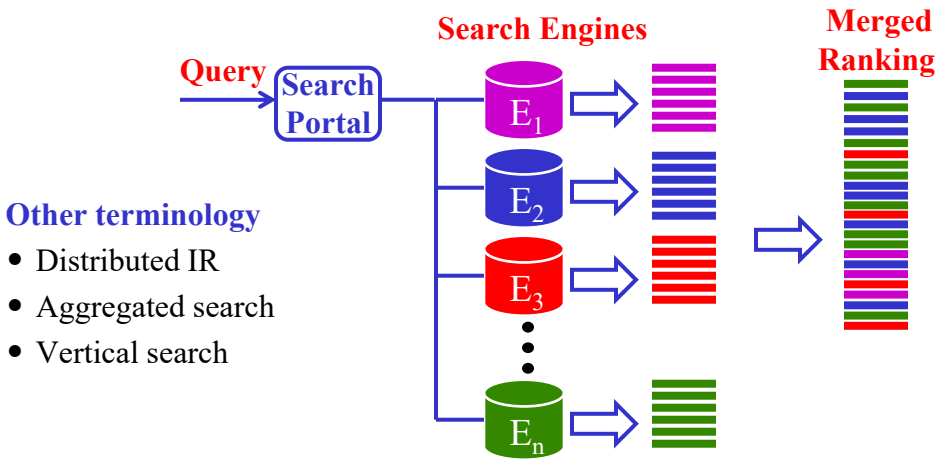
11-442 / 11-642 / 11-742:
Search Engines

Federated Search

Jamie Callan
Carnegie Mellon University
callan@cs.cmu.edu

1

What is Federated Search?



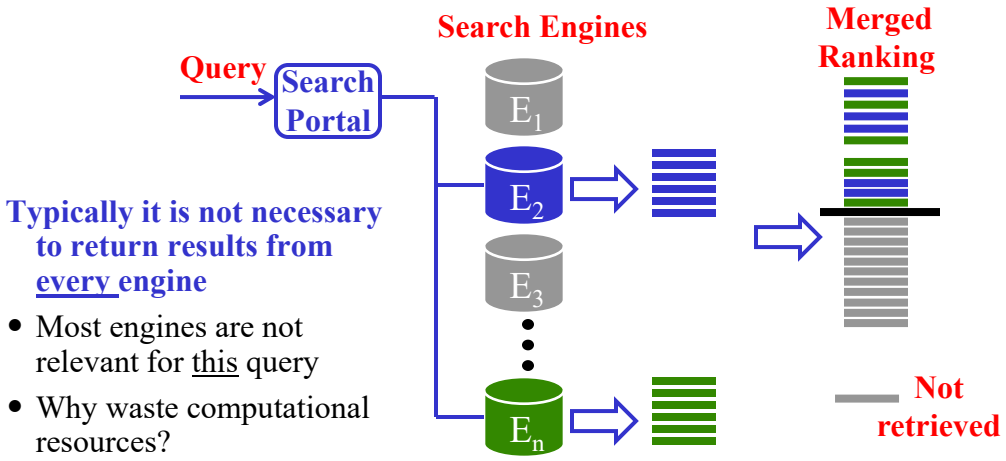
Other terminology

- Distributed IR
- Aggregated search
- Vertical search

2

2

What is Federated Search?



Typically it is not necessary to return results from every engine

- Most engines are not relevant for this query
- Why waste computational resources?

What is Federated Search? Multiple Retrieval Methods

Search portals can have different strategies for handling different types of requests

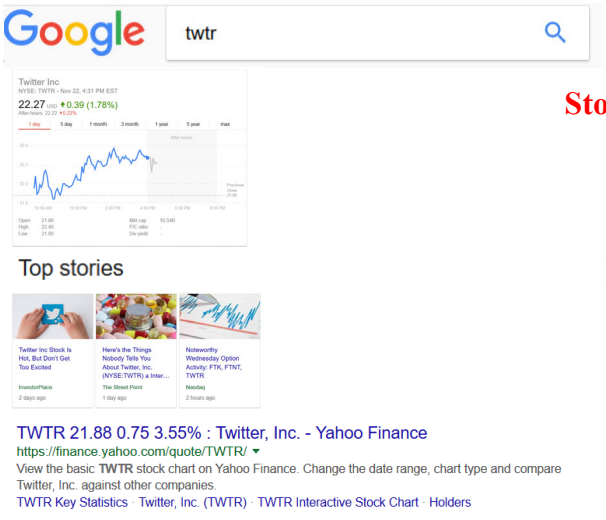
- **Search unstructured data**
  - Send the query to search engines specialized for different content
    - » Autos, music, images, videos, ...
- **Search structured data (databases)**
  - E.g., zip codes, stock symbols, …
- **Invoke a service or process**
  - E.g., calculator, stock prices, flight tracking, …

