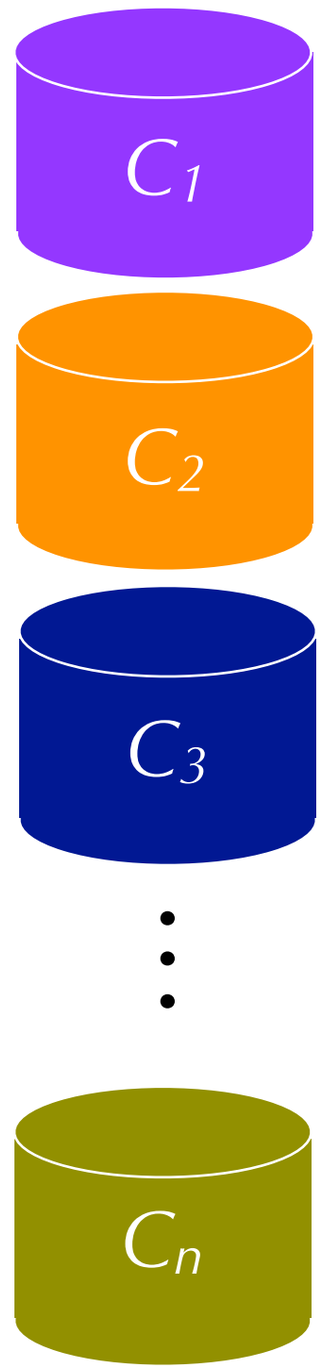
$$\text{scale factor}(C_i) = \frac{|C_i|}{|S_i|}$$

- "Scale-up" sample retrieval

Small Document Models

ReDDE (Si and Callan, 2003)



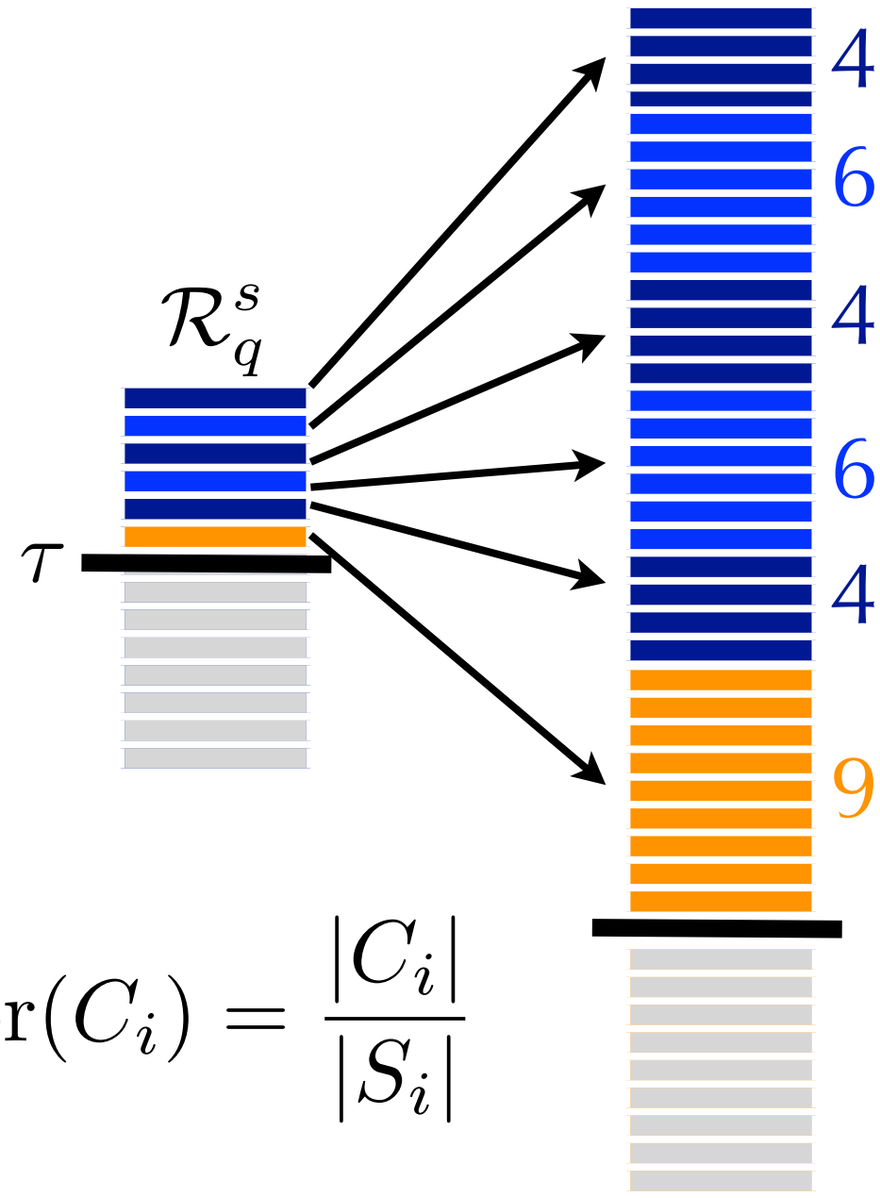


$$\text{scale factor}(C_i) = \frac{|C_i|}{|S_i|}$$

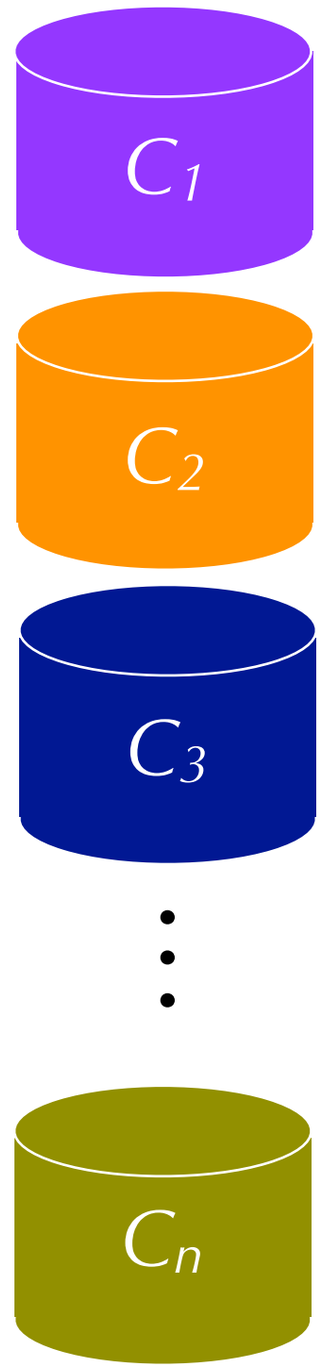
- "Scale-up" sample retrieval

Small Document Models

ReDDE (Si and Callan, 2003)



$$\text{scale factor}(C_i) = \frac{|C_i|}{|S_i|}$$



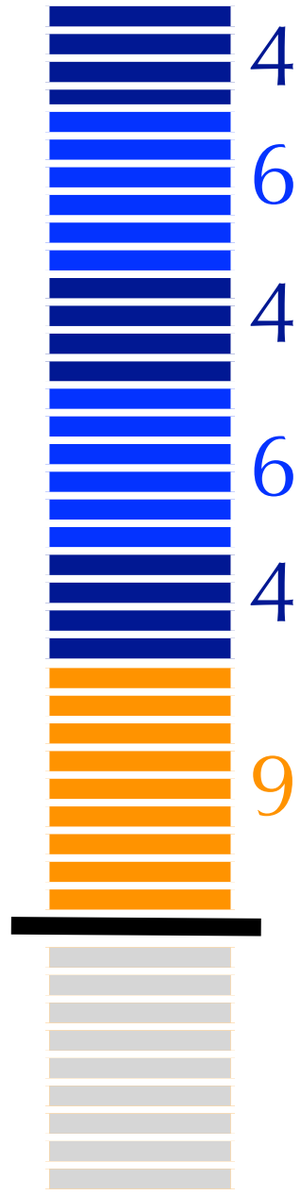
- "Scale-up" sample retrieval

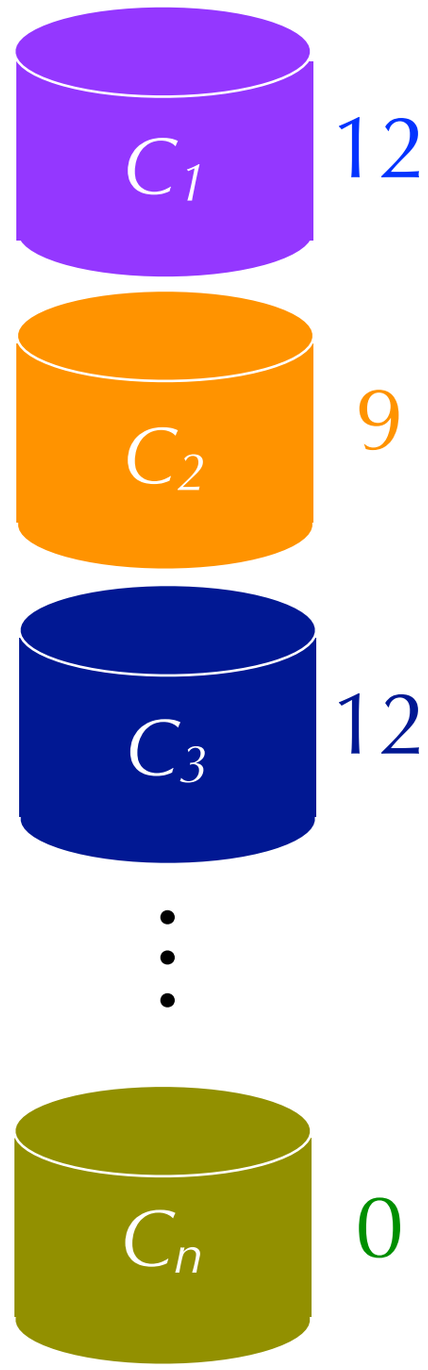
Small Document Models

ReDDE (Si and Callan, 2003)

1. Score collections by their estimated number of relevant documents
2. Select the top k

$$\text{scale factor}(C_i) = \frac{|C_i|}{|S_i|}$$





Small Document Models

ReDDE Variants

- ReDDE can be viewed as a voting method: each (predicted) relevant sample is a vote for its collection
- Discussion: potential limitations?

