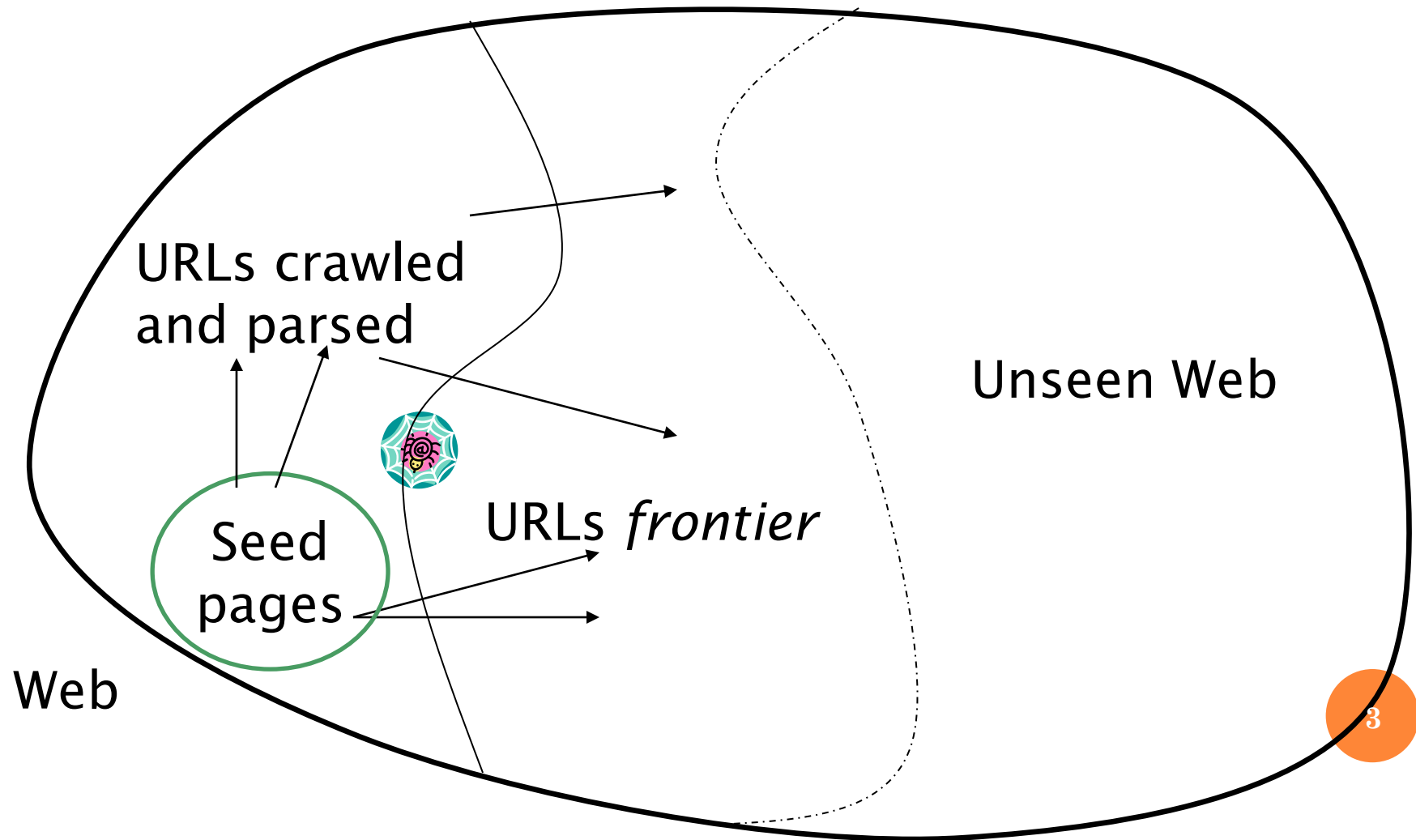


# CRAWLING AND WEB INDEXES

# BASIC CRAWLER OPERATION

- Begin with known “seed” URLs
- Fetch and parse them
  - Extract URLs they point to
  - Place the extracted URLs on a queue
- Fetch each URL on the queue and repeat

# CRAWLING PICTURE



## SIMPLE PICTURE – COMPLICATIONS

- Web crawling isn't feasible with one machine
  - All of the above steps distributed
- Malicious pages
  - Spam pages
  - Spider traps – incl dynamically generated
- Even non-malicious pages pose challenges
  - Latency/bandwidth to remote servers vary
  - Webmasters' stipulations
    - How “deep” should you crawl a site's URL hierarchy?
  - Site mirrors and duplicate pages
- Politeness – don't hit a server too often

## WHAT ANY CRAWLER *MUST* DO

- Be Polite: Respect implicit and explicit politeness considerations
  - Only crawl allowed pages
  - Respect *robots.txt* (more on this shortly)
- Be Robust: Be immune to spider traps and other malicious behavior from web servers

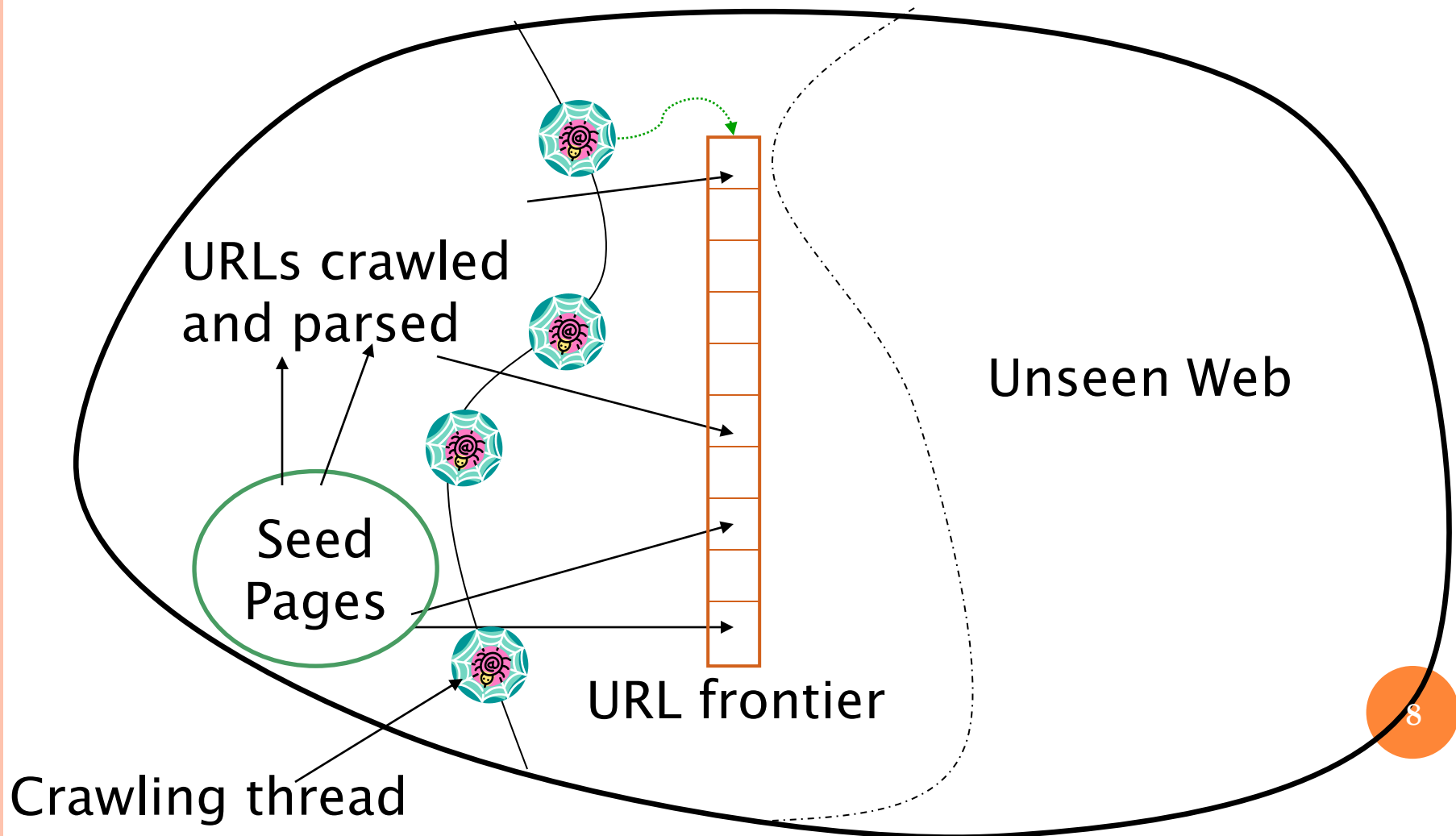
## WHAT ANY CRAWLER *SHOULD* DO

- Be capable of distributed operation: designed to run on multiple distributed machines
- Be scalable: designed to increase the crawl rate by adding more machines
- Performance/efficiency: permit full use of available processing and network resources

## WHAT ANY CRAWLER *SHOULD* DO

- Fetch pages of “higher quality” first
- Continuous operation: Continue fetching fresh copies of a previously fetched page
- Extensible: Adapt to new data formats, protocols

# UPDATED CRAWLING PICTURE





## URL FRONTIER

- Can include multiple pages from the same host
- Must avoid trying to fetch them all at the same time
- Must try to keep all crawling threads busy