What is Federated Search?
Multiple Retrieval Methods

5

5

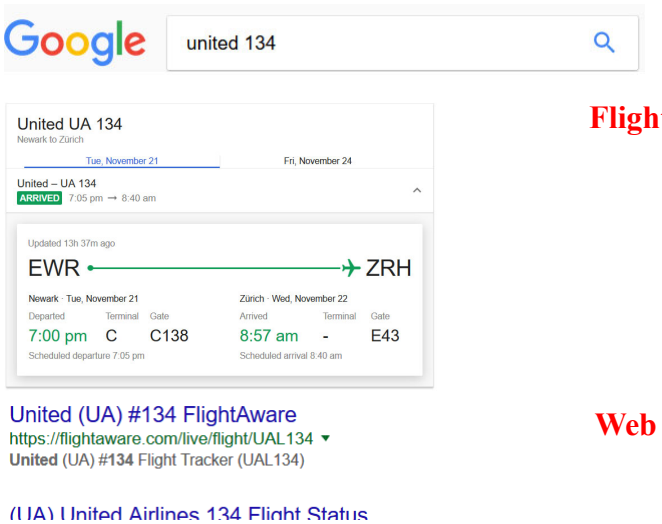
What is Federated Search?
Multiple Retrieval Methods



6

6

What is Federated Search?
Multiple Retrieval Methods



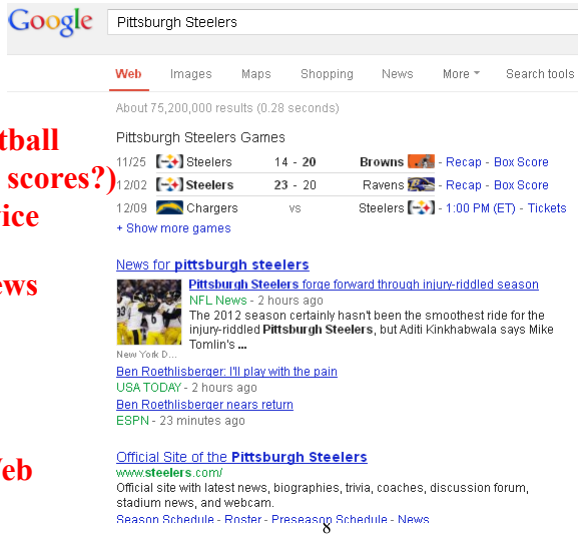
Flight status

Web search

7

7

Vertical Search:
Federated Search in Web Search Engines



A football (or sports scores?) service

News

Web

8

8

What is Federated Search?
Multiple Retrieval Methods

For simplicity, we consider everything a retrieval method

- E.g., a calculator "retrieves" the answer to a calculation query

Web search services have many retrieval methods

- At least a few dozen … maybe many more

Big enterprise systems may also have multiple retrieval methods

We won't worry about what the retrieval methods are

- Use your experience and imagination
- Assume that they change constantly

9

9

What is Federated Search?
Important Constraints

Uncooperative environment

- No special support for federated search
- Resources are not trusted

Cooperative environment

- Resources support common protocols / APIs
- Resources are trusted to provide accurate information

Different environments require different types of solutions

10

10

Components of a Federated Search System

Resource representation

- Gathering information about each resource

Offline (indexing)

Resource selection

- Selecting a set of resources for a particular query

Result merging

- Combining results from several resources into a single ranking
  - Can be an easy problem or a hard problem, depending upon the types of resources
  - Not covered today due to lack of time

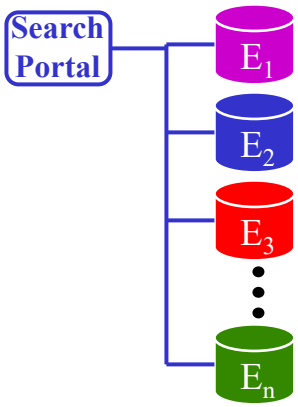
At query time

Outline

- Introduction
- **Unsupervised approaches**
  - Resource representation
  - Resource selection (CORI, ReDDE)
  - Evaluation
- Supervised approaches

Federated Search Components: Resource Representation

Search Engines



What information needs do each of these engines satisfy?