Small Document Models

ReDDE Variants

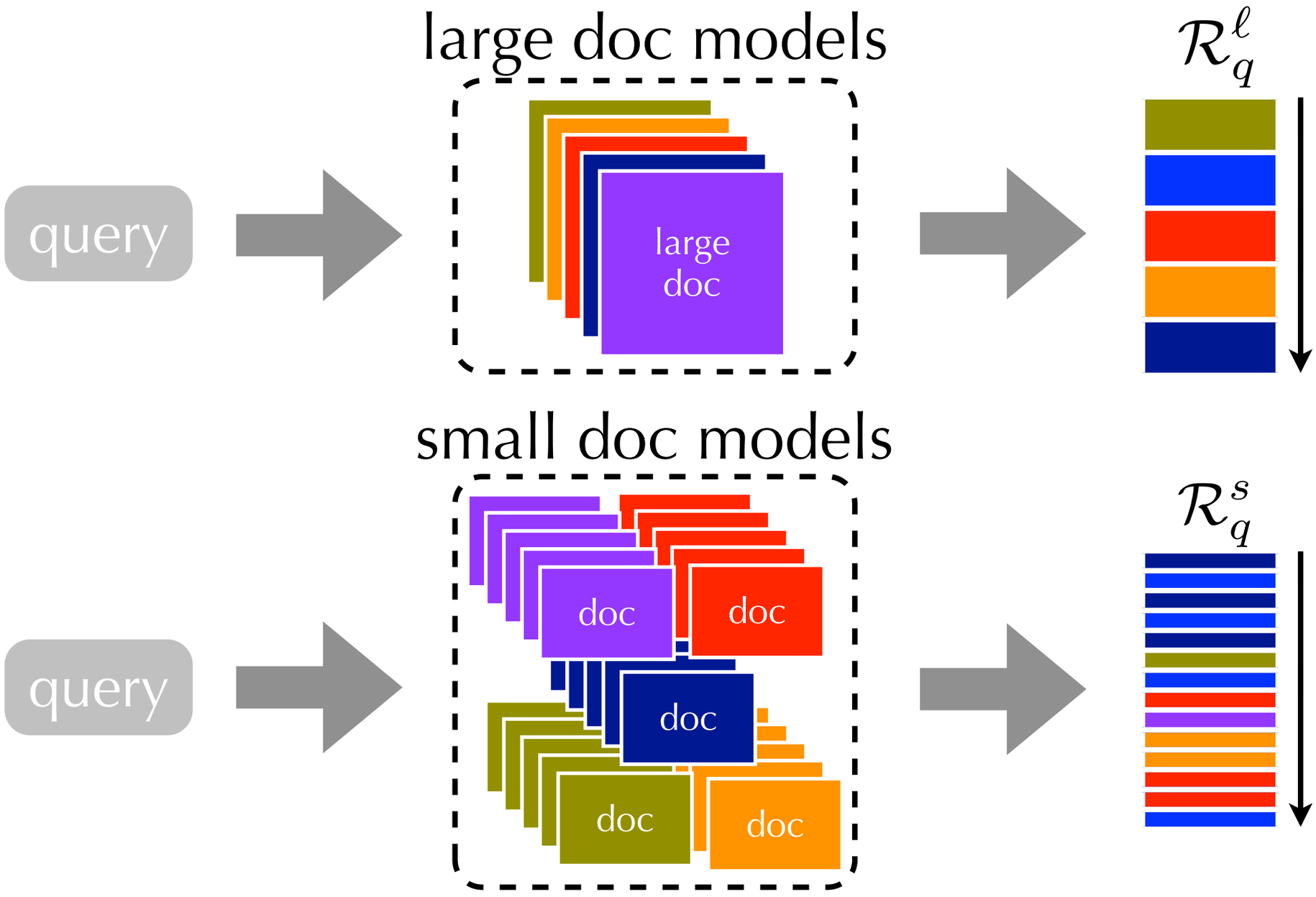
- ReDDE can be viewed as a voting method: each (predicted) relevant sample is a vote for its collection
- Discussion: potential limitations?
  - sensitivity to threshold parameter: samples that are more relevant (i.e., ranked higher) should get more votes (Shokouhi, 2007; Thomas, 2009)
  - a resource may not retrieve its relevant documents: samples from resources predicted to be more reliable should get more votes (Si and Callan, 2004)
- No ReDDE variant outperforms another across all experimental testbeds

Resource Selection
ReDDE vs. CORI

- ReDDE wins: it never does worse and often does better
- ReDDE outperforms CORI when the collection size distribution is skewed
  - CORI is biased towards small, topically-focused collections
  - favors collections that are proportionately relevant
  - misses large collections with many relevant documents

Resource Selection

content-based methods

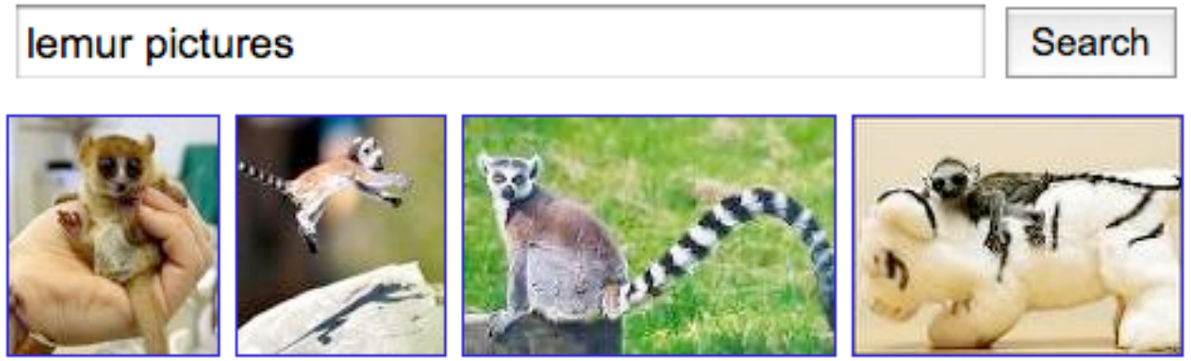


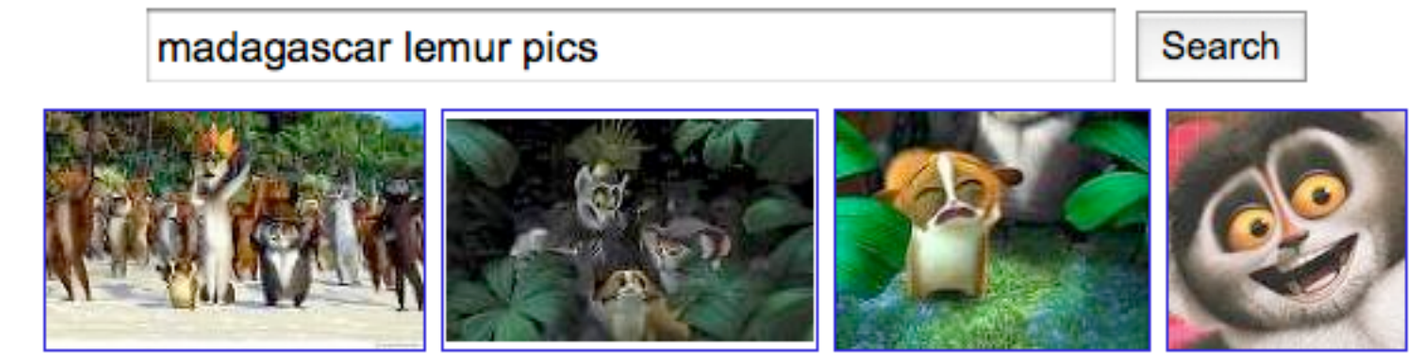
- Resource relevance as a function of content relevance

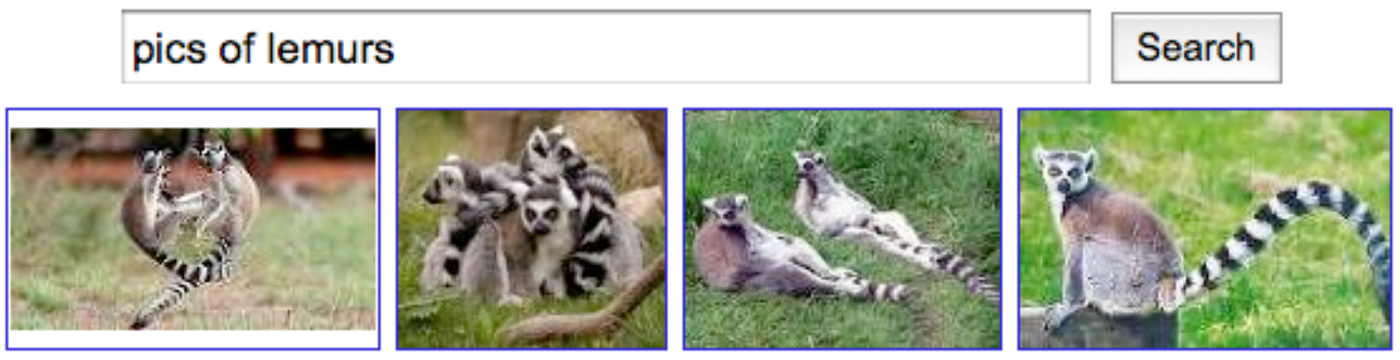
Resource Selection

query-similarity methods

- Key assumption: similar queries retrieve similar results







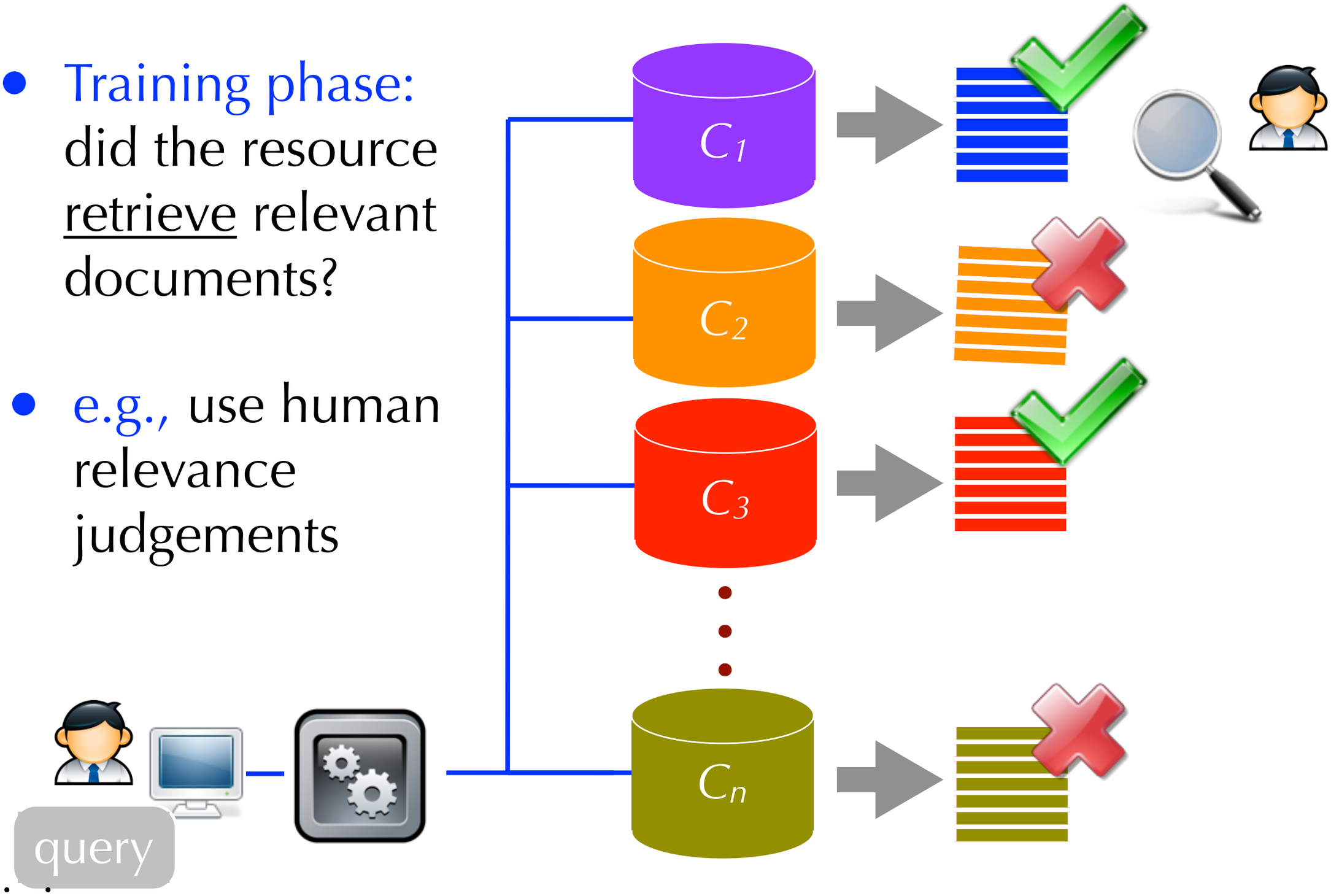
Resource Selection
query-similarity methods