## EXPLICIT AND IMPLICIT POLITENESS

- Explicit politeness: specifications from webmasters on what portions of site can be crawled
  - robots.txt
- Implicit politeness: even with no specification, avoid hitting any site too often

## ROBOTS.TXT

- Protocol for giving spiders (“robots”) limited access to a website, originally from 1994
  - [www.robotstxt.org/wc/norobots.html](http://www.robotstxt.org/wc/norobots.html)
- Website announces its request on what can(not) be crawled
  - For a server, create a file `/robots.txt`
  - This file specifies access restrictions

## ROBOTS.TXT EXAMPLE

- No robot should visit any URL starting with "/yoursite/temp/", except the robot called "searchengine":

```
User-agent: *
```

```
Disallow: /yoursite/temp/
```

```
User-agent: searchengine
```

```
Disallow:
```

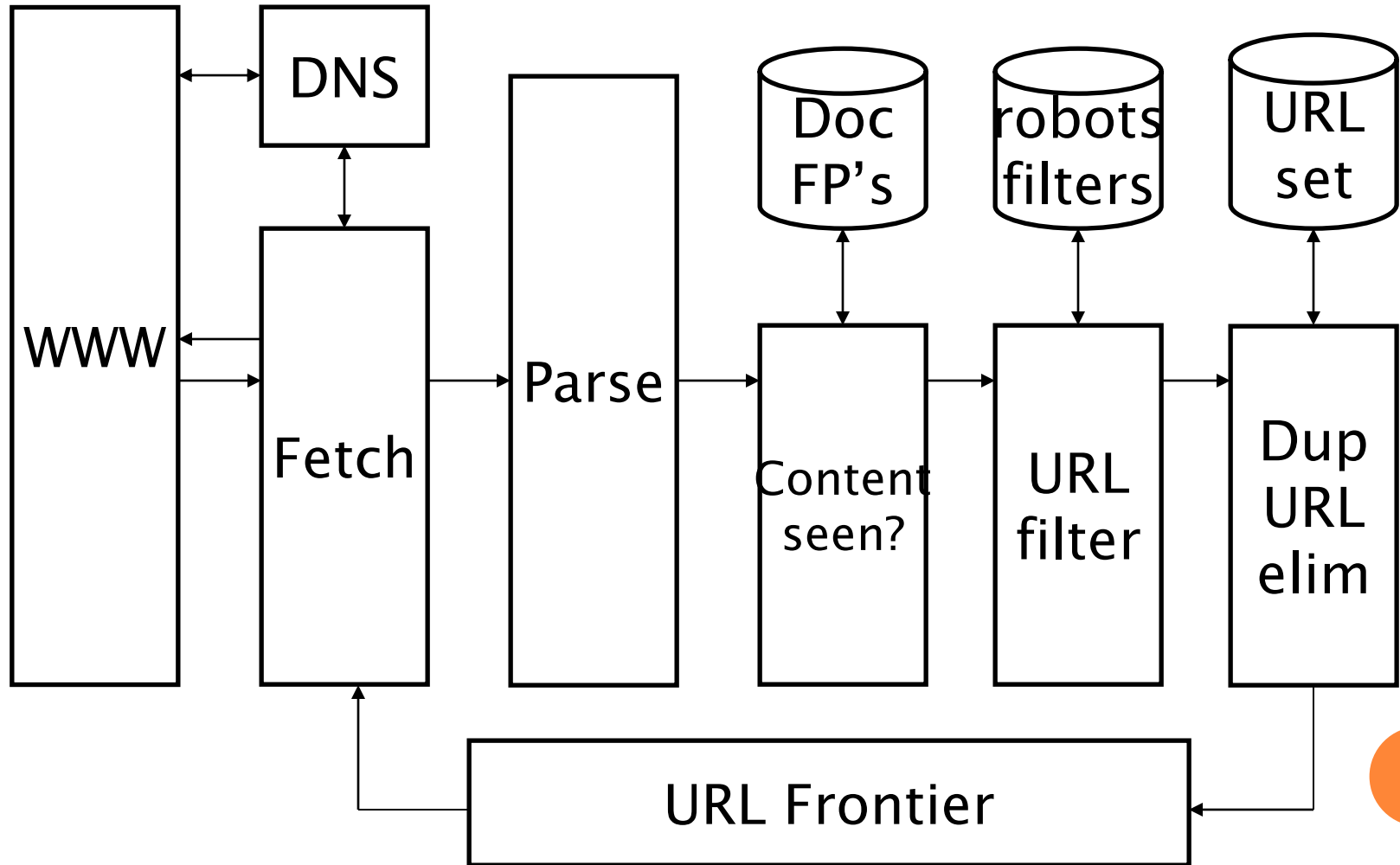
# PROCESSING STEPS IN CRAWLING

- Pick a URL from the frontier
- Fetch the document at the URL
- Parse the URL
  - Extract links from it to other docs (URLs)
- Check if URL has content already seen
  - If not, add to indexes
- For each extracted URL
  - Ensure it passes certain URL filter tests
  - Check if it is already in the frontier (duplicate URL elimination)

← Which one?

E.g., only crawl .edu, obey robots.txt, etc.

# BASIC CRAWL ARCHITECTURE



# DNS (DOMAIN NAME SERVER)

- A lookup service on the internet
  - Given a URL, retrieve its IP address
  - Service provided by a distributed set of servers – thus, lookup latencies can be high (even seconds)
- Common OS implementations of DNS lookup are *blocking*: only one outstanding request at a time
- Solutions
  - DNS caching
  - Batch DNS resolver – collects requests and sends them out together

## PARSING: URL NORMALIZATION

- When a fetched document is parsed, some of the extracted links are *relative* URLs
- E.g., [http://en.wikipedia.org/wiki/Main\\_Page](http://en.wikipedia.org/wiki/Main_Page) has a relative link to /wiki/Wikipedia:General\_disclaimer which is the same as the absolute URL [http://en.wikipedia.org/wiki/Wikipedia:General\\_disclaimer](http://en.wikipedia.org/wiki/Wikipedia:General_disclaimer)
- During parsing, must normalize (expand) such relative URLs



## CONTENT SEEN?

- Duplication is widespread on the web
- If the page just fetched is already in the index, do not further process it
- This is verified using document fingerprints or shingles

# FILTERS AND ROBOTS.TXT

- Filters – regular expressions for URL's to be crawled/not
- Once a robots.txt file is fetched from a site, need not fetch it repeatedly
  - Doing so burns bandwidth, hits web server
- Cache robots.txt files

## DUPLICATE URL ELIMINATION

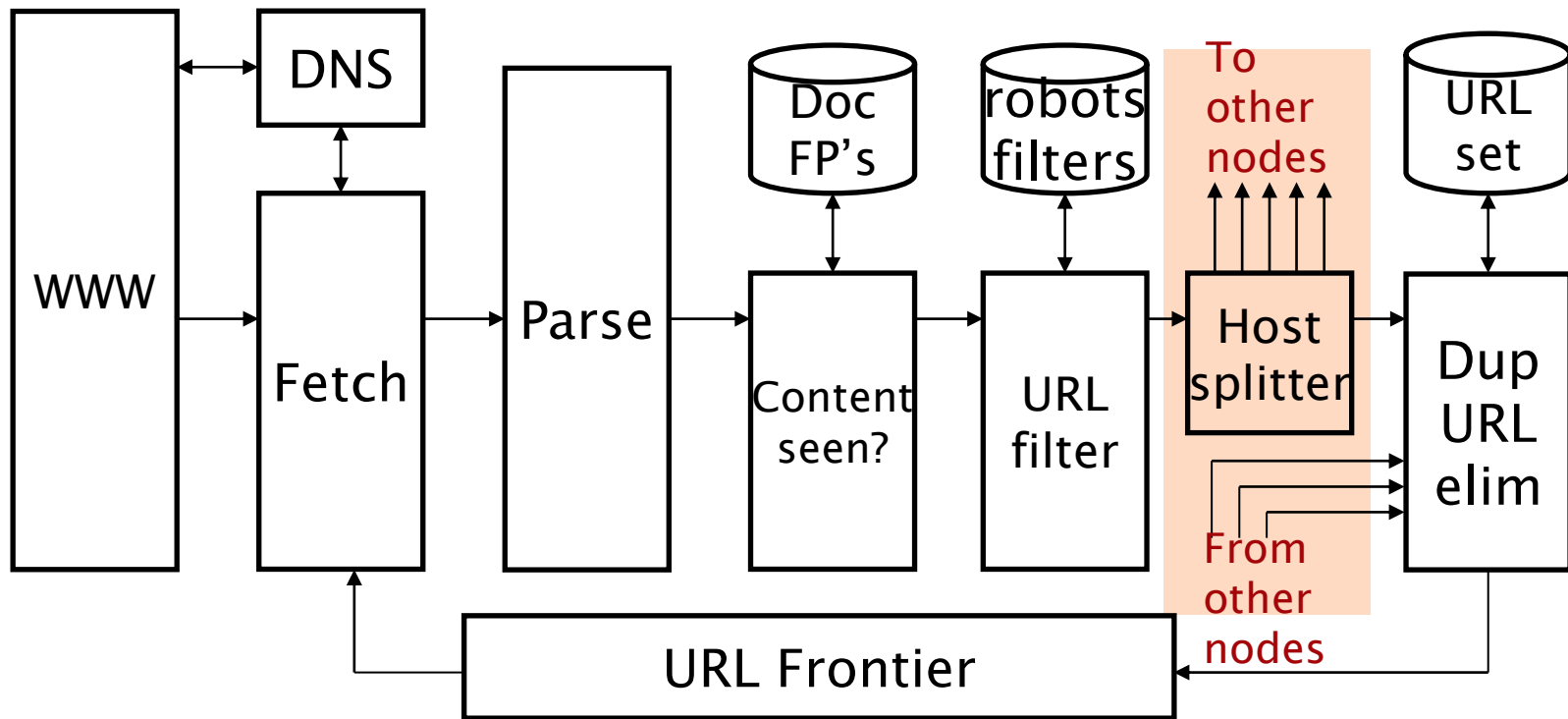
- For a non-continuous (one-shot) crawl, test to see if an extracted+filtered URL has already been passed to the frontier
- For a continuous crawl – see details of frontier implementation

# DISTRIBUTING THE CRAWLER

- Run multiple crawl threads, under different processes – potentially at different nodes
  - Geographically distributed nodes
- Partition hosts being crawled into nodes
  - Hash used for partition
- How do these nodes communicate and share URLs?