- **Often expressed as "What does each engine contain"?**
- **Cooperative environment**
  - The engine tells you whatever you want to know
- **Uncooperative environment**
  - The engine provides no special services

Defining the Resource Representation

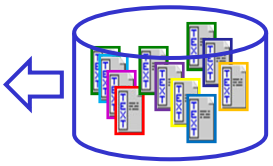
How should a resource's contents be represented?

- **Bag of words:** terms and frequencies
- **Sample queries:** Queries that this resource is good for
- **Sample documents:** Typical documents from this resource

Different representations support different types of solutions

Defining the Resource Representation

A Bag of Words Representation



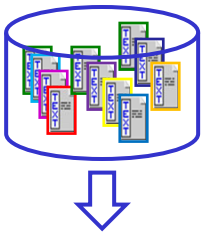
: : : :	: : :
stock	46704
stockad	5
stockard	3
stockbridg	2
stockbrok	351
stockbrokag	1
stockbrokerag	101
stockdal	8
stockhold	970
: : : :	: : :

(Porter stemming)

15

15

Defining the Resource Representation



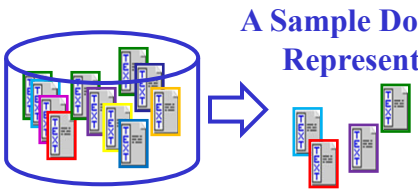
: : : : :
"clash of the titans"
"titanic 3D"
"hunger games"
"george clooney"
: : : : :

A Sample Queries Representation

16

16

Defining the Resource Representation



17

17

Acquiring the Resource Representation

How is information about the resource contents acquired?

- **Request from the resource via a protocol**
  - E.g., STARTS [Gravano, et al., 1997]
- **Request relevance assessments for a query log**
- **Query-based sampling:** Submit a query, see what comes back

All are used, but we only cover query-based sampling today

18

18

Acquiring a Resource Representation: Query-Based Sampling

The search engine is assumed to be uncooperative

- Maybe operated by an unaffiliated organization
- Maybe just a search engine that doesn't support a protocol

Procedure

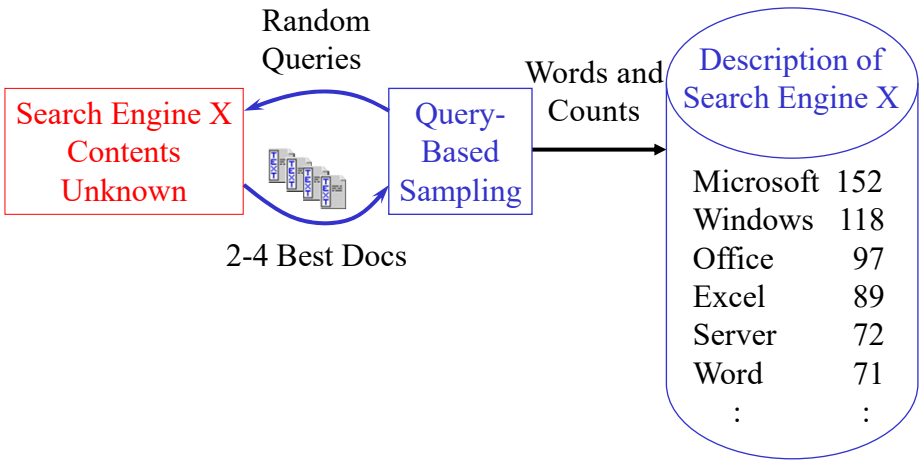
- Pick an initial query (somehow)
- Repeat N times (e.g., N=100)
  - Submit a query to the search engine
  - Download a few result documents (e.g., 2-4)
  - Update the engine's representation (words and frequencies)
  - Select query term(s) randomly from the representation

(Callan, et al., 1999)

19

19

Acquiring a Resource Selection: Query-Based Sampling



(Callan, et al., 1999)