- Select resources based on their expected retrieval effectiveness for the given query
- Requires two components:

1. retrieval effectiveness: a way to determine that a previously seen query produced an effective retrieval from the resource
2. query-similarity: a way to predict that a new (unseen) query will retrieve similar results from the resource

Query-Similarity Methods
(Voorhees *et al.*, 1995)

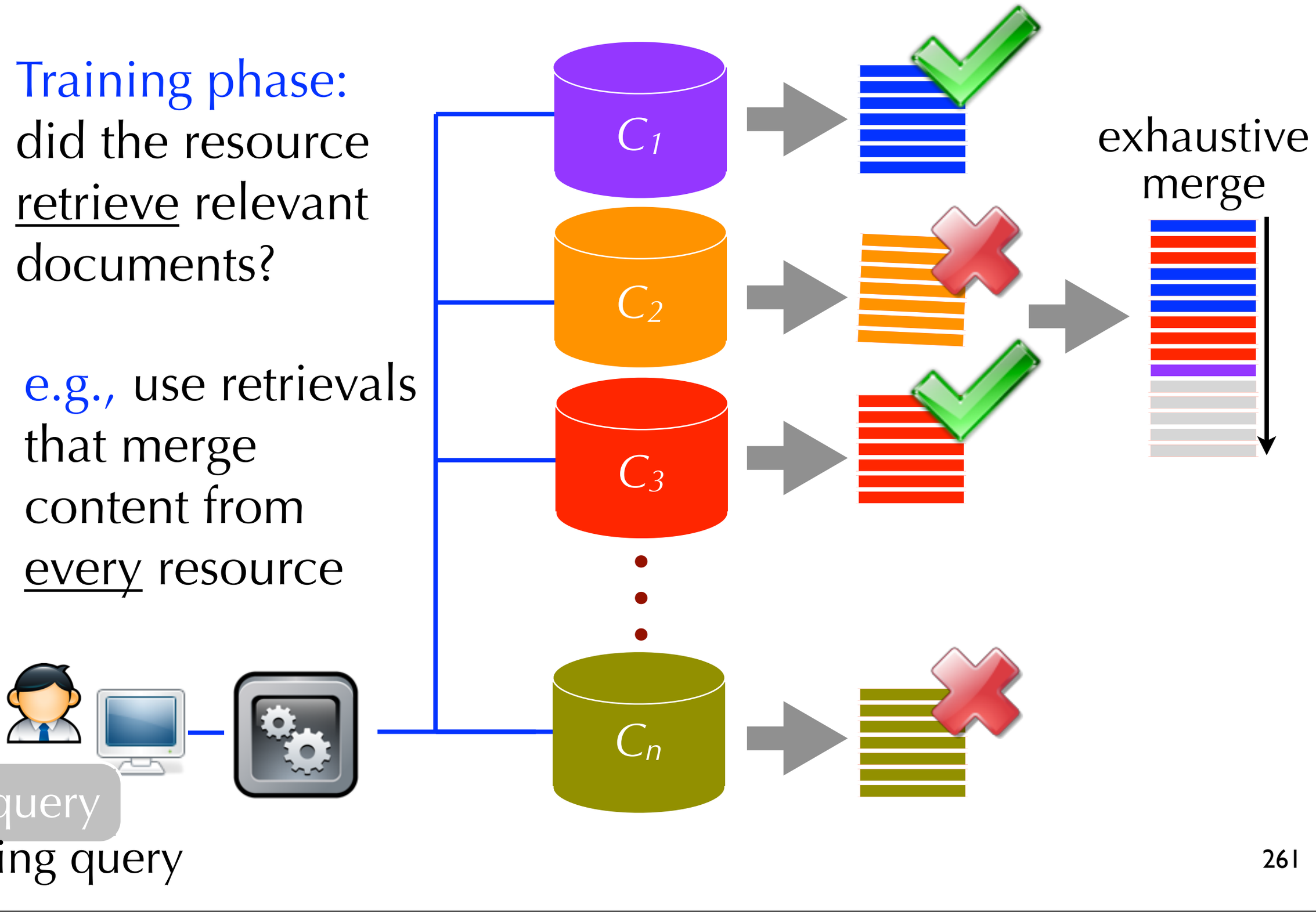
- Training phase: did the resource retrieve relevant documents?
- e.g., use human relevance judgements



training query

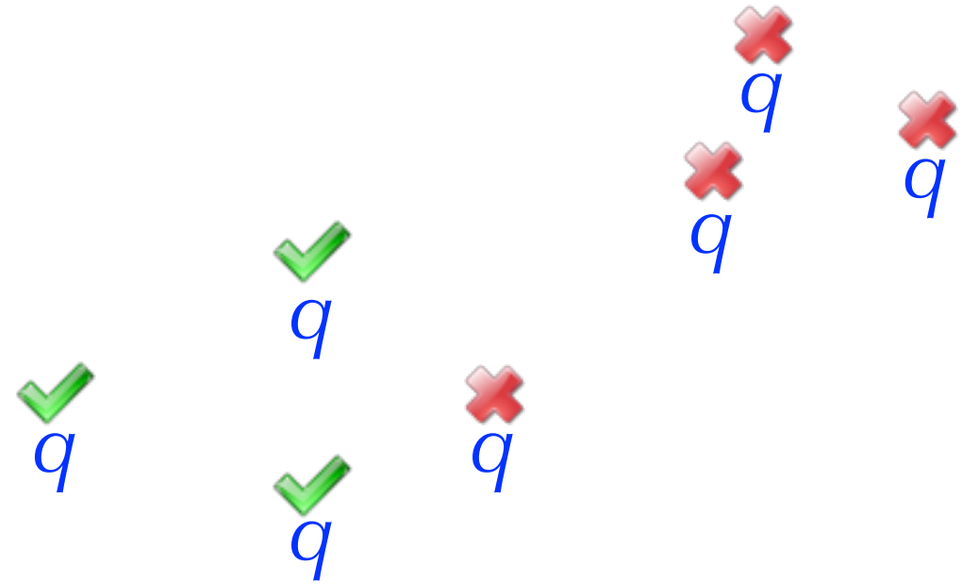
Query-Similarity Methods
(Arguello *et al.*, 2008)

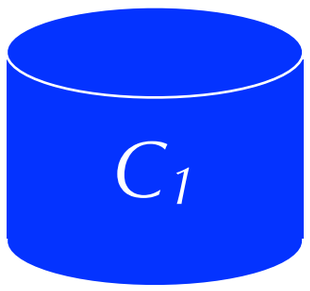
- Training phase: did the resource retrieve relevant documents?
- e.g., use retrievals that merge content from every resource



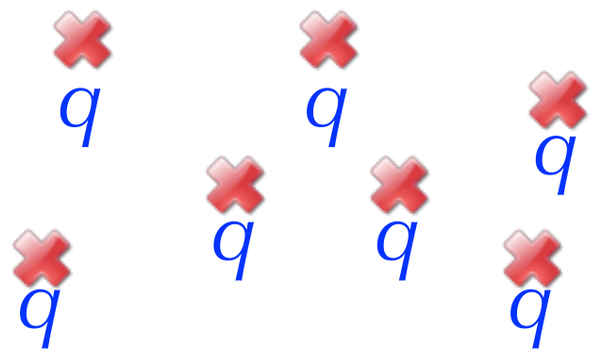
Query-Similarity Methods

- Training phase: did the resource retrieve relevant documents?