# COMMUNICATION BETWEEN NODES

- Output of the URL filter at each node is sent to the Dup URL Eliminator of the appropriate node



## URL FRONTIER: TWO MAIN CONSIDERATIONS

- Politeness: do not hit a web server too frequently
- Freshness: crawl some pages more often than others
  - E.g., pages (such as News sites) whose content changes often

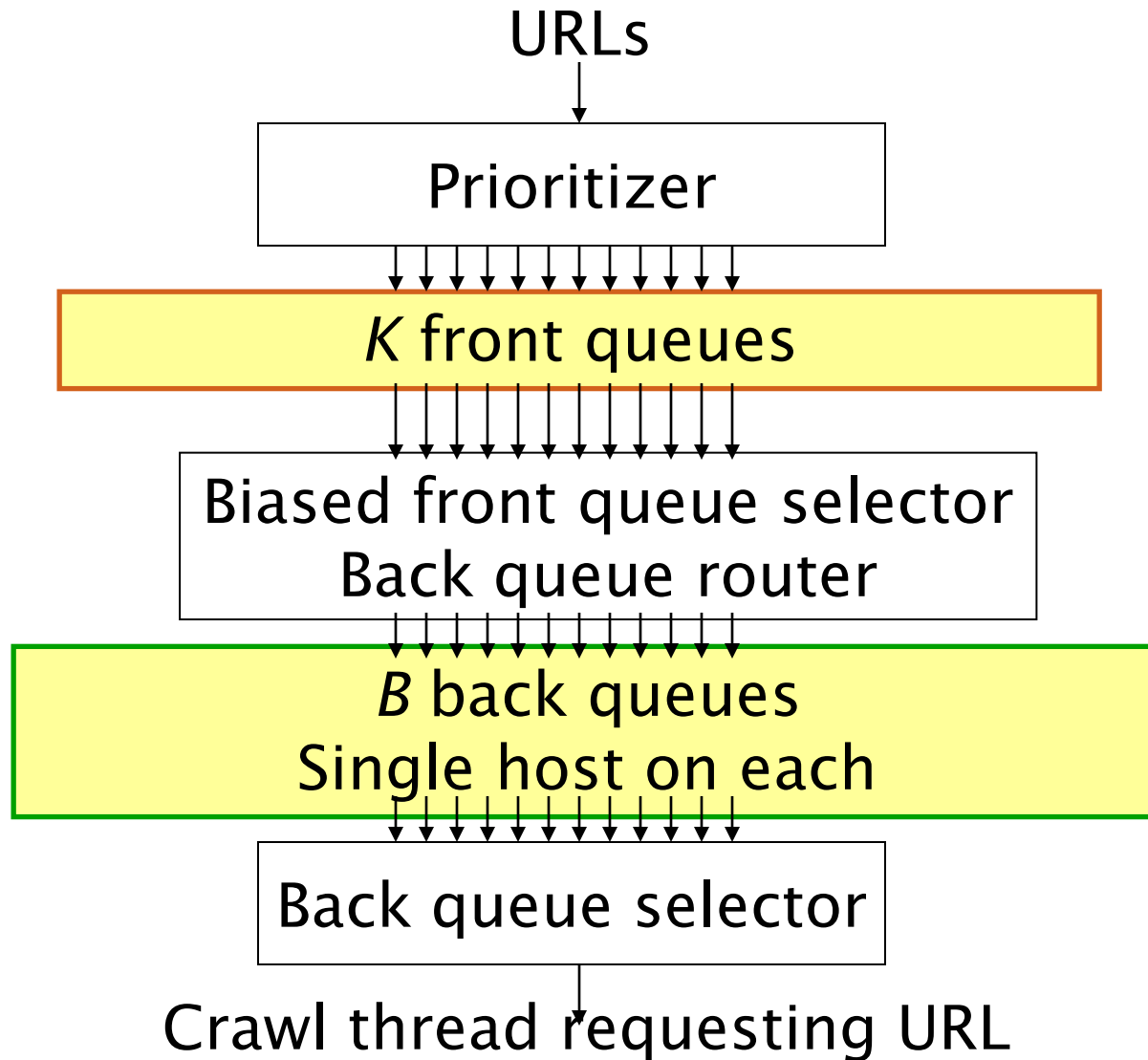
These goals may conflict each other.

(E.g., simple priority queue fails – many links out of a page go to its own site, creating a burst of accesses to that site.)

## POLITENESS – CHALLENGES

- Even if we restrict only one thread to fetch from a host, can hit it repeatedly
- Common heuristic: insert time gap between successive requests to a host that is  $\gg$  time for most recent fetch from that host

# URL FRONTIER: MERCATOR SCHEME

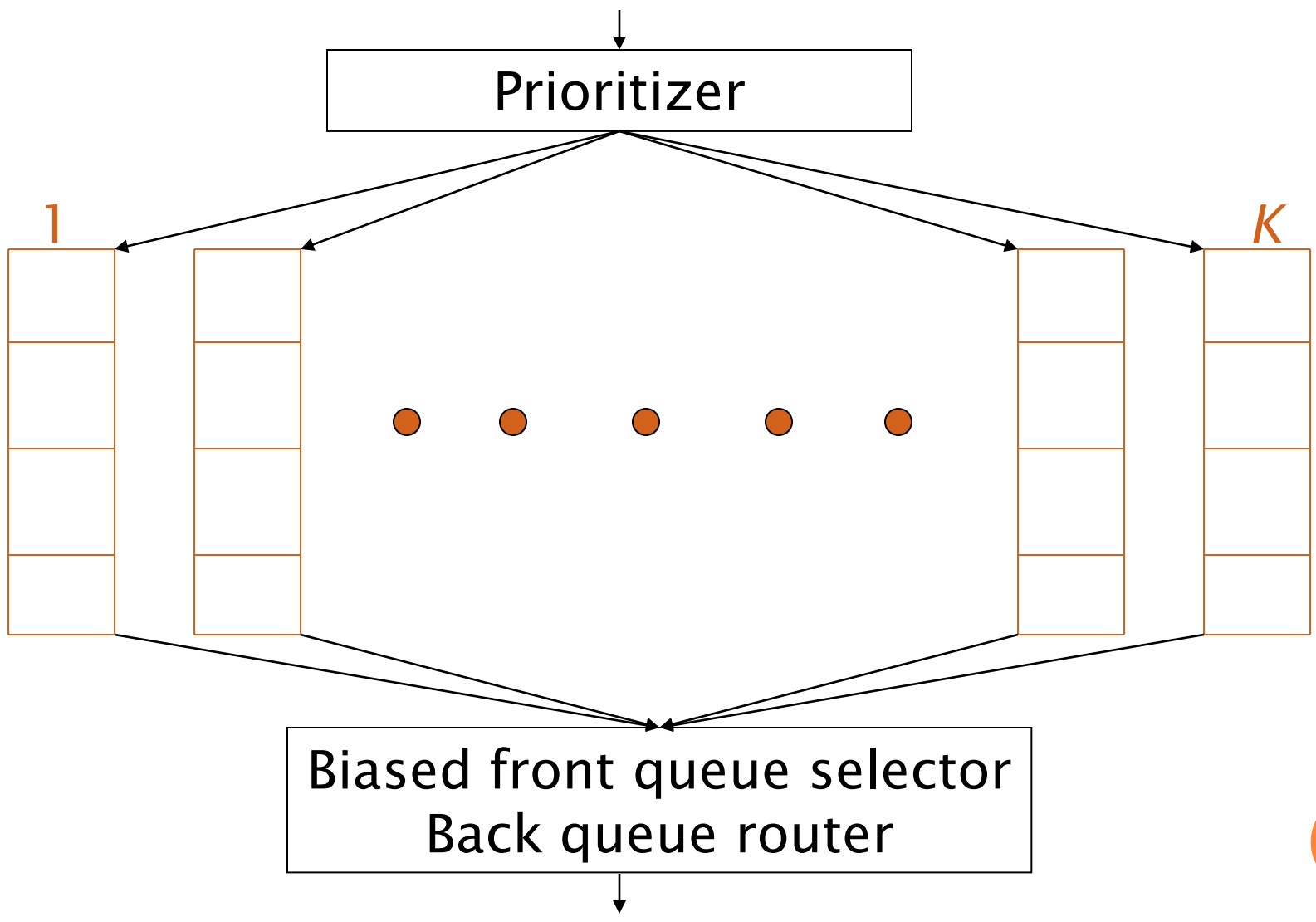




## MERCATOR URL FRONTIER

- URLs flow in from the top into the frontier
- **Front queues** manage prioritization
- **Back queues** enforce politeness
- Each queue is FIFO