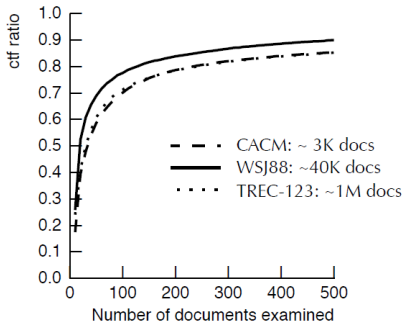
20

20

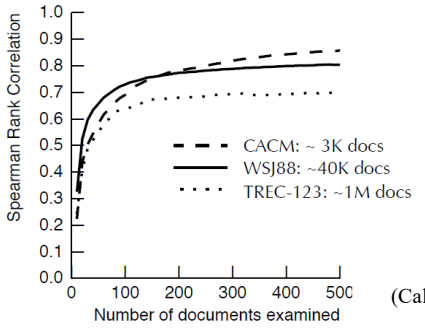
Acquiring a Resource Selection: Query-Based Sampling

A small sample of documents finds the common vocabulary terms

- E.g., 300-500 documents
- Characteristic of this corpus



If the vocabulary is sorted by frequency, the sample order is similar to the actual order



(Callan, et al., 1999)

21

21

Query-Based Sampling: Top Terms for 6 Document Collections

C10	
israeli	1,394
palestinians	1,130
israel	1,060
arab	1,009
army	985

C35	
study	779
gene	772
dna	759
human	735
cell	703

C23	
tax	1,582
budget	1,356
billion	918
house	708
bush	692

C66	
fair	3,121
cloudy	2,127
rain	1,056
snow	991
new	968

C13	
court	2,071
law	815
federal	720
judge	622
case	620

C50	
systems	622
system	490
software	463
computer	343
information	336

(Arguello, 2010)

22

22

Acquiring a Resource Selection: Query-Based Sampling

Why does it work?

- Remember Heaps' Law

What if the first query is "bad"?

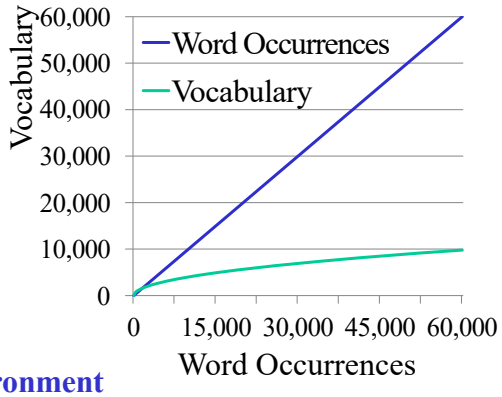
- E.g., "car" in a medical corpus

What is the effect of sample size?

- E.g., number of documents per query?

Random sampling in a cooperative environment

- Only a little better than query-based sampling (!)



23

23

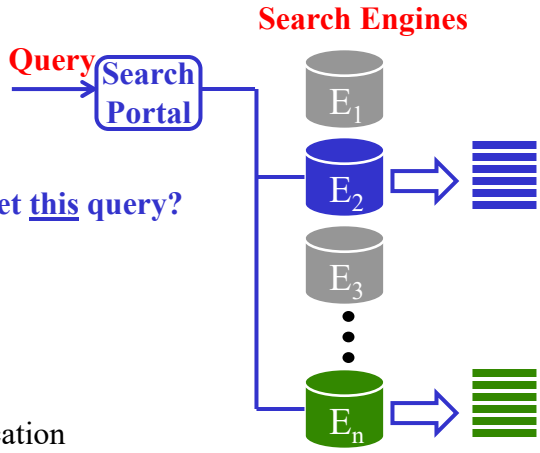
Outline

- Introduction
- **Unsupervised approaches**
  - Resource representation
  - Resource selection (CORI, ReDDE)
  - Evaluation
- Supervised approaches

24

24

Federated Search Components: Resource Selection



Which engines should get this query?

- **Unsupervised**
  - Usually ranking
    - » E.g., $P(r_i|q)$
- **Supervised**
  - Ranking or classification

Unsupervised Resource Selection

Task: Given a query q, decide which resources to search

Unsupervised methods treat this as a resource ranking problem

- Estimate $p(r_i|q)$
  - r_i: The i^{th} resource
- Select (search) the top k resources
  - Typically k is given
  - Setting k dynamically is an open research problem

Unsupervised Resource Selection

There are two main approaches

- **Content-based methods**
  - Rank resources based on the similarity of the query to the content contained in the resource
  - Different approaches are distinguished by
    - » Representation type: bag of words vs. sampled documents
    - » Ranking algorithm: many choices (e.g., CORI)
- **Query-based methods**
  - Rank resources based on the similarity of the query to past queries that the resource matched well
  - Not used much until recently due to lack of good query logs