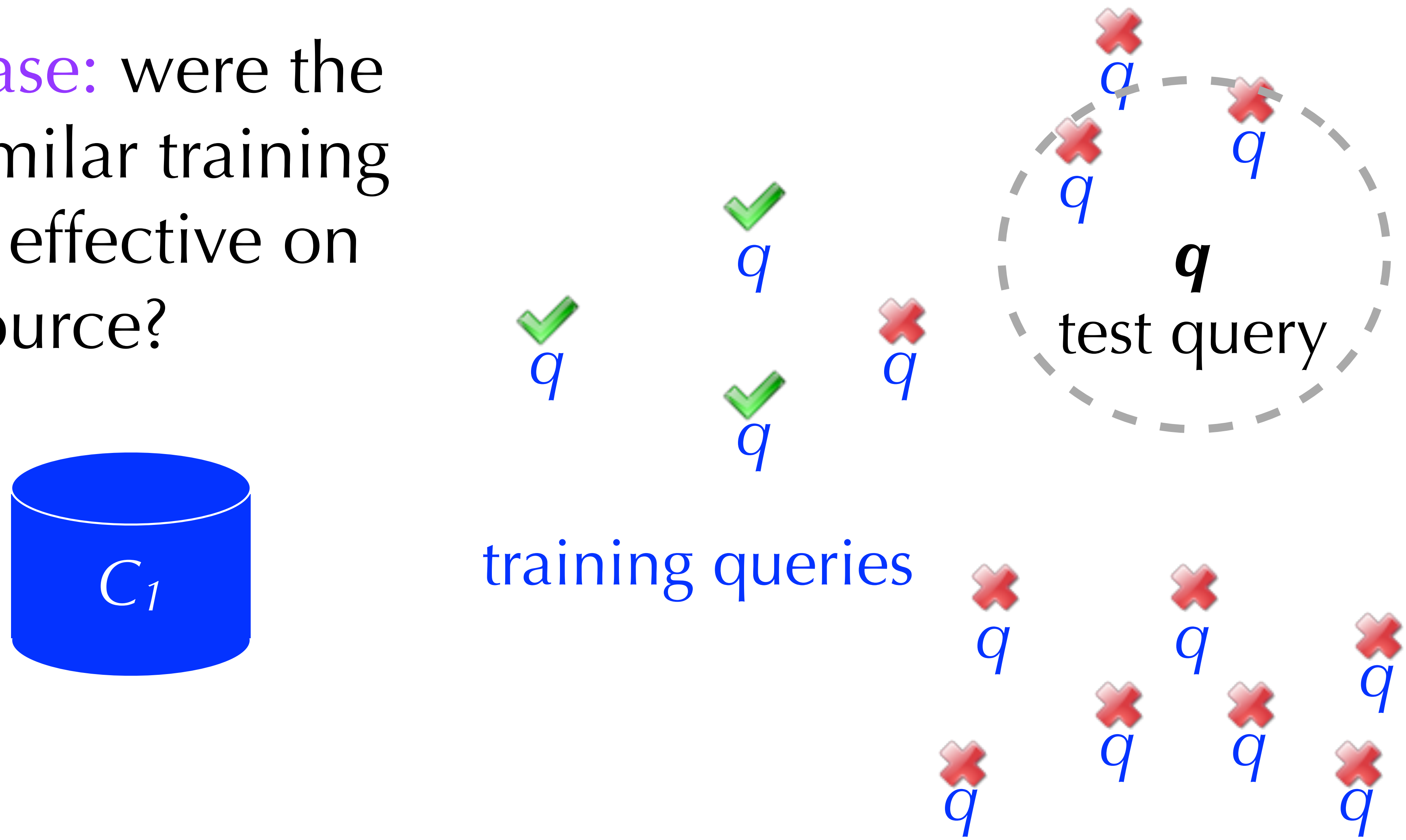


training queries



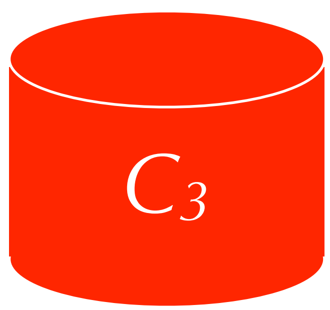
Query-Similarity Methods

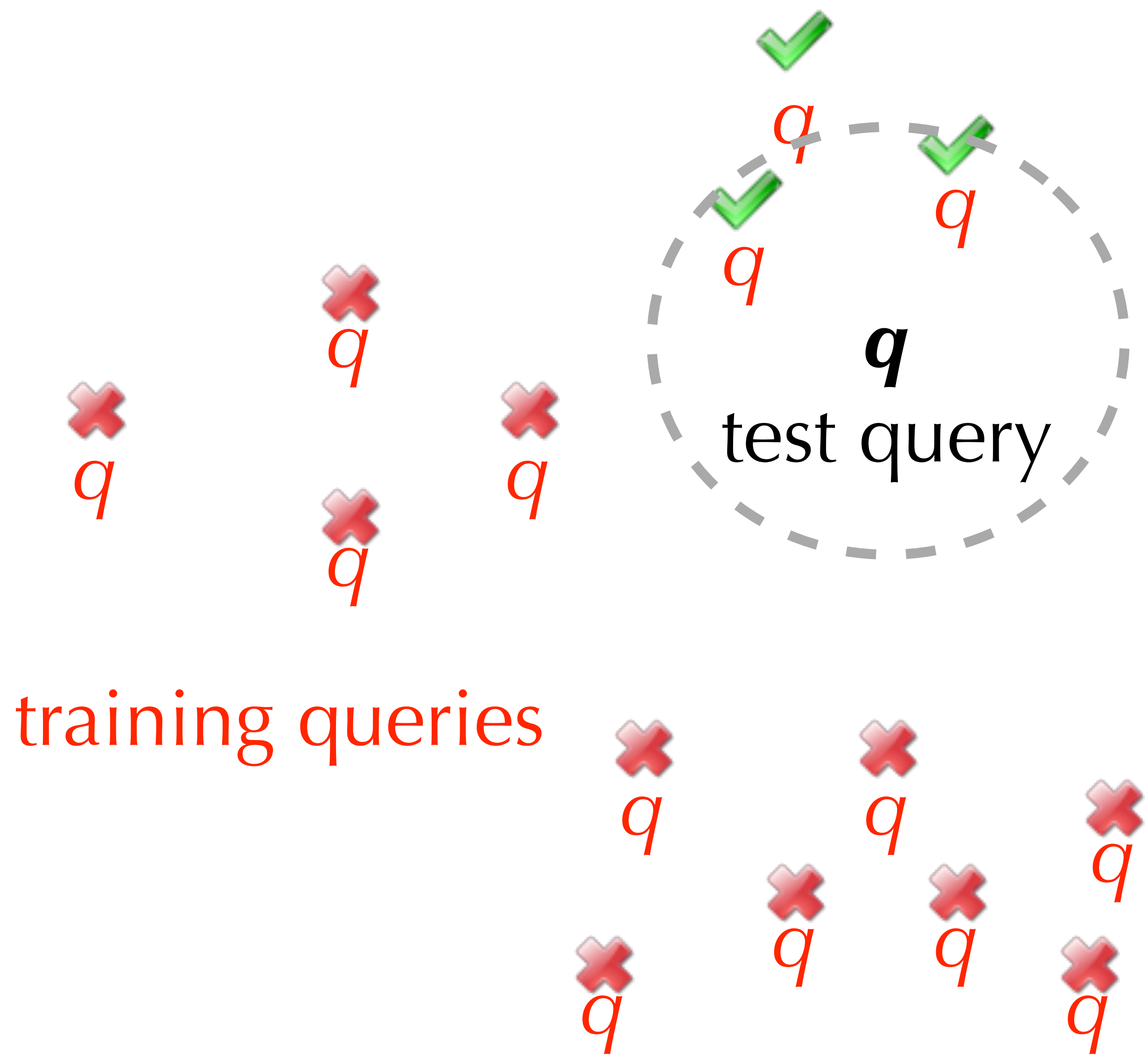
- Test phase: were the most similar training queries effective on the resource?



Query-Similarity Methods

- Test phase: were the most similar training queries effective on the resource?