# FRONT QUEUES



## FRONT QUEUES

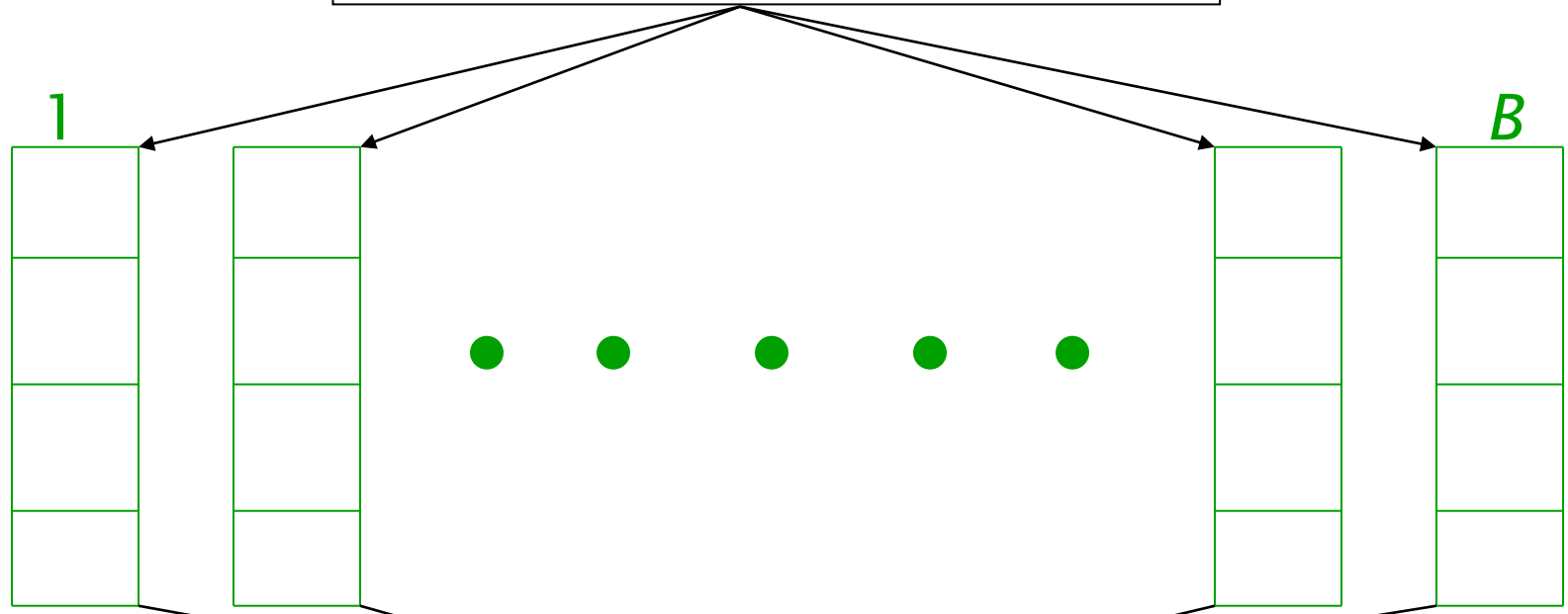
- Prioritizer assigns to URL an integer priority between 1 and  $K$ 
  - Appends URL to corresponding queue
- Heuristics for assigning priority
  - Refresh rate sampled from previous crawls
  - Application-specific (e.g., “crawl news sites more often”)

## BIASED FRONT QUEUE SELECTOR

- When a back queue requests a URL (in a sequence to be described): picks a **front queue** from which to pull a URL
- This choice can be round robin biased to queues of higher priority, or some more sophisticated variant
  - Can be randomized

# BACK QUEUES

Biased front queue selector  
Back queue router



Back queue selector



# BACK QUEUE INVARIANTS

- Each back queue is kept non-empty while the crawl is in progress
- Each back queue only contains URLs from a single host
  - Maintain a table from hosts to back queues

Host name	Back queue
...	3
	1
	<i>B</i>

## BACK QUEUE HEAP

- One entry for each back queue
- The entry is the earliest time  $t_e$  at which the host corresponding to the back queue can be hit again
- This earliest time is determined from
  - Last access to that host
  - Any time buffer heuristic we choose

## BACK QUEUE PROCESSING

- A crawler thread seeking a URL to crawl:
- Extracts the root of the heap
- Fetches URL at head of corresponding back queue  $q$  (look up from table)
- Checks if queue  $q$  is now empty – if so, pulls a URL  $v$  from front queues
  - If there's already a back queue for  $v$ 's host, append  $v$  to that back queue and pull another URL from front queues, repeat
  - Else add  $v$  to  $q$
- When  $q$  is non-empty, create heap entry for it

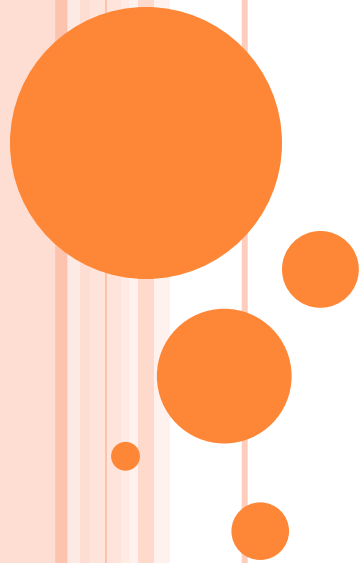


## NUMBER OF BACK QUEUES $B$

- Keep all threads busy while respecting politeness
- Mercator recommendation: three times as many back queues as crawler threads

# RESOURCES

- IIR Chapter 20
- Mercator: A scalable, extensible web crawler (Heydon et al. 1999)
- A standard for robot exclusion



# EVALUATION

# THIS LECTURE

- How do we know if our results are any good?
  - Evaluating a search engine
    - Benchmarks
    - Precision and recall
- Results summaries:
  - Making our good results usable to a user



# EVALUATING SEARCH ENGINES

37

# MEASURES FOR A SEARCH ENGINE

- How fast does it index (offline)
  - Number of documents/hour
  - (Average document size)
- How fast does it search (online)
  - Latency as a function of index size
- Expressiveness of query language
  - Ability to express complex information needs
  - Speed on complex queries
- Uncluttered UI
- Is it free?

## MEASURES FOR A SEARCH ENGINE

- All of the preceding criteria are *measurable*: we can quantify speed/size
  - we can make expressiveness precise
- The key measure: user happiness
  - What is this?
  - Speed of response/size of index are factors
  - But blindingly fast, useless answers won't make a user happy
- Need a way of quantifying user happiness

# MEASURING USER HAPPINESS

- Issue: who is the user we are trying to make happy?
  - Depends on the setting
- Web engine:
  - User finds what s/he wants and returns to the engine
    - Can measure rate of return users
  - User completes task – search as a means, not end
  - See Russell <http://dmrussell.googlepages.com/JCDL-talk-June-2007-short.pdf>
- eCommerce site: user finds what s/he wants and buys
  - Is it the end-user, or the eCommerce site, whose happiness we measure?
  - Measure time to purchase, or fraction of searchers who become buyers?



# MEASURING USER HAPPINESS

- Enterprise (company/govt/academic): Care about “user productivity”
  - How much time do my users save when looking for information?
  - Many other criteria having to do with breadth of access, secure access, etc.

# HAPPINESS: ELUSIVE TO MEASURE

- Most common proxy: *relevance* of search results
- But how do you measure relevance?
- We will detail a methodology here, then examine its issues
- Relevance measurement requires 3 elements:
  1. A benchmark document collection
  2. A benchmark suite of queries
  3. A usually binary assessment of either Relevant or Nonrelevant for each query and each document
    - Some work on more-than-binary, but not the standard

# EVALUATING AN IR SYSTEM

Because the query language may be ill-designed!

- Note: the **information need** is translated into a **query**
- Relevance is assessed relative to the **information need** *not* the **query**
- E.g., Information need: *I'm looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine.*
- Query: *wine red white heart attack effective*
- Evaluate whether the doc addresses the information need, not whether it has these words

## STANDARD RELEVANCE BENCHMARKS

- TREC - National Institute of Standards and Technology (NIST) has run a large IR test bed for many years
- Reuters and other benchmark doc collections used
- “Retrieval tasks” specified
  - sometimes as queries
- Human experts mark, for each query and for each doc, Relevant or Nonrelevant
  - or at least for subset of docs that some system returned for that query

# UNRANKED RETRIEVAL EVALUATION: PRECISION AND RECALL

- **Precision:** fraction of retrieved docs that are relevant =  $P(\text{relevant} \mid \text{retrieved})$
- **Recall:** fraction of relevant docs that are retrieved  
=  $P(\text{retrieved} \mid \text{relevant})$

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

- Precision  $P = \text{tp}/(\text{tp} + \text{fp})$
- Recall  $R = \text{tp}/(\text{tp} + \text{fn})$

# SHOULD WE INSTEAD USE THE ACCURACY MEASURE FOR EVALUATION?

- Given a query, an engine classifies each doc as “Relevant” or “Nonrelevant”
- The **accuracy** of an engine: the fraction of these classifications that are correct
  - $(tp + tn) / (tp + fp + fn + tn)$
- **Accuracy** is a commonly used evaluation measure in machine learning classification work

## WHY NOT JUST USE ACCURACY?

- How to build a 99.99% accurate search engine on a low budget....

snoogle.com

Search for:

*0 matching results found.*

- People doing information retrieval *want to find something* and have a certain tolerance for junk.