Unsupervised Resource Selection: CORI

CORI adapts BM25 to resource ranking

- **Model each resource by a bag of words**
  - A "big document"
- **Rank resources by**

$$P(q_i \mid R_j) = \frac{df}{df + 50 + 150 * \frac{coll\_length_j}{avg\_coll\_length}} * \frac{\log\left(\frac{C + 0.5}{cf_i}\right)}{\log(C + 1.0)}$$

Inverse Collection Frequency (normalized to 1)

df : Number of documents in R_j containing q_i

cf : Number of resources containing q_i

C : Number of resources

(Callan, et al., 1995)

Unsupervised Resource Selection: Other Term-Based Algorithms

You could use almost any ranking algorithm instead

- E.g., vector space, Kullback-Leibler Divergence, query likelihood, …

The main idea

- Treat each resource as a (very large) bag of words
- Store only vocabulary and frequency information
  - Term positions are not recorded – why?

29

29

Unsupervised Resource Selection: Evaluation

What is the desired ("gold standard") order of resources?

- Order by number of relevant documents the resource contains?
  - Most common choice
- Order by the number of relevant documents the resource returns?
  - Some resources may have bad search engines
- …

30

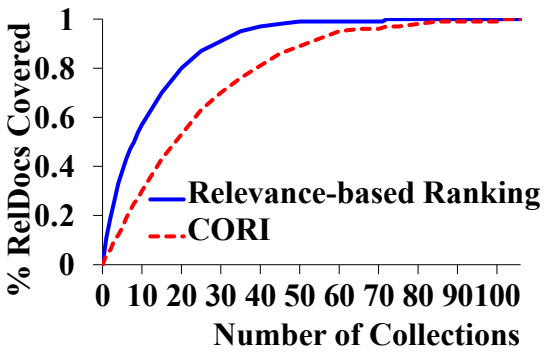
30

Unsupervised Resource Selection: Evaluation

Analysis of rankings (1 query)

Rank	Score	Actual # RelDocs
1	0.571	153
2	0.569	34
3	0.563	77
4	0.407	0
5	0.404	1
6	0.401	4
7	0.399	1
8	0.399	0
9	0.341	0

Relevance-Based Ranking (RBR) curves



31

31

Unsupervised Resource Selection: Sample Documents Methods

Bag of words methods (e.g., CORI) select resources that are similar to the query

- This really isn't the goal
- These methods favor resources that have high $p(q_i|R_j)$
  - Often that means homogeneous (often small) resources
  - This is not necessarily what we want

The goal is to select resources that return more relevant documents for this query

- Sampled documents methods address this goal more directly

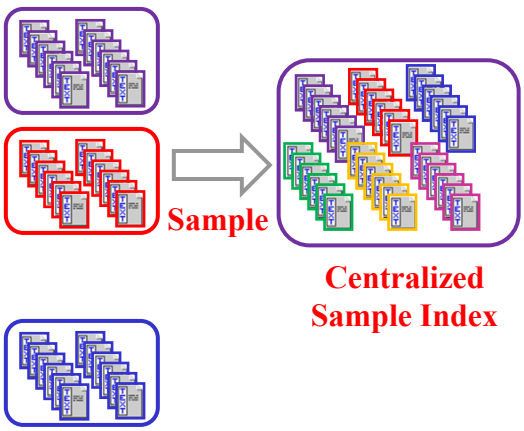
32

32

Unsupervised Resource Selection: ReDDE

Combine samples in a centralized index

- Keep track of which resource supplied each document

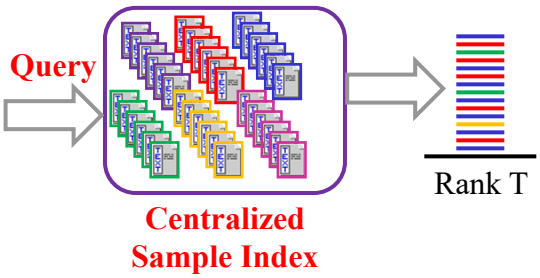


(Si and Callan, 2003)

Unsupervised Resource Selection: ReDDE

Given a query

- Search the centralized sample index
- Consider all documents above rank T to be relevant
- Examine which resources supplied the relevant documents
- Estimate the number of relevant documents in each resource