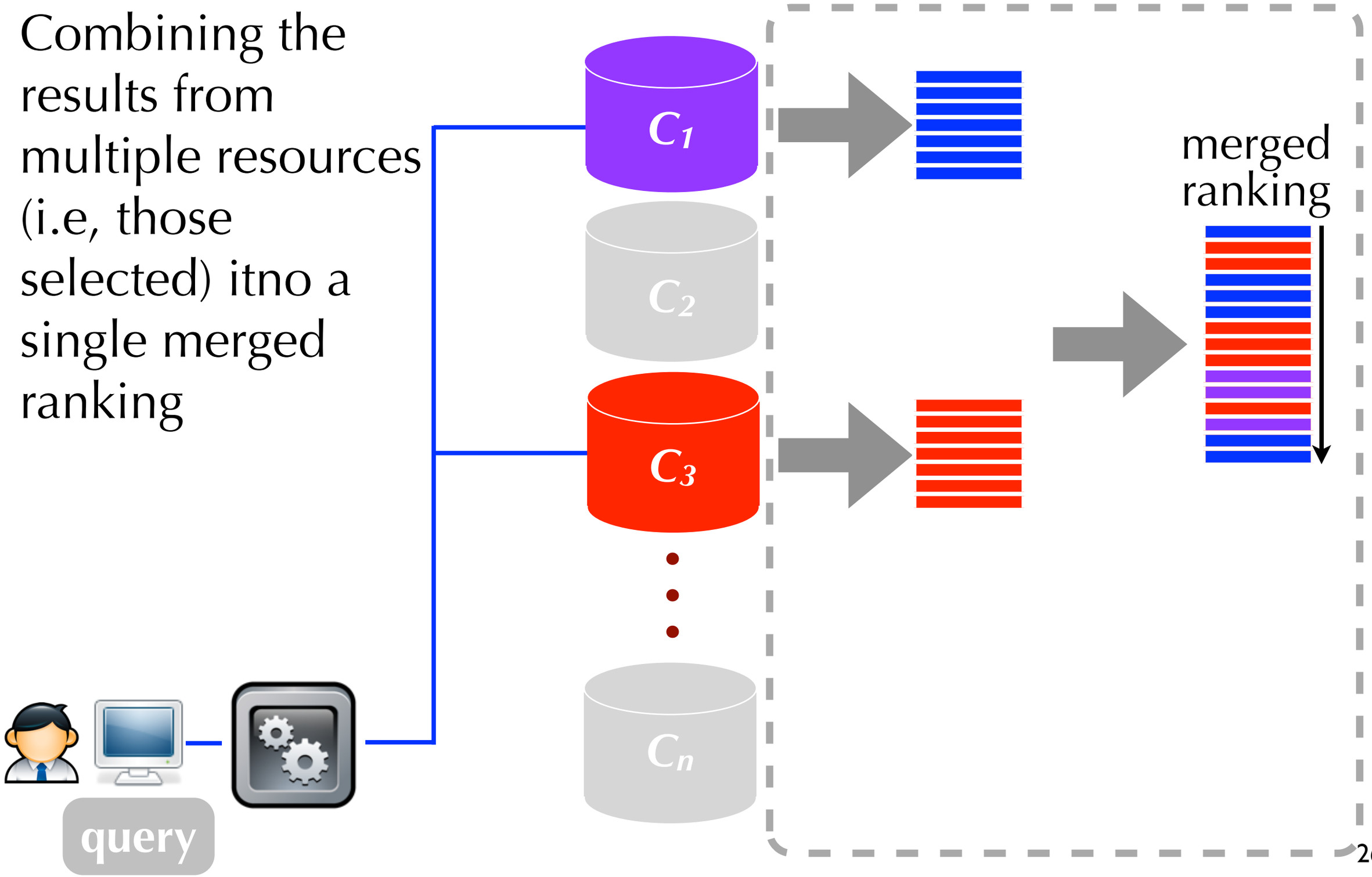




Results Merging

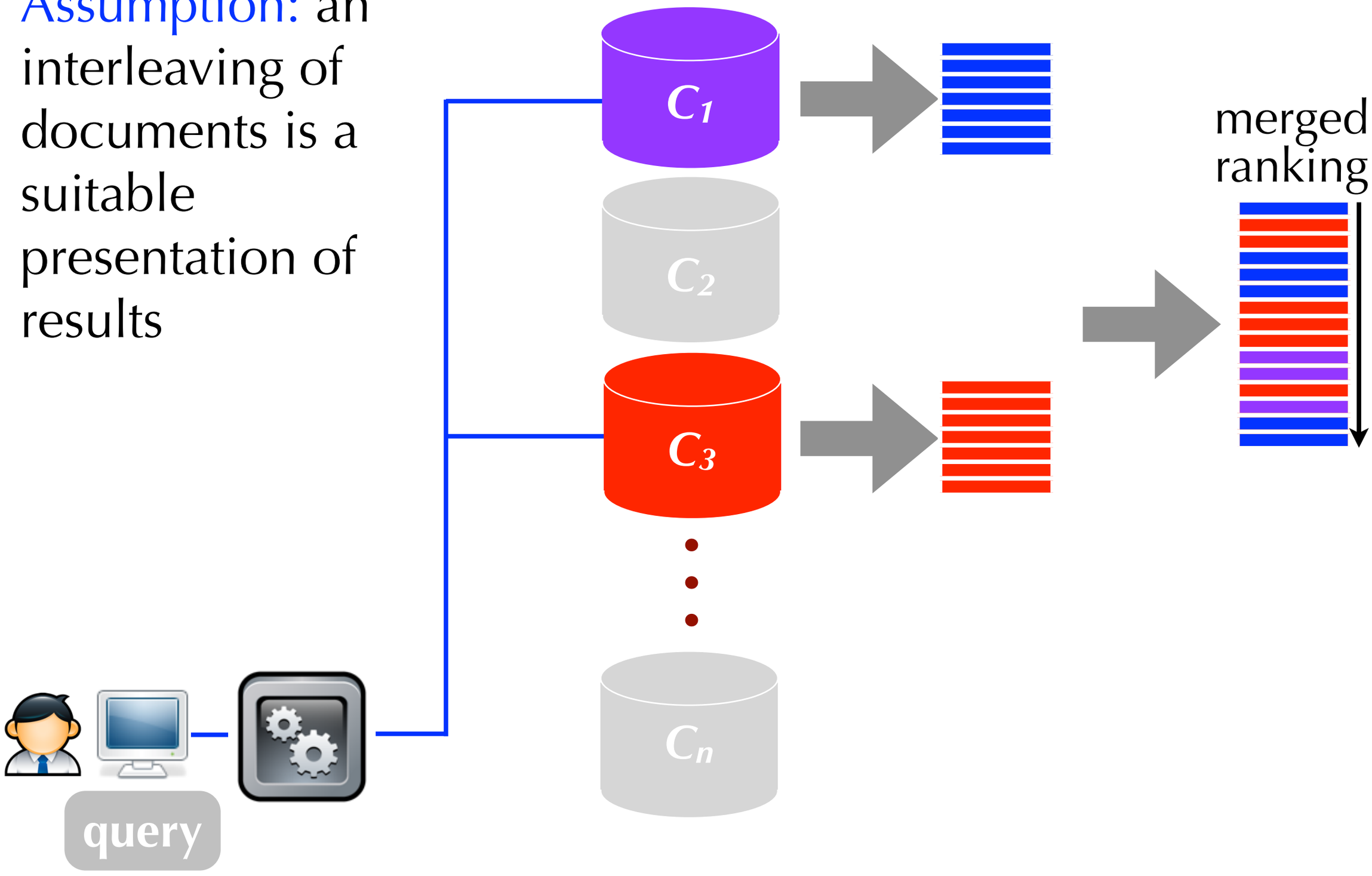
Results Merging

- Combining the results from multiple resources (i.e, those selected) itno a single merged ranking



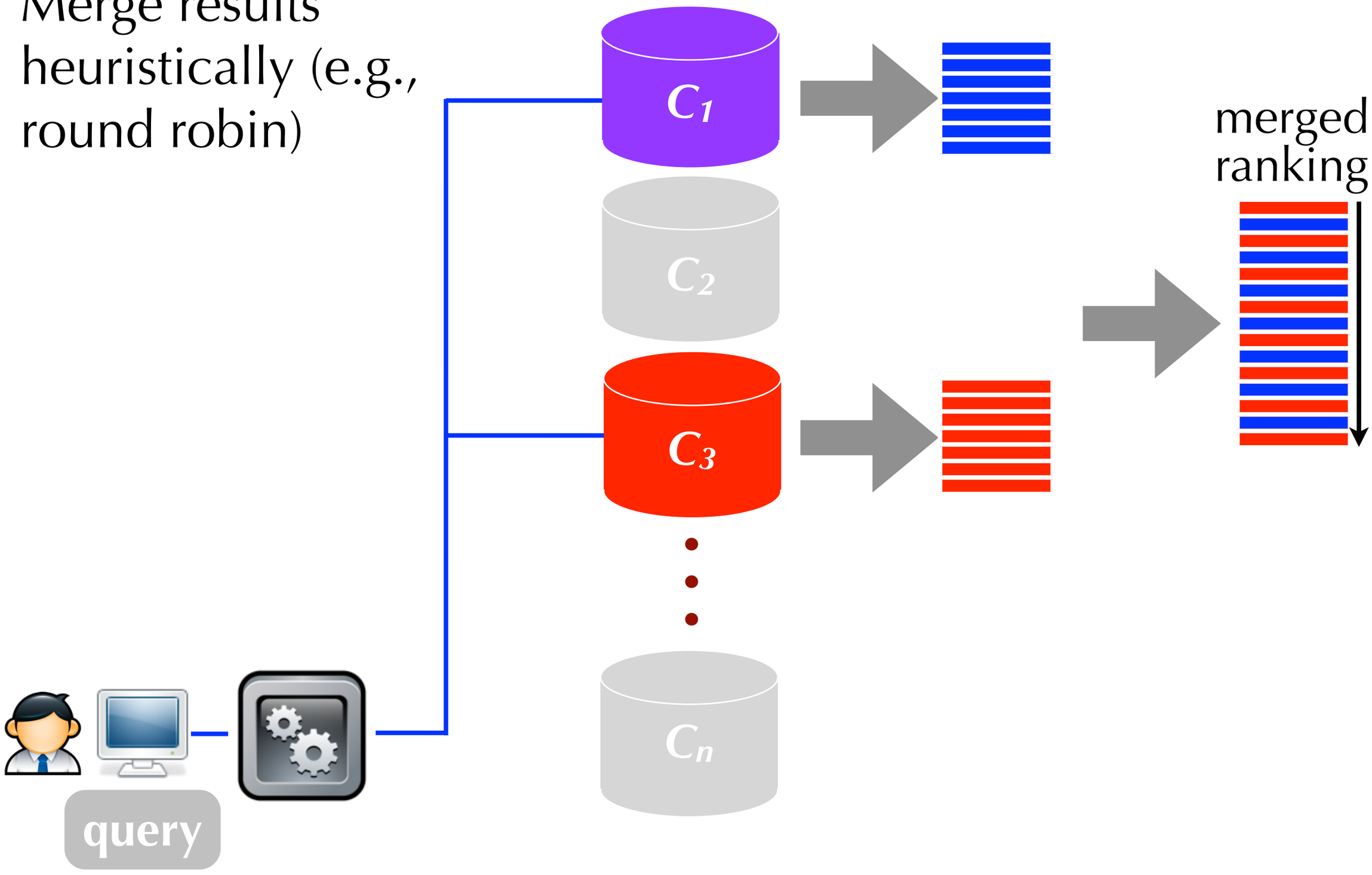
Results Merging

- Assumption: an interleaving of documents is a suitable presentation of results



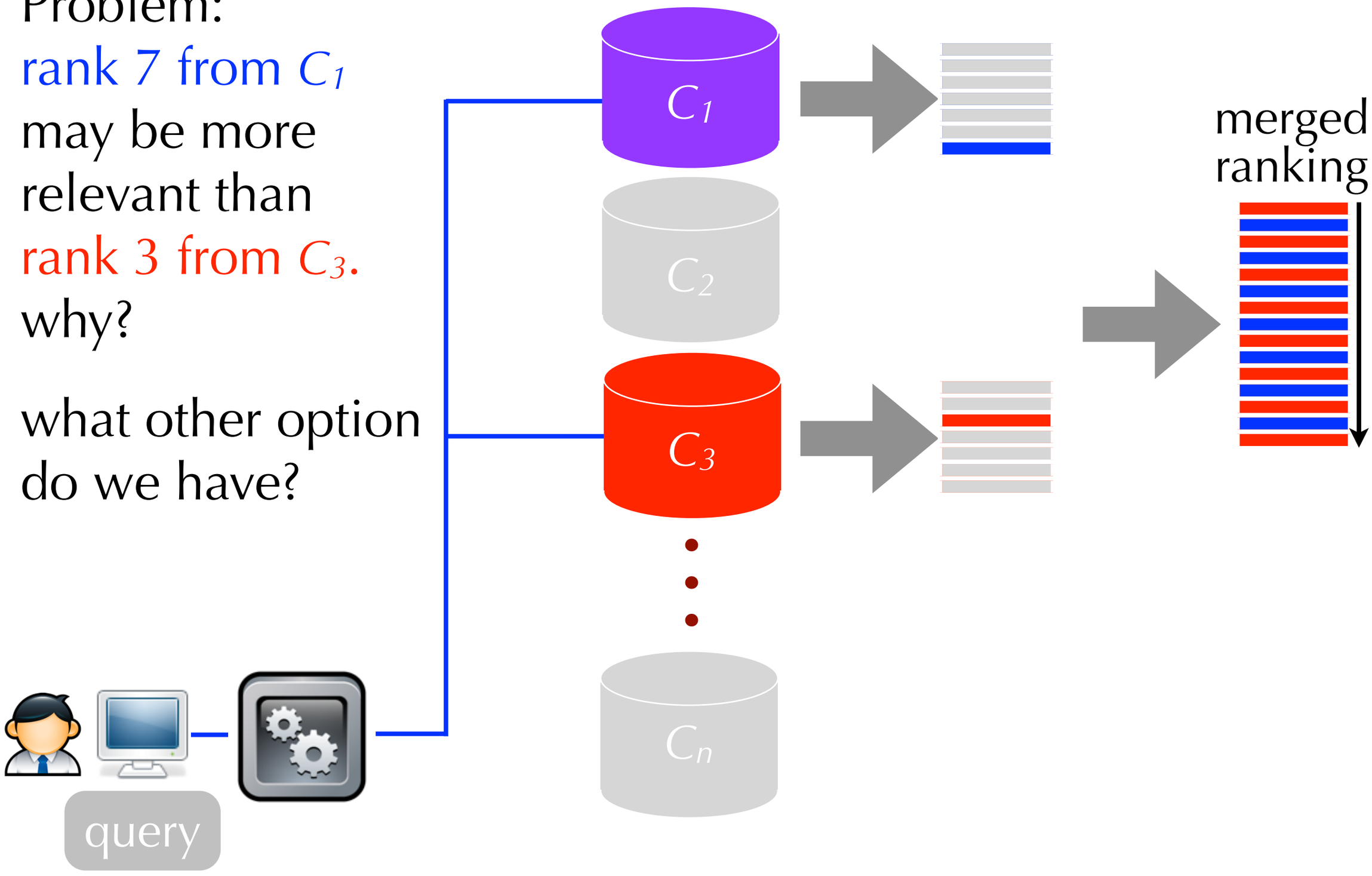
Results Merging
Naive Interleaving

- Merge results heuristically (e.g., round robin)