## QUIZ: SNOOGLE

- Why does Snoogle on the previous page produce 99.99% accuracy? (recall the definition of accuracy.)



# PRECISION/RECALL

- You can get high recall (but low precision) by retrieving all docs for all queries!
- Recall is a non-decreasing function of the number of docs retrieved
- In a good system, precision decreases as either the number of docs retrieved or recall increases
  - This is not a theorem, but a result with strong empirical confirmation

# DIFFICULTIES IN USING PRECISION/RECALL

- Should average over large document collection/query ensembles
- Need human relevance assessments
  - People aren't reliable assessors
- Assessments have to be binary
  - Nuanced assessments?
- Heavily skewed by collection/authorship
  - Results may not translate from one domain to another

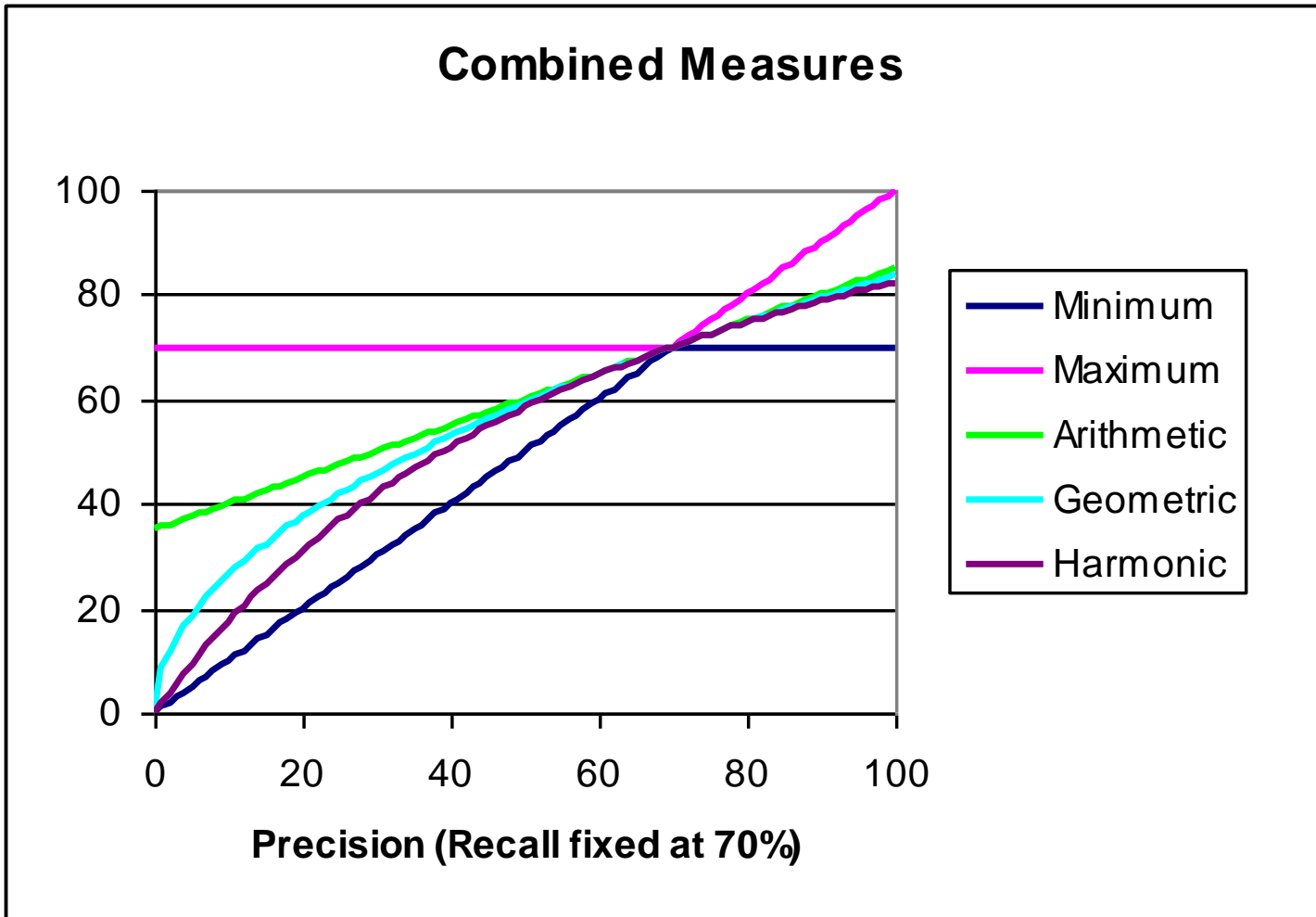
## A COMBINED MEASURE: $F$

- Combined measure that assesses precision/recall tradeoff is **F measure** (weighted harmonic mean):

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- People usually use balanced  $F_1$  measure
  - i.e., with  $\beta = 1$  or  $\alpha = \frac{1}{2}$ , i.e.,  $2*PR/(P+R)$
- **Harmonic mean** is a conservative average
  - See CJ van Rijsbergen, *Information Retrieval*

# $F_1$ (HARMONIC) AND OTHER AVERAGES



## QUIZ: P/R, ACCURACY AND F1

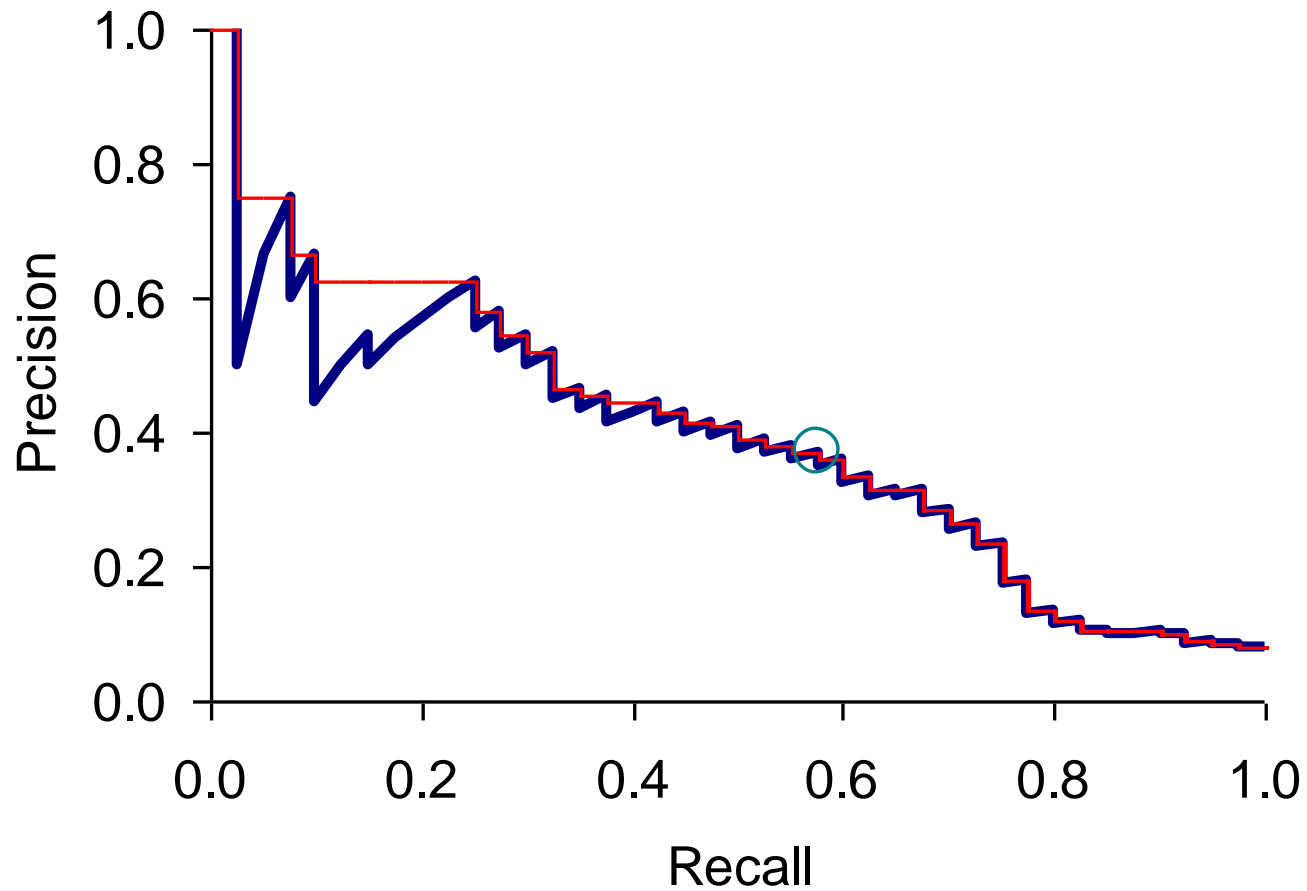
- Compute the Precision, Recall, Accuracy and F1 according to the following table:

	Relevant	Nonrelevant
Retrieved	50	30
Not Retrieved	100	150

# EVALUATING RANKED RESULTS

- Evaluation of ranked results:
  - The system can return any number of results
  - By taking various numbers of the top returned documents (levels of recall), the evaluator can produce a *precision-recall curve*

# A PRECISION-RECALL CURVE



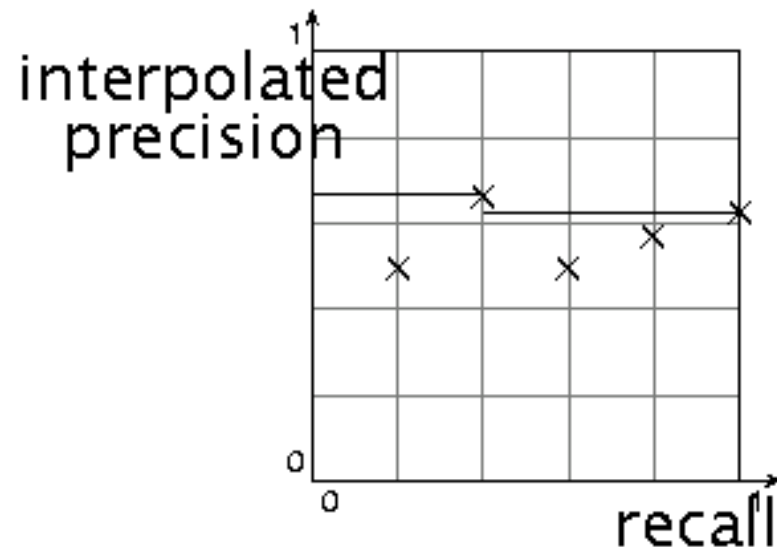
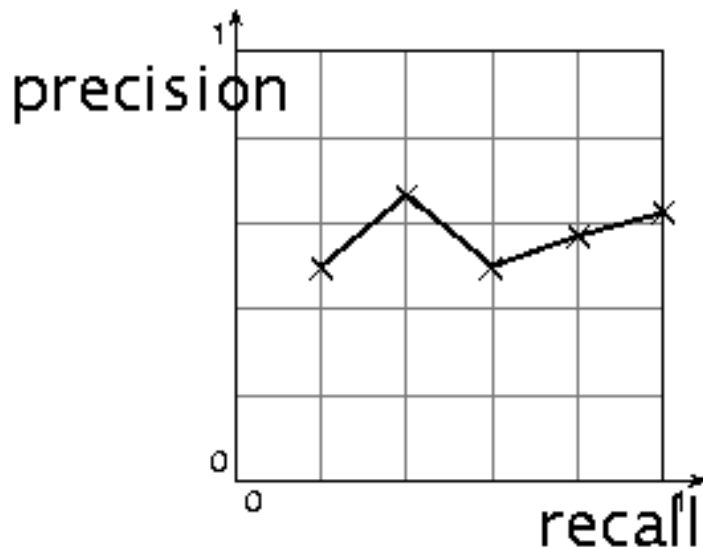
## AVERAGING OVER QUERIES

- A precision-recall graph for **one query** isn't a very sensible thing to look at
- You need to average performance over a whole bunch of queries.
- But there's a technical issue:
  - Precision-recall calculations place some discontinuous points on the graph
  - How do you determine a value (interpolate) between the points?



# INTERPOLATED PRECISION

- Idea: If locally precision increases with increasing recall, then you need to accommodate for that...
- So you take the max of precisions to right of value

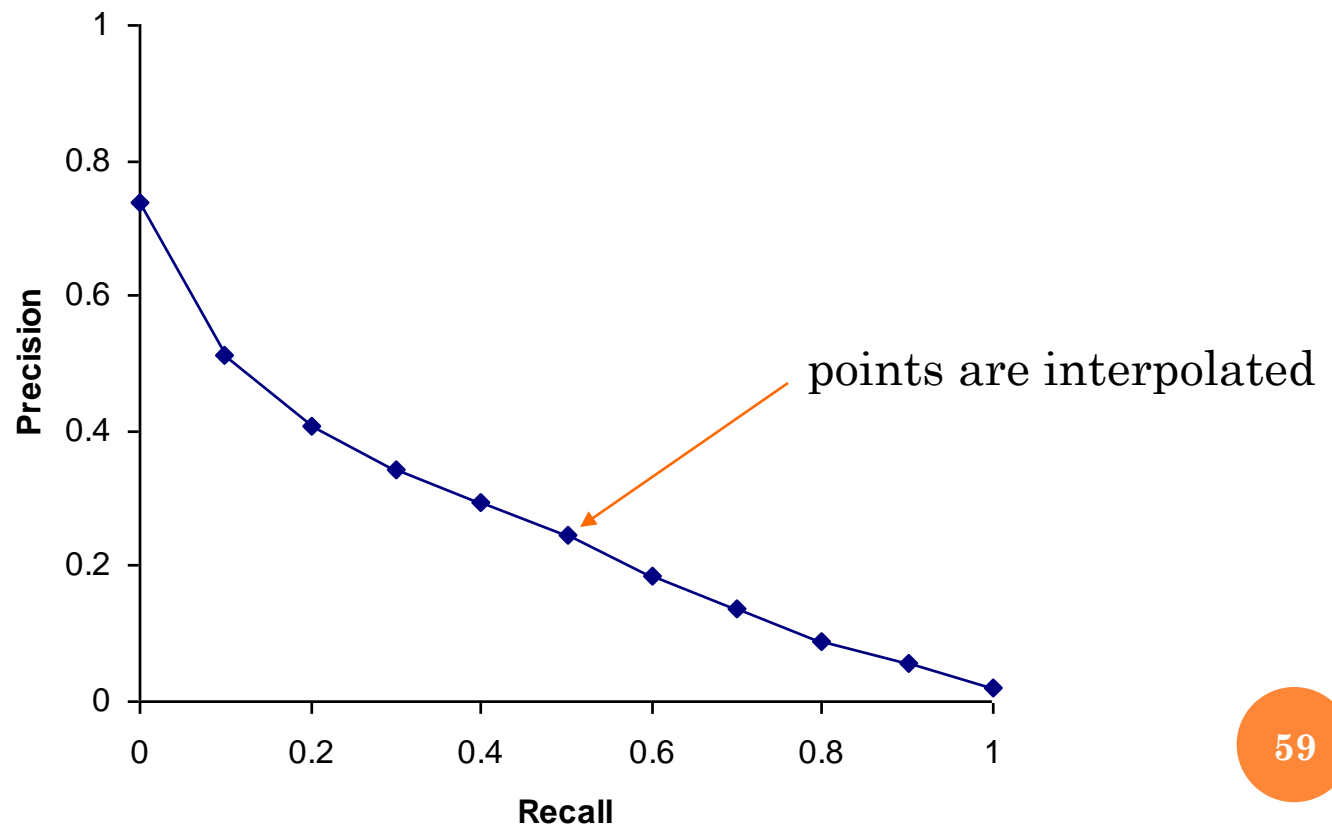


# EVALUATION

- Graphs are good, but people want summary measures!
  - Precision at fixed retrieval level
    - Precision-at- $k$ : Precision of top  $k$  results
    - Perhaps appropriate for most of web search: all people want are good matches on the first one or two results pages
    - But: averages badly and has an arbitrary parameter of  $k$
  - 11-point interpolated average precision
    - The standard measure in the early TREC competitions: you take the precision at 11 levels of recall varying from 0, 0.1, 0.2, 0.3 through 1.0, using interpolation (the value for 0 is always interpolated!), and average them
    - Evaluates performance at all recall levels

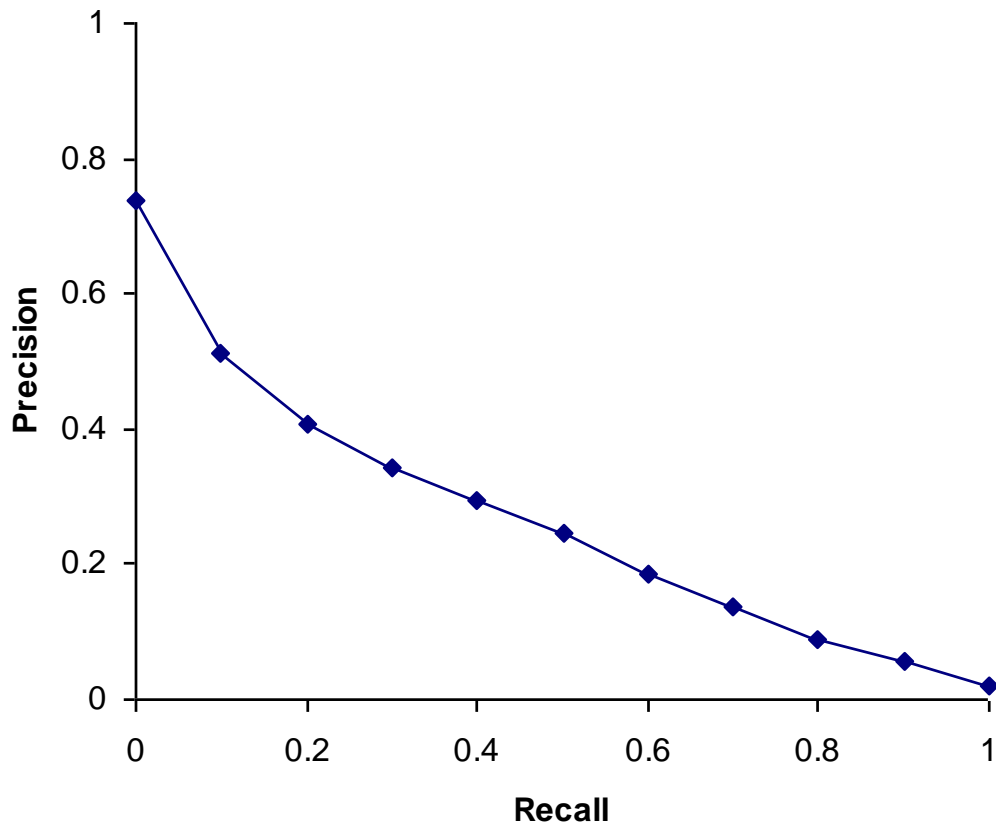
# TYPICAL (GOOD) 11 POINT PRECISIONS

- SabIR/Cornell 8A1 11pt precision from TREC 8 (1999)



# QUIZ: 11-POINT AVERAGE PRECISION

- How do you estimate the precision at recall = 0?