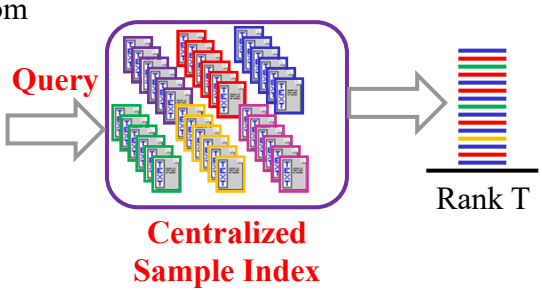


(Si and Callan, 2003)

Unsupervised Resource Selection: ReDDE

Estimating the number of relevant documents in the i'th resource

- Count the number of documents from resource i above threshold T
- Multiply this value by resource_size / sample_size



(Si and Callan, 2003)

Unsupervised Resource Selection: Voting Methods

ReDDE can be viewed as a sample-based voting method

- Each top-ranked document votes for its collection

ReDDE can be generalized to create resource selection algorithms that satisfy explicit optimization goals

- **High Recall:** Select resources that contain many relevant documents
- **High Precision:** Select resources that return many relevant documents at the top of a merged set of results

Framing resource selection in terms of optimization goals allows development of more sophisticated resource selection algorithms

– E.g., Returned Utility Maximization

(Si and Callan, 2004)

Unsupervised Resource Selection: Voting Methods

ReDDE can be viewed as a sample-based voting method

- Each top-ranked document votes for its collection

There are many variants of the algorithm

- Samples from more reliable resources get more votes
- Samples that are more relevant get more votes
- …

No variant outperforms others on all experimental testbeds

37

37

ReDDE vs. CORI

ReDDE is a little more accurate

- It never does worse; often it does better

ReDDE outperforms CORI when the distribution of collection sizes is skewed

- CORI is biased towards small collections
  - They are more likely to be homogeneous
  - It misses large, heterogeneous collections
- ReDDE has a (weaker) bias towards large collections

CORI is more efficient than ReDDE

- 1 "document" per resource vs. many documents per resource

38

38

ReDDE vs. CORI: Efficiency

What is resource selection cost for the query 'apple'?

- Assume v verticals
- Assume s sampled documents per vertical
  - E.g., s=300

CORI: Maximum inverted list length is v

- One posting per vertical that contains 'apple'
- A count of how many documents in v_i contain 'apple'

ReDDE: Maximum inverted list length is $s \times v$

- One posting per sampled document that contains 'apple'
- Sample documents from each v_i that contain apple

Unsupervised Resource Selection: Bag of Words Methods

Characteristics of bag of words ("large documents") approaches

- Large documents dominate a resource representation
- Favors resources with a larger proportion of relevant content
  - i.e., small or homogeneous resources

Bag of words resource selection is surprisingly effective

- The state-of-the-art from 1994 until about 2003
- Very efficient
- Still very competitive

Outline

- Introduction
- Unsupervised approaches
  - Resource representation
  - Resource selection (CORI, ReDDE)
  - Evaluation
- **Supervised approaches**

Resource Selection: Learning to Rank Resources (L2R)

A standard learning-to-rank architecture can be applied to resource ranking

- Algorithm: SVM^{Rank}
- Standard pairwise training

  $h(\boldsymbol{\Phi}(q, s_i)) > h(\boldsymbol{\Phi}(q, s_j))$

  h: learned model

  $\boldsymbol{\Phi}$: feature generator

  – Search engine s_i should be ranked higher for query q than search engine s_j

  – E.g., $h(\boldsymbol{\Phi}(\text{“iron man”, imdb.com})) > h(\boldsymbol{\Phi}(\text{“iron man”, pubmed.gov}))$

(Dai, et al., 2017)

Resource Selection: Learning to Rank Resources (L2R)

Example features

- **Query-independent information**
  - $p(s_i)$: popularity of s_i
    - » The percentage of queries in a search log that went to search engine s_i

(Dai, et al., 2017)

Resource Selection: Learning to Rank Resources (L2R)

Example features

- **Term-based statistics**
  - Two features derived from the Taily resource selection algorithm, $f(q, s_i)$
    - » Inverse rank
    - » Binned rank (bins of size 10)
      - Ignore differences between ranks 4 and 5, but not 4 and 14
  - Champion list features: Top k documents contributed by each shard for term t
  - Query likelihood of the query with the shard language model
  - Query term statistics: $max_{t \in q} ctf(t, s_i)$ and $min_{t \in q} ctf(t, s_i)$
  - Bigram log frequency: $\sum_{b \in q} log\big(ctf(b, s_i)\big)$ for bigrams with ctf > 50

(Dai, et al., 2017)

Resource Selection: Learning to Rank Resources (L2R)

Example features (continued)

- **Sample-document (CSI features)**
  - – ReDDE and Rank-S scores, inverse ranks, binned ranks (bins of 10)
  - – Average distance to the shard centroid
    - » Distance of the top-k CSI documents to their shards' centroids using cosine & KLD
    - » Are these documents representative of their shards?