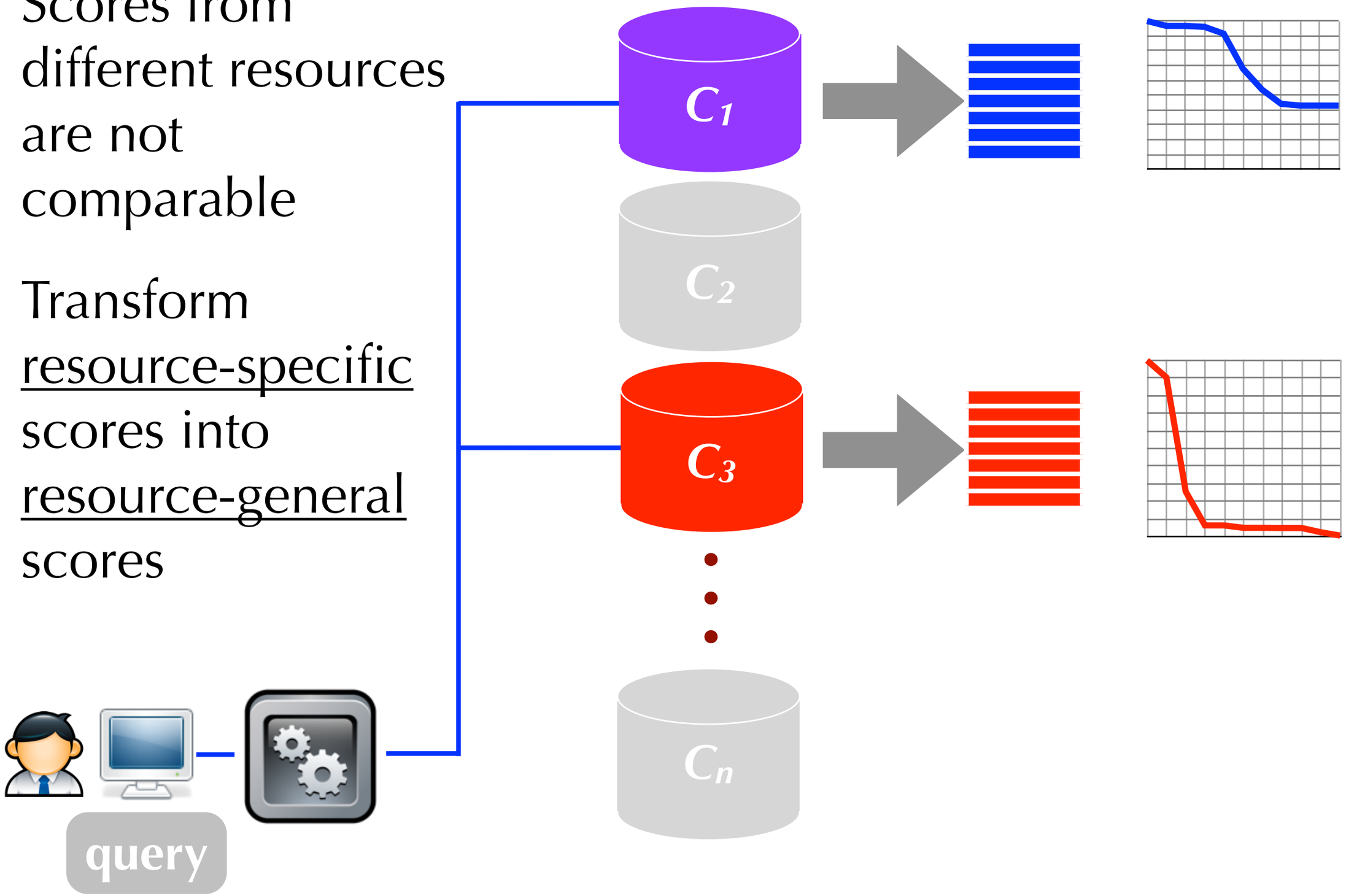
Results Merging
Naive Interleaving

- Problem:
  rank 7 from C_1
  may be more
  relevant than
  rank 3 from C_3.
  why?
- what other option
  do we have?



Results Merging
Score Normalization

- Scores from different resources are not comparable
- Transform resource-specific scores into resource-general scores



Results Merging

CORI-Merge (Callan *et al.*, 1995)

- Combine resource ranking and document ranking scores

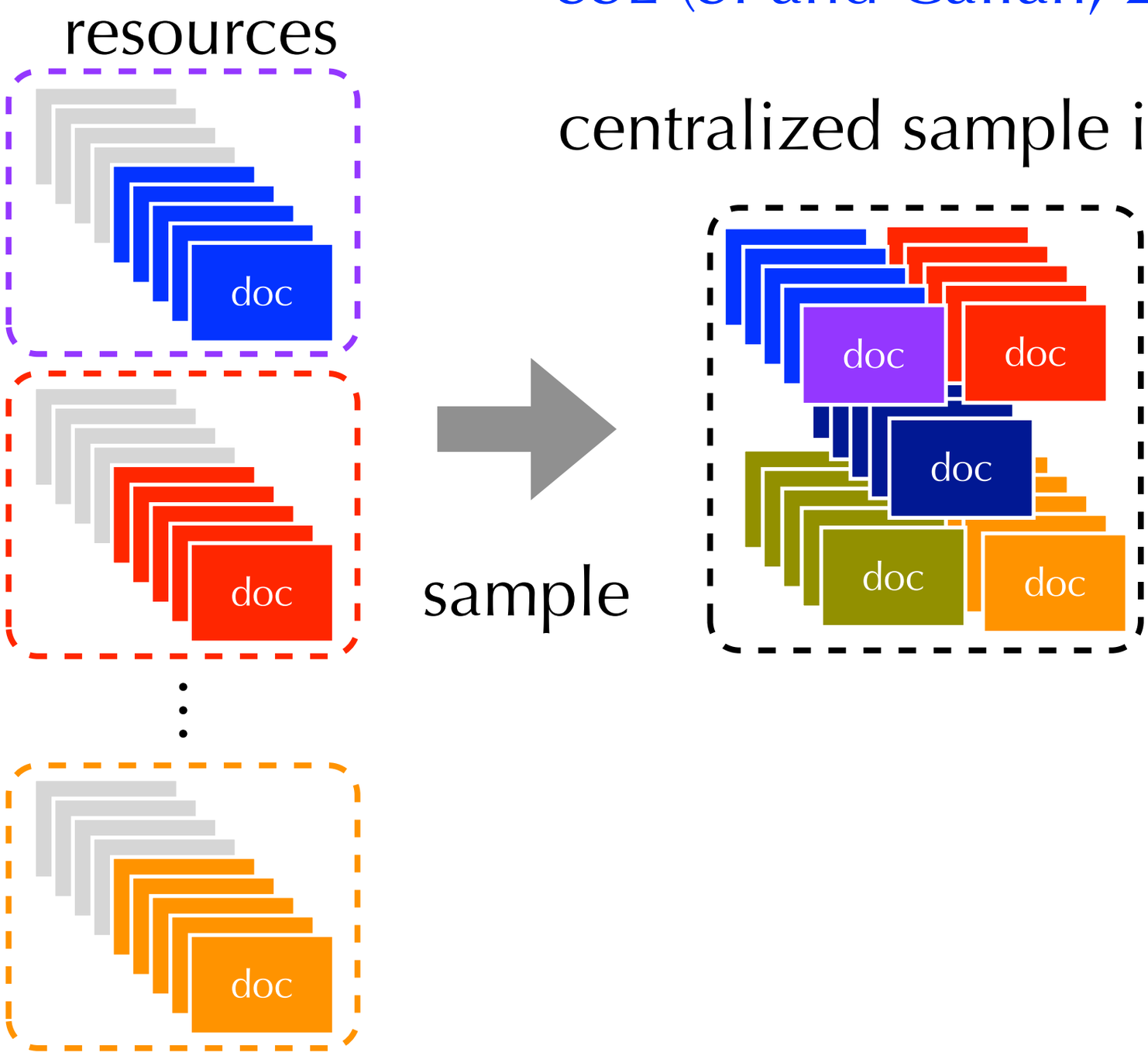
$$S_C(D) = \frac{S'_i(D) + 0.4 \times S'_i(D) \times S'(C_i)}{1.4}$$

$$S'_i(D) = \frac{S_i(D) - S_i(D_{\min})}{S_i(D_{\max}) - S_i(D_{\min})}$$

$$S'(C_i) = \frac{S(C_i) - S(C_{\min})}{S(C_{\max}) - S(C_{\min})}$$

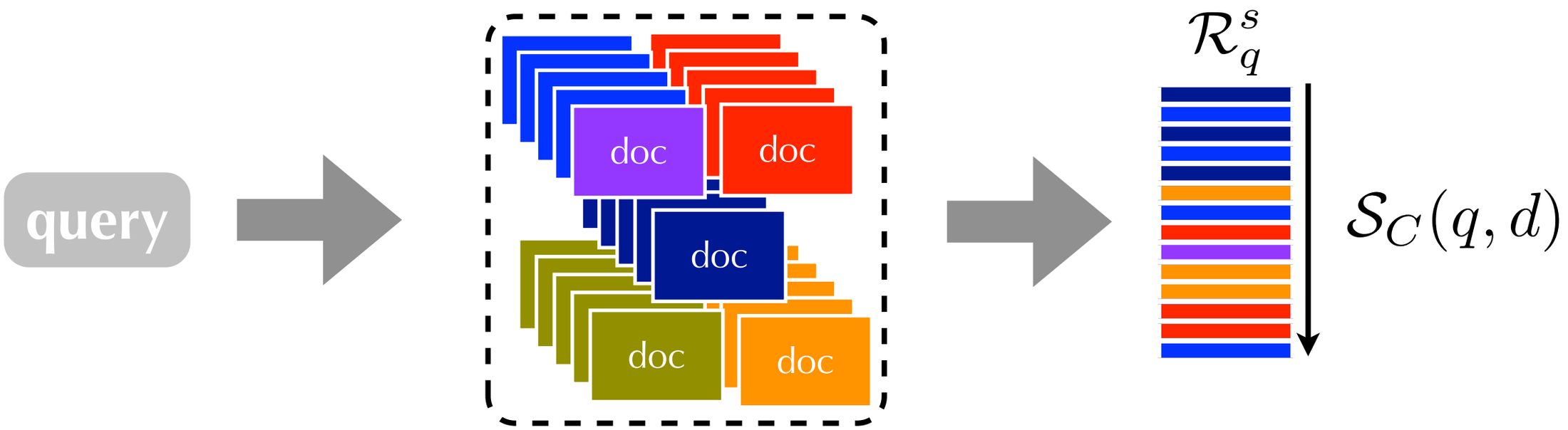
Results Merging

SSL (Si and Callan, 2003)