## YET MORE EVALUATION MEASURES...

- Mean average precision (MAP)
  - Average of the precision value obtained for the top  $k$  documents, each time a relevant doc is retrieved
  - Avoids interpolation, use of fixed recall levels
  - MAP for query collection is arithmetic ave.
    - Macro-averaging: each query counts equally
- R-precision
  - If we have a known (though perhaps incomplete) set of relevant documents of size  $Rel$ , then calculate precision of the top  $Rel$  docs returned
  - Perfect system could score 1.0.

# VARIANCE

- For a test collection, it is usual that a system does crummily on some information needs (e.g., MAP = 0.1) and excellently on others (e.g., MAP = 0.7)
- Indeed, it is usually the case that the variance in performance of the same system across queries is much greater than the variance of different systems on the same query.
- That is, there are easy information needs and hard ones!



# CREATING TEST COLLECTIONS FOR IR EVALUATION

63

# TEST COLLECTIONS

**TABLE 4.3 Common Test Corpora**

<i>Collection</i>	<i>NDocs</i>	<i>NQrys</i>	<i>Size (MB)</i>	<i>Term/Doc</i>	<i>Q-D RelAss</i>
ADI	82	35			
AIT	2109	14	2	400	>10,000
CACM	3204	64	2	24.5	
CISI	1460	112	2	46.5	
Cranfield	1400	225	2	53.1	
LISA	5872	35	3		
Medline	1033	30	1		
NPL	11,429	93	3		
OSHMED	34,8566	106	400	250	16,140
Reuters	21,578	672	28	131	
TREC	740,000	200	2000	89-3543	» 100,000



# FROM DOCUMENT COLLECTIONS TO TEST COLLECTIONS

- Still need
  - Test queries
  - Relevance assessments
- Test queries
  - Must be relevant to docs available
  - Best designed by domain experts
  - Random query terms generally not a good idea
- Relevance assessments
  - Human judges, time-consuming
  - Are human panels perfect?

# KAPPA MEASURE FOR INTER-JUDGE (DIS)AGREEMENT

- Kappa measure
  - Agreement measure among judges
  - Designed for categorical judgments
  - Corrects for chance agreement
- $\text{Kappa} = [ P(A) - P(E) ] / [ 1 - P(E) ]$
- $P(A)$  – proportion of time judges agree
- $P(E)$  – what agreement would be by chance
- $\text{Kappa} = 0$  for chance agreement, 1 for total agreement.

P(A)? P(E)?

## KAPPA MEASURE: EXAMPLE

Number of docs	Judge 1	Judge 2
300	Relevant	Relevant
70	Nonrelevant	Nonrelevant
20	Relevant	Nonrelevant
10	Nonrelevant	Relevant

## KAPPA EXAMPLE

- $P(A) = 370/400 = 0.925$
- $P(\text{nonrelevant}) = (10+20+70+70)/800 = 0.2125$
- $P(\text{relevant}) = (10+20+300+300)/800 = 0.7878$
- $P(E) = 0.2125^2 + 0.7878^2 = 0.665$
- $\text{Kappa} = (0.925 - 0.665)/(1-0.665) = 0.776$
  
- $\text{Kappa} > 0.8 = \text{good agreement}$
- $0.67 < \text{Kappa} < 0.8 \rightarrow \text{“tentative conclusions”}$   
(Carletta '96)
- Depends on purpose of study
- For  $>2$  judges: average pairwise kappas

## QUIZ: KAPPA COEFFICIENT

Compute Kappa coefficient for the following test set (Kappa =  $[ P(A) - P(E) ] / [ 1 - P(E) ]$ ):

Number of docs	Judge 1	Judge 2
200	Relevant	Relevant
80	Nonrelevant	Nonrelevant
100	Relevant	Nonrelevant
120	Nonrelevant	Relevant

# TREC

- TREC Ad Hoc task from first 8 TRECs is standard IR task
  - 50 detailed information needs a year
  - Human evaluation of pooled results returned
  - More recently other related things: Web track, HARD track
- A TREC query (TREC 5)

<top>

<num> Number: 225

<desc> Description:

What is the main function of the Federal Emergency Management Agency (FEMA) and the funding level provided to meet emergencies? Also, what resources are available to FEMA such as people, equipment, facilities?

</top>

# STANDARD RELEVANCE BENCHMARKS: OTHERS

- GOV2
  - Another TREC/NIST collection
  - 25 million web pages
  - Largest collection that is easily available
  - But still 3 orders of magnitude smaller than what Google/Bing/Baidu index
- NTCIR
  - East Asian language and cross-language information retrieval
- Cross Language Evaluation Forum (CLEF)
  - This evaluation series has concentrated on European languages and cross-language information retrieval.
- Many others

## IMPACT OF INTER-JUDGE AGREEMENT

- Impact on absolute performance measure can be significant (0.32 vs 0.39)
- Little impact on ranking of different systems or relative performance
- Suppose we want to know if algorithm A is better than algorithm B
- A standard information retrieval experiment will give us a reliable answer to this question.



# CRITIQUE OF PURE RELEVANCE

- Relevance vs Marginal Relevance
  - A document can be redundant even if it is highly relevant
  - Duplicates
  - The same information from different sources
  - Marginal relevance is a better measure of utility for the user.
- Using facts/entities as evaluation units more directly measures true relevance.
- But harder to create evaluation set
- See Carbonell reference

# CAN WE AVOID HUMAN JUDGMENT?

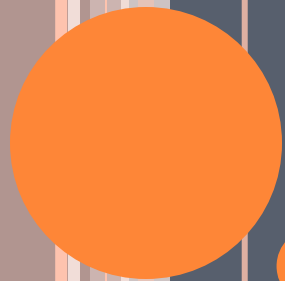
- No.
- Makes experimental work hard
  - Especially on a large scale
- In some very specific settings, can use proxies
  - E.g.: for approximate vector space retrieval, we can compare the cosine distance closeness of the closest docs to those found by an approximate retrieval algorithm
- But once we have test collections, we can reuse them (so long as we don't overtrain too badly)

# EVALUATION AT LARGE SEARCH ENGINES

- Search engines have test collections of queries and hand-ranked results
- Recall is difficult to measure on the web
- Search engines often use precision at top  $k$ , e.g.,  $k = 10$
- . . . or measures that reward you more for getting rank 1 right than for getting rank 10 right.
  - NDCG (Normalized Discounted Cumulative Gain)
- Search engines also use non-relevance-based measures.
  - Clickthrough on first result
    - Not very reliable if you look at a single clickthrough ... but pretty reliable in the aggregate.
  - Studies of user behavior in the lab
  - A/B testing

# A/B TESTING

- Purpose: Test a single innovation
- Prerequisite: You have a large search engine up and running.
- Have most users use old system
- Divert a small proportion of traffic (e.g., 1%) to the new system that includes the innovation
- Evaluate with an “automatic” measure like clickthrough on first result
- Now we can directly see if the innovation does improve user happiness.
- Probably the evaluation methodology that large search engines trust most
- In principle less powerful than doing a multivariate regression analysis, but easier to understand



# RESULTS PRESENTATION

# RESULT SUMMARIES

- Having ranked the documents matching a query, we wish to present a results list
- Most commonly, a list of the document titles plus a short summary, aka “10 blue links”

## [John McCain](#)

**John McCain** 2008 - The Official Website of **John McCain's** 2008 Campaign for President ... African American Coalition; Americans of Faith; American Indians for **McCain**; Americans with ...  
[www.johnmccain.com](http://www.johnmccain.com) · [Cached page](#)

## [JohnMcCain.com - McCain-Palin 2008](#)

**John McCain** 2008 - The Official Website of **John McCain's** 2008 Campaign for President ... African American Coalition; Americans of Faith; American Indians for **McCain**; Americans with ...  
[www.johnmccain.com/Informing/Issues](http://www.johnmccain.com/Informing/Issues) · [Cached page](#)

## [John McCain News- msnbc.com](#)

Complete political coverage of **John McCain**. ... Republican leaders said Saturday that they were worried that Sen. **John McCain** was heading for defeat unless he brought stability to ...  
[www.msnbc.msn.com/id/16438320](http://www.msnbc.msn.com/id/16438320) · [Cached page](#)

## [John McCain | Facebook](#)

Welcome to the official Facebook Page of **John McCain**. Get exclusive content and interact with **John McCain** right from Facebook. Join Facebook to create your own Page or to start ...  
[www.facebook.com/johnmccain](http://www.facebook.com/johnmccain) · [Cached page](#)

# SUMMARIES

- The title is often automatically extracted from document metadata. What about the summaries?
  - This description is crucial.
  - User can identify good/relevant hits based on description.
- Two basic kinds:
  - Static
  - Dynamic
- A **static summary** of a document is always the same, regardless of the query that hit the doc
- A **dynamic summary** is a *query-dependent* attempt to explain why the document was retrieved for the query at hand

# STATIC SUMMARIES

- In typical systems, the static summary is a subset of the document
- Simplest heuristic: the first 50 (or so – this can be varied) words of the document
  - Summary cached at indexing time
- More sophisticated: extract from each document a set of “key” sentences
  - Simple NLP heuristics to score each sentence
  - Summary is made up of top-scoring sentences.
- Most sophisticated: NLP models used to synthesize a summary
  - Seldom used in classic IR; cf. text summarization work, but increasing so with ChatGPT, etc.



