Web Science

5 February 2019

Christina Lioma
c.lioma@di.ku.dk

Lecture 1 plan

- General course information
- Introduction to the WWW

Teaching Team

Lectures (Tuesday 13h00-15h00) @ UP1

- Isabelle Augenstein
- Christina Lioma (course responsible)

Labs (Tuesday 15h00-17h00) @ bib 4-0-17

- Casper Hansen
- Christian Hansen
- Lucas Chaves

Course Info Resources

Absalon:

- Lecture plan, projects, readings, slides, latest news and other **important** information
- Keep an eye on the course homepage **throughout** the block for information updates
- **Familiarise** yourselves with Absalon

Course Info Resources

Absalon:

- Lecture plan, projects, readings, slides, latest news and other **important** information
- Keep an eye on the course homepage **throughout** the block for information updates
- **Familiarise** yourselves with Absalon
- Last minute changes (e.g. class cancellation, change of room)

Course Info Resources

Absalon:

- Lecture plan, projects, readings, slides, latest news and other **important** information
- Keep an eye on the course homepage **throughout** the block for information updates
- **Familiarise** yourselves with Absalon
- Last minute changes (e.g. class cancellation, change of room)

Also:

https://kurser.ku.dk/course/ndak14004u/2018-2019

Readings:

- On Absalon course page (no single textbook)
- Provide important context to **supplement** lectures; they do not replace lectures

Readings:

- On Absalon course page (no single textbook)
- Provide important context to **supplement** lectures; they do not replace lectures

Lectures:

- Slides supplement the oral lectures; they do not replace them
- Pointers to more readings and sources

Labs:

- Help with the projects; not solve them for you
- Answer questions about the project or lectures

Attendance:

- Your responsibility to attend; if not, no *formal* way of catching up

Plagiarism:

- automatic fail on project
- referral to head of students

Prerequisites:

- Programming
- Machine Learning

To pass the course

1) Continuous project (throughout the course):

- Individual
- Includes oral presentation & QA
- 40% of final grade

2) Take home assignment:

- Individual
- 1-5 April 2019
- 60% of final grade

Re-exam

New projects AND oral exam on the full course syllabus without preparation

Re-exam

New projects AND oral exam on the full course syllabus without preparation

Both are compulsory:

- If you do not submit the new projects, you cannot take the oral exam → automatic fail
- If you do not show up at the oral → automatic fail

Final grade is based on the overall assessment of both (not average)

Today’s lecture

- What is Web Science
- What is the Web
- What is the Internet
- Web graph
- Main challenges of web data processing
- Web crawling

Web Science: non-trivial processing and application of implicit, previously unknown, and potentially useful information from web data

Web Science: non-trivial processing and