Results Merging
SSL (Si and Callan, 2003)

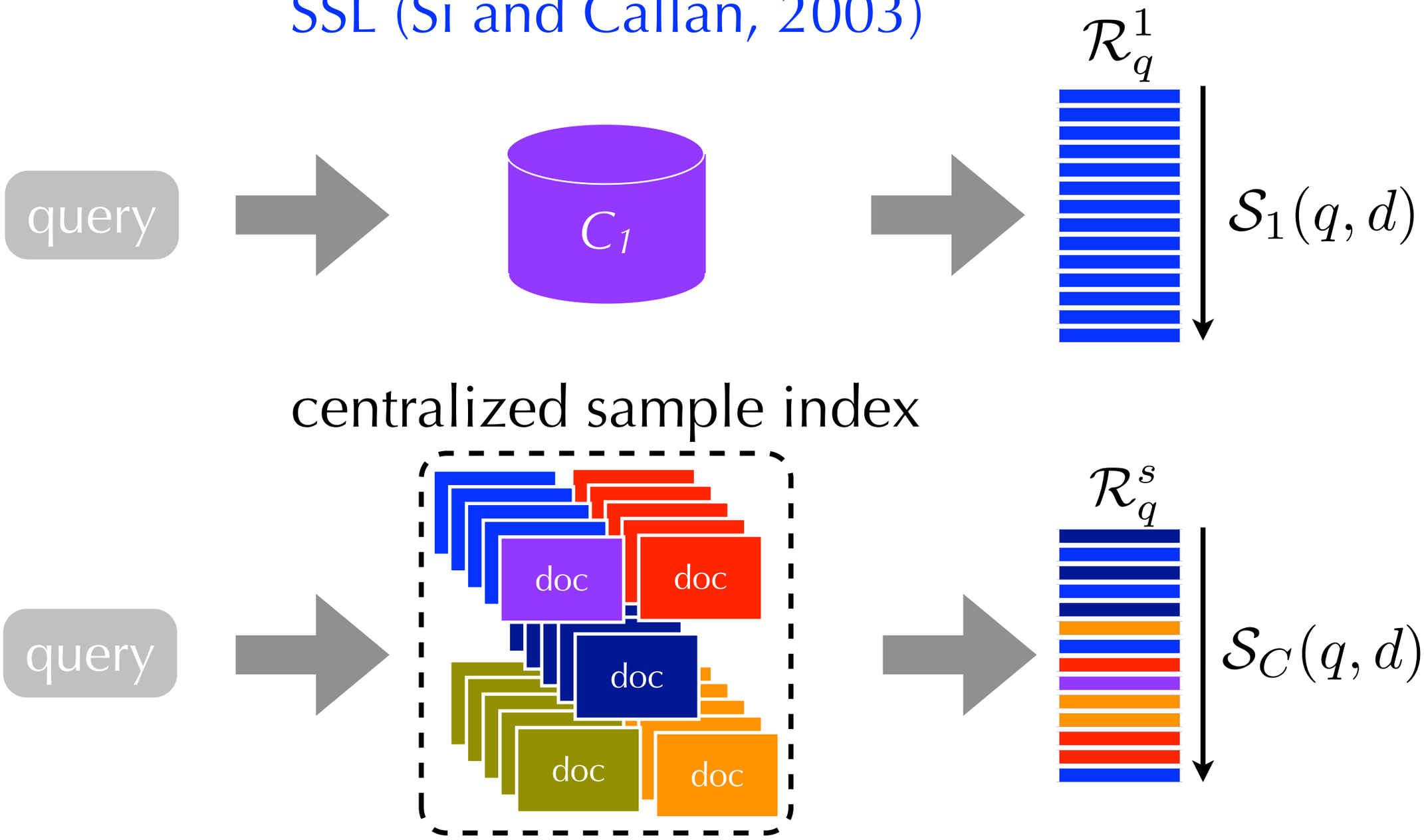
centralized sample index



- Assumption: centralized sample index scores are directly comparable
  - same ranking/scoring algorithm
  - same IDF values
  - same document-length normalization

Results Merging

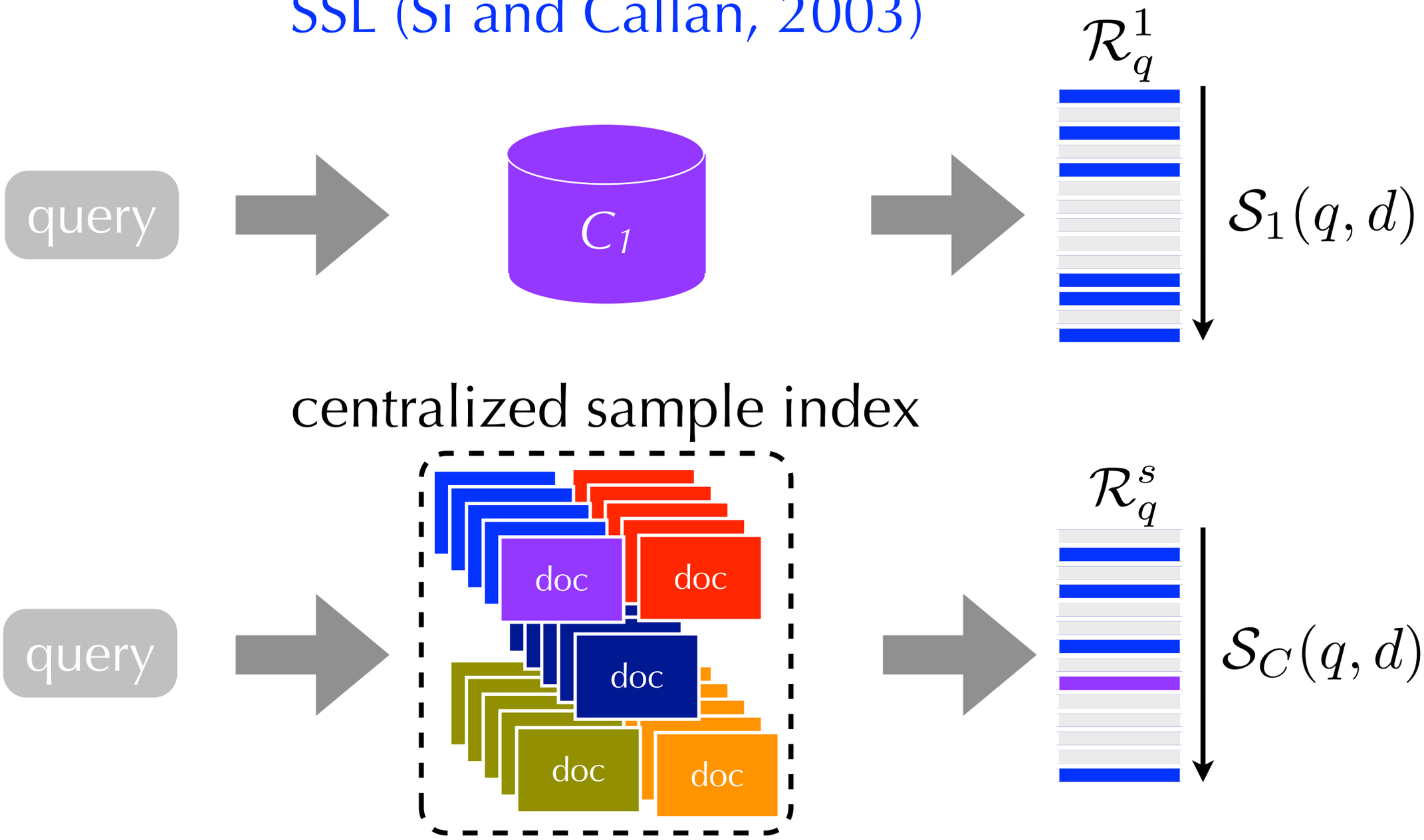
SSL (Si and Callan, 2003)



- Objective: given a query, transform C_1 scores to values that are comparable across collections

Results Merging

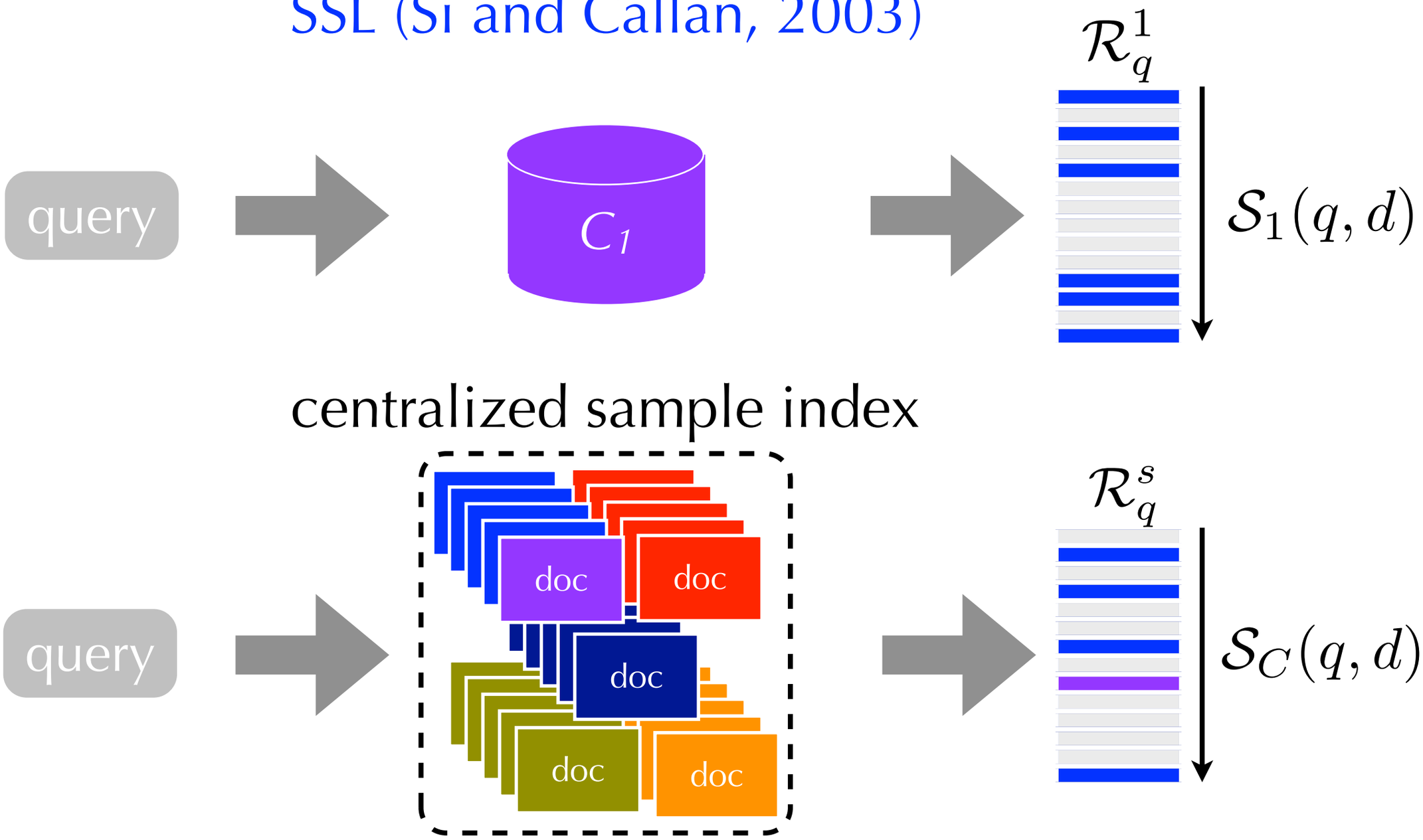
SSL (Si and Callan, 2003)