# DYNAMIC SUMMARIES

- Present one or more “windows” within the document that contain several of the query terms
  - “KWIC” snippets: “Keyword in Context” presentation

The image displays three search engine results for the query 'christopher manning' and 'christopher manning machine translation'. Each result shows the search engine logo, the search query in a text box, and a snippet of the search results.

**Google**  [Christopher Manning, Stanford NLP](#)  
**Christopher Manning**, Associate Professor of Computer Science and Linguistics, Stanford University.  
[nlp.stanford.edu/~manning/](http://nlp.stanford.edu/~manning/) - 12k - [Cached](#) - [Similar pages](#)

**Google**  [Christopher Manning, Stanford NLP](#)  
**Christopher Manning**, Associate Professor of Computer Science and Linguistics, ...  
 computational semantics, **machine translation**, grammar induction, ...  
[nlp.stanford.edu/~manning/](http://nlp.stanford.edu/~manning/) - 12k - [Cached](#) - [Similar pages](#)

**YAHOO!**  [Christopher Manning, Stanford NLP](#)  
**Christopher Manning**, Associate Professor of Computer Science and Linguistics, Stanford University ... **Chris Manning** works on systems and formalisms that can ...  
[nlp.stanford.edu/~manning](http://nlp.stanford.edu/~manning/) - [Cached](#)

# TECHNIQUES FOR DYNAMIC SUMMARIES

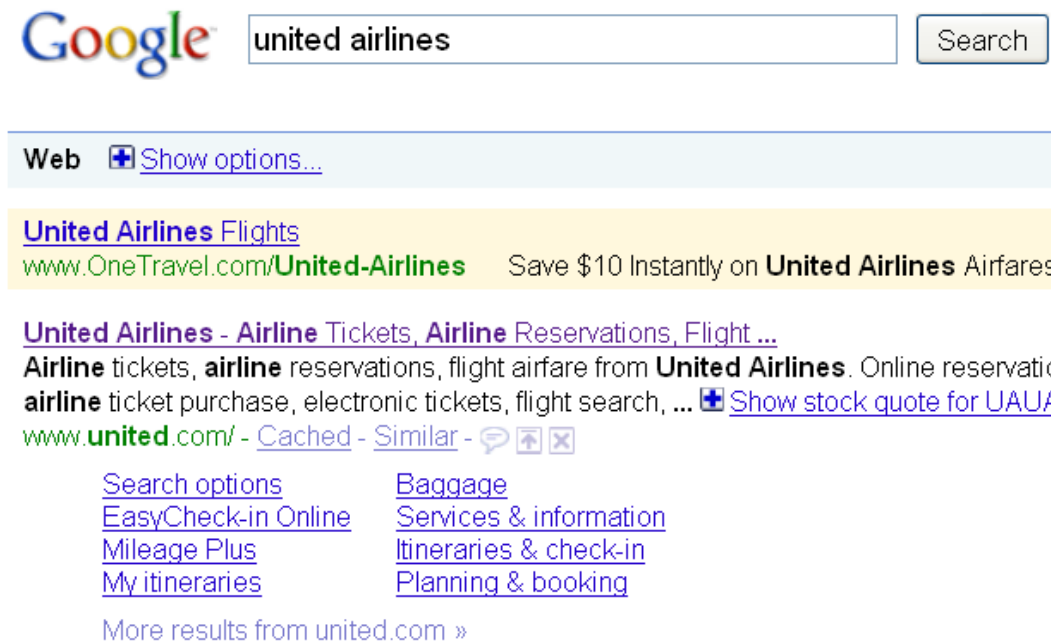
- Find small windows in doc that contain query terms
  - Requires fast window lookup in a document cache
- Score each window wrt query
  - Use various features such as window width, position in document, etc.
  - Combine features through a scoring function – methodology to be covered later
  - Challenges in evaluation: judging summaries
  - Easier to do pairwise comparisons rather than binary relevance assessments

## QUIZ: STATIC SUMMARY

- A static summary can be anything below **except**:
  - a) First 50 words of original document
  - b) Formulated based on the query
  - c) Extracted sentences from original document
  - d) Synthesized from original document

# QUICKLINKS

- For a *navigational query* such as ***united airlines*** user's need likely satisfied on [www.united.com](http://www.united.com)
- Quicklinks provide navigational cues on that home page



Google united airlines Search

Web [+ Show options...](#)

[United Airlines Flights](#)  
[www.OneTravel.com/United-Airlines](http://www.OneTravel.com/United-Airlines) Save \$10 Instantly on **United Airlines** Airfares.

[United Airlines - Airline Tickets, Airline Reservations, Flight ...](#)  
**Airline** tickets, **airline** reservations, flight airfare from **United Airlines**. Online reservation **airline** ticket purchase, electronic tickets, flight search, ... [+ Show stock quote for UUA](#)  
[www.united.com/](http://www.united.com/) - [Cached](#) - [Similar](#) - [🗨](#) [📄](#) [✕](#)

<a href="#">Search options</a>	<a href="#">Baggage</a>
<a href="#">EasyCheck-in Online</a>	<a href="#">Services &amp; information</a>
<a href="#">Mileage Plus</a>	<a href="#">Itineraries &amp; check-in</a>
<a href="#">My itineraries</a>	<a href="#">Planning &amp; booking</a>

[More results from united.com »](#)

united airlines

Search Pad

SearchScan - On

102,000,000 results for united airlines:

Show All

United Air Lines

Wikipedia

Also try: [united airlines reservations](#), [united airlines flight](#), [More...](#)

[United Airlines - Airline Tickets, Airline Reservations ...](#) (Nasdaq: [UAUA](#))

Official site for **United Airlines**, commercial air carrier transporting people, property, and mail across the U.S. and worldwide.

[www.united.com](#) - 65k - [Cached](#)

[Planning & Booking](#)

[Shop for Flights](#)

[Itineraries & Check-in](#)

[Special Deals](#)

[Mileage Plus](#)

[Flight Status](#)

[Services & Information](#)

[Customer Service](#)

[more results from united.com »](#)

united airlines



## UNITED AIRLINES

[United Airline Fleet](#)

[United Airline Schedule](#)

[United Airlines Reservations](#)

[United Airline Jobs](#)

[Reference](#)

ALL RESULTS

[Cheap Flight Tickets](#) · [www.CheapOair.com](#)

CheapOair - The Only Way to Go!! Find Over 18 Million Exclusive Fares.

[Fly United Airlines](#) · [www.OneTravel.com/United-Airline](#)

Save \$10 Instantly on **United Airlines** Flights. Book Now, Hurry!

Best match

[United Airlines - Airline Tickets, Airline Reservations, Flight ...](#)

[www.united.com](#) · Official site

**Airline** tickets, **airline** reservations, flight airfare from **United Airlines**. Online reservations, **airline** ticket purchase, electronic tickets, flight search, fares and availability ...

[Flights](#)

[Redeem miles](#)

[Check In Online](#)

[Children, pets, & assistance](#)

[My itineraries](#)

[Change your travel plans](#)

[Baggage](#)

[Special deals](#)

Customer service 800-864-8331

## RELATED SEARCHES

[United Airlines Flight Status](#)

[US Airways](#)


[Continental Airlines](#)

# ALTERNATIVE RESULTS PRESENTATIONS?

YAHOO!®

Web Images Video Local Shopping News more ▾

uni Search

- united airlines** ▶ **UNITED AIRLINES - AIRLINE TICKETS,...**  
Airline tickets, airline reservations, flight airfare from United Airlines.  
Online reservations, ...  
 [www.united.com](http://www.united.com)
- univision
- university of phoenix
- asian unicorn
- universal studios
- united states postal service
- united healthcare

**MORE INFO**

<a href="#">Flights</a>	<a href="#">Check In Online</a>
<a href="#">Mileage Plus</a>	<a href="#">My Itineraries</a>
<a href="#">Baggage</a>	<a href="#">Redeem Miles</a>

# RESOURCES FOR THIS LECTURE

- IIR 8
- MIR Chapter 3
- MG 4.5
- Carbonell and Goldstein 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. SIGIR 21.