(Dai, et al., 2017)

Resource Selection: Learning to Rank Resources (L2R)

Datasets

- **ClueWeb09-B:** 50 million web pages clustered into 123 shards
  - – Clustering produces topic-oriented index shards
  - – Unusual application: Use federated search to improve distributed search efficiency
- **Gov2:** 25 million web pages clustered into 199 shards
  - – Same comments

Parameters

- Centralized sample index: 1% sample
- Search engine: Indri with SDM queries

(Dai, et al., 2017)

Resource Selection: Learning to Rank Resources (L2R)

How well does it work?

Method	CW09-B (123 shards)						Gov2 (199 shards)					
	T=4			T=8			T=6			T=12		
	P@10	NDCG@30	MAP@1000	P@10	NDCG@30	MAP@1000	P@10	NDCG@30	MAP@1000	P@10	NDCG@30	MAP@1000
Redde	0.355	0.262	0.176	0.363*	0.275*	0.187	0.580*	0.445	0.267	0.587*	0.4600*	0.289
Rank-S	0.350	0.259	0.175	0.360*	0.268	0.183	0.570	0.440	0.263	0.585*	0.461*	0.286
Taily	0.346	0.260	0.172	0.346	0.260	0.175	0.518	0.403	0.235	0.530	0.418	0.256
Jnt	0.370*	0.269	0.178	0.367*	0.277*	0.192	0.582*	0.459	0.278	0.588*	0.465*	0.292
L2R-TREC	0.374*	0.281*	0.192▲	0.377*	0.286▲*	0.202▲*	0.593*	0.469*	0.299▲	0.591*	0.475▲*	0.313▲*
L2R-AOL	0.374*	0.281▲*	0.191▲	0.375*	0.287▲*	0.202▲*	0.593*	0.470▲*	0.291▲	0.587*	0.470*	0.307▲*
L2R-MQT	0.382*	0.285▲*	0.193▲	0.375*	0.286▲*	0.202▲*	0.586*	0.465*	0.292▲	0.593*	0.474▲*	0.309▲*
Exh	0.372	0.288	0.208	0.372	0.288	0.208	0.585	0.479	0.315	0.585	0.479	0.315

TREC, -AOL, and –MQT are different types of training data

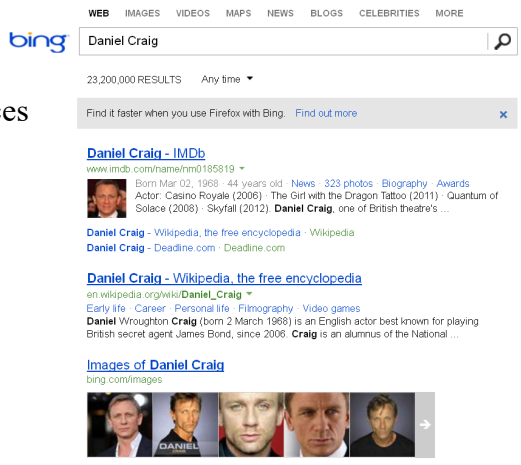
T=4, 6, 8, 12 is how many top-ranked index shards (search engines) are searched

(Dai, et al., 2017)

Vertical Search: Federated Search in Web Search Engines

How would these ideas be applied in a web search environment?

- "Federated search" → "vertical search"
  - Different people use different terminology
- Typically a more diverse set of information services
  - Some aren't search engines
  - Some don't return text
    - » E.g., maps, images, …
  - Some aren't topically coherent
    - » E.g., local search
- There may be much training data



Resource Selection: Query Features

Queries may contain clues about which verticals are appropriate

- "Pittsburgh weather", "United flight 1243", "washing machine repair videos", "Daniel Craig pictures", "Cajun shrimp recipes", …

Query features

- **Boolean:** keywords and regular expressions
  - E.g., "weather", "news", "videos", …
- **Geographic:** Probabilities associated with geographic entities
  - E.g., "Pittsburgh pizza"
- **Category:** query's affinity to a set of topic categories
  - E.g., "Cancun vacations"

(Arguello, et al., 2009)

Resource Selection: Corpus Features

Methods that predict the effectiveness of a query on a given corpus

- Typical resource selection algorithms
  - E.g., CORI, ReDDE, Taily
- Query difficulty prediction algorithms
  - E.g., Clarity

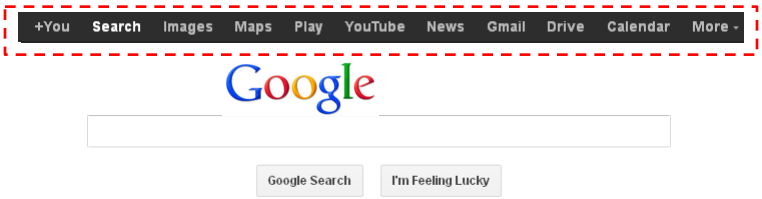
(Arguello, et al., 2009)

Resource Selection: Query Log Features

Where do query log features come from?

- Manual assessments? Clickthrough?

Some vertical search engines can be accessed directly by users



- The user believed that the engine was a good choice for the query
- The vertical search engine's query log is a good source of training data

(Arguello, et al., 2009)

Resource Selection: Query Log Features

The qlog feature

- Use the queries from the query log to form a language model for vertical v
- Use a typical query likelihood model

$$p(q|v) = \prod_{t \in q} p(t|v)$$

- $p(t|v)$ is a smoothed MLE
  - E.g., Jelinek-Mercer or Dirichlet

Query log for vertical *v*

: : : :
hotels in cancun
cheap flights
tickets to miami
chicago hotels
hotel deals
tokyo attractions
flights to los angeles
broadway tickets
: : : :

(Arguello, et al., 2009)

Resource Selection: Query Log Features

The Soft.ReDDE feature

- Use the queries from the query log to form a language model for vertical *v*
- Use the query to retrieve the top *n* documents from an external collection
  - E.g., wikipedia
  - High-quality documents related to the query
- Use a voting algorithm such as ReDDE
  - The vote of document d_i is KLD (d_i || *v*)
  - Documents that are similar to the query log have higher votes

(Arguello, et al., 2009)

Supervised Resource Selection

Supervised resource selection is similar to LTR for document ranking

- Similar architecture
  - E.g., SVM^{Rank} with pairwise or listwise training
- Different features
  - Older heuristic algorithms are some of the strongest features
  - Features based on search logs are very important

The state-of-the-art algorithms are close to exhaustive search

- Search a fraction of the available information without losing Precision
  - It is harder to maintain Recall

Outline

- **Introduction**
- **Unsupervised approaches**
  - Resource representation
  - Resource selection (CORI, ReDDE)
  - Evaluation
- **Supervised approaches**

55

Summary

Integration of diverse information resources is an increasingly important problem

Components of a distributed / federated / vertical search system

- Resource representation
- Resource selection
- Result merging (not covered)

Problem requirements that affect the type of solution

- Cooperative vs. uncooperative
- Unsupervised vs. supervised

56

Next Semester…

This course will be offered next semester

I will need Teaching Assistants

- 8-12 hours/week × 7 weeks ("grading weeks")
- 3-5 hours/week × 9 weeks (office hours, piazza)

Please send me email if you are interested in being a TA

- I will start TA interviews after grades are posted
  (probably Monday, May 10 or Tuesday, May 11)

For More Information

- J. Arguello, J. Callan, and F. Diaz. Classification-based resource selection. *CIKM 2009*. 2009.
- J. Arguello, J. Callan, F. Diaz, and J.-F. Crespo. Sources of evidence for vertical selection. *SIGIR 2009*. 2009.
- J. Callan and M. Connell. Query-based sampling of text databases. *TOIS*. 2001.
- Z. Dai, Y. Kim and J. Callan. Learning to rank resources (short paper). In Proceedings of the 40th International ACM SIGIR Conference on Research & Development in Information Retrieval. ACM. 2017.
- J. P. Callan, Z. Lu, and W.B. Croft. Searching distributed collections with inference networks. *SIGIR 1995*. 1995.
- L. Si and J. Callan. Relevant document distribution estimation method for resource selection. *SIGIR 2003*. 2003.
- L. Si and J. Callan. A semi-supervised learning method to merge search engine results. *TOIS*. 2003.