- Step 1: identify the overlap documents

Results Merging

SSL (Si and Callan, 2003)



- Step 2: use these pairs of document-scores to learn a linear transformation from C_1 scores to *CSI* scores

Results Merging
SSL (Si and Callan, 2003)

- Step 2: use these pairs of document scores to learn a linear transformation from C_1 to *CSI* scores

- Standard linear regression (query and collection specific)

$$\mathcal{S}_C(q,d) = a \times \mathcal{S}_i(q,d) + b$$

$$\arg\min_{a,b} \sum_{d} \Big(\big(f(a,b,\mathcal{S}_i(q,d)) - \mathcal{S}_C(q,d) \big) \Big)^2$$

overlap documents
(query and collection specific)

Federated Search Summary

- QBS produces effective collection representations
  - ~500 docs are enough, doesn't require cooperation
- Small document models > large document models
  - But, both assume an effective retrieval
- Query-based methods avoid this by modeling the expected retrieval using previous retrievals
  - But, require training data. or, Do they?
- Centralized sample index scores are "resource-general"
  - learn a regression model to re-score and merge

Cross-Lingual Information Retrieval

Cross-Lingual IR

- Goal: issue the query in language E and retrieve documents in languages C, F, G, and S.
  - maintained in separate indexes.
- Approach: process the query it can retrieval relevant documents in C, F, G, and S.

(1) Query Translation

- Advantages
  - lots of web-based APIs for language translation
  - the Microsoft API supports 50+ languages
- Disadvantages
  - requires sending the query to an on-line service
  - translation requires disambiguation
  - queries are terse; almost no context to support disambiguation

(2) Dictionary-based Approaches

- Goal: “panama papers” ➔ “papeles de panama”
- Bilingual Dictionary: senses + synonyms in target language:
  - material used to write on (papeles)
  - newspaper (periodicos, diarios)
  - assignment or examination (papeles)
  - document providing identity (papeles, documentos)
  - publication (publication)
- Approach: consider all combinations and keep the one with the highest co-occurrence.

Mutual Information

$$MI(w_1, w_2) = \log\left(\frac{P(w_1, w_2)}{P(w_1)P(w_2)}\right)$$

- $P(w_1, w_2)$: probability that words w_1 and w_2 both appear in a text
- $P(w_1)$: probability that word w_1 appears in a text, with or without w_2
- $P(w_2)$: probability that word w_2 appears in a text, with or without w_1
- The definition of "a text" is up to you (e.g., a sentence, a paragraph, a document)

Mutual Information

$$MI(w_1, w_2) = \log\left(\frac{P(w_1, w_2)}{P(w_1)P(w_2)}\right)$$

- If $P(w_1, w_2) = P(w_1)\,P(w_2)$, it means that the words are independent: knowing that one appears conveys no information that the other one appears
- If $P(w_1, w_2) > P(w_1)\,P(w_2)$, it means that the words are not independent: knowing that one appears makes it more probable that the other one appears

Mutual Information

estimation (using documents as units of analysis)

	word w_1 appears	word w_1 does not appear
word w_2 appears	a	b
word w_2 does not appear	c	d

every document falls under one of these quadrants

total # of documents
N = a + b + c +d

$P(w_1, w_2) = ?$

$P(w_1) = ?$

$P(w_2) = ?$

Mutual Information

estimation (using documents as units of analysis)

	word w_1 appears	word w_1 does not appear
word w_2 appears	a	b
word w_2 does not appear	c	d

every document falls under one of these quadrants

total # of documents
$N = a + b + c + d$

$P(w_1, w_2) = a / N$

$P(w_1) = (a + c) / N$

$P(w_2) = (a + b) / N$

(3) Parallel Corpora-based Approaches

- Goal: “panama papers” ➔ “papeles de panama”
- Parallel corpora: collections of corresponding documents in different languages
- Approach: (1) run the query against query-language collection, (2) find the target-language correspondents of the top results, (3) add terms from those target-language documents to the query (high TF.IDF terms)

(3) Parallel Corpora-based Approaches

Visit the main page

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

In other projects
Wikimedia Commons
Wikiquote

Not logged in Talk Contributions Create account Log in

Article Talk | Read Edit View history | Search

Panama Papers

From Wikipedia, the free encyclopedia

The **Panama Papers** are 11.5 million leaked documents that detail financial and attorney–client information for more than 214,488 offshore entities.[1] The leaked documents were created by Panamanian law firm and corporate service provider Mossack Fonseca;[2] some dated back to the 1970s.[3]

The leaked documents illustrate how wealthy individuals, including public officials, are able to keep personal financial information private.[4] While the use of offshore business entities is often not illegal, reporters found that some of the shell corporations were used for illegal purposes, including fraud, kleptocracy, tax evasion, and evading international sanctions.[5]

"John Doe", who leaked the documents to German newspaper *Süddeutsche Zeitung* (SZ), remains anonymous, even to journalists on the investigation. "My life is in danger", he told them.[6] In a May 6 statement, John Doe cited income inequality, and said he leaked the documents "simply because I understood enough about their contents to realise the scale of the injustices they described". He added that he has never worked for any government or intelligence agency and expressed willingness to help prosecutors. After SZ verified that it did come from the Panama Papers source, ICIJ posted the full written statement on its website.[7][8]

Countries with politicians, public officials or close associates implicated in the leak on April 15, 2016

Because of the number of documents, SZ asked the International Consortium of Investigative Journalists (ICIJ) for help. Journalists from 107 media organizations in 80 countries analyzed documents detailing the operations of the law firm.[3] After more than a year of analysis, the first news stories were published on April 3, 2016, along with 149 of the documents themselves.[9] The project represents an important milestone in the use of data journalism software tools and mobile collaboration.

The documents were quickly dubbed the Panama Papers. The Panamanian government strongly objects to the name; so do other entities in Panama and elsewhere. Some media outlets covering the story have used the name "Mossack Fonseca papers."[10]

Contents [hide]

(3) Parallel Corpora-based Approaches

WIKIPEDIA
La enciclopedia libre

Portada
Portal de la comunidad
Actualidad
Cambios recientes
Páginas nuevas
Página aleatoria
Ayuda
Donaciones
Notificar un error

Imprimir/exportar
Crear un libro
Descargar como PDF
Versión para imprimir

En otros proyectos
Wikimedia Commons

Herramientas
Lo que enlaza aquí
Cambios en enlazadas
Subir archivo
Páginas especiales
Enlace permanente
Información de la página
Elemento de Wikidata
Citar esta página

Otros proyectos
Commons

En otros idiomas
Wikiquote

No has iniciado sesión Discusión Contribuciones Crear una cuenta Acceder

Artículo Discusión Leer Editar Ver historial Buscar

Panama Papers

Este artículo o sección se refiere o está relacionado con un evento reciente o actualmente en curso.
La información de este artículo puede cambiar frecuentemente. Por favor, no agregues datos especulativos y recuerda colocar referencias a fuentes fiables para dar más detalles.

Panama Papers o **papeles de Panamá**[1] es el nombre dado por los medios de comunicación a una filtración informativa de documentos confidenciales de la firma de abogados panameña Mossack Fonseca, a través de una entrega de 2,6 terabytes de información por parte de una fuente no identificada al periódico alemán *Süddeutsche Zeitung*,[2] que posteriormente compartió con el Consorcio Internacional de Periodistas de Investigación (ICIJ, por sus iniciales en inglés), revelando el ocultamiento de propiedades de empresas, activos, ganancias y evasión tributaria de jefes de Estado y de gobierno, líderes de la política mundial, personas políticamente expuestas y personalidades de las finanzas, negocios, deportes y arte.[3]

Los implicados contrataban con el bufete de abogados consultores de empresas, Mossack Fonseca, servicios consistentes en fundar y establecer compañías inscritas en un paraíso fiscal de modo tal que cumpliesen con el objetivo primario de «ocultar la identidad de los propietarios».[2]

Los primeros resultados de la investigación periodística fueron presentados simultáneamente el 3 de abril de 2016 por 109 medios de comunicación (periódicos, canales de televisión y plataformas digitales noticiosas) en 76 países. El 9 de mayo de 2016 el ICIJ publicó la base de datos completa, que funciona bajo licencia *Open Database License* (ODbL, v1.0) y sus contenidos fueron liberados bajo licencia Creative Commons Atribución-CompartirIgual 3.0 Unported (CC BY-SA 3.0).[4] [5] [6]

Índice [ocultar]
1 Fuentes y trabajo de investigación periodística
1.1 Manifiesto de *John Doe*
1.2 Datos de la fuente primaria y su procesamiento
2 Trasfondo
3 Discusión sobre privacidad versus publicación de información de interés público
4 Personas implicadas
5 Bancos involucrados
6 Repercusiones
6.1 África
6.1 Client services

PANAMA PAPERS
Tipografía usada por el diario *Süddeutsche Zeitung* desde la primera publicación.

Países con políticos implicados directa o indirectamente en los *Panama Papers*.

Outline

Information Retrieval

Search Engine Components

Document Representation

Retrieval Models

Evaluation

Federated Search and Cross-lingual IR

Open-source Toolkits

Lucene

- Developed by the Apache Foundation
- Witten in Java
- Support: boolean operators, fielded search, stemming, different retrieval algorithms (e.g., VSM, QLM), simple federation (retrieval and merging), simultaneous update and retrieval
- Extensions: SOLR, Elasticsearch

https://lucene.apache.org/core/features.html

Lemur

- Developed by CMU and UMass
- Witten in C++
- Support: boolean operators and more complex query language than Lucene, fielded search, stemming, different retrieval algorithms (e.g., VSM, QLM), more complex federation (retrieval and merging), simultaneous update and retrieval

http://www.lemurproject.org/

Terrier

- Developed by University of Glasgow
- Witten in Java
- Support: extended boolean operators, stemming, different retrieval algorithms (e.g., VSM, QLM, DFR), simultaneous update and retrieval, web-based interface

http://terrier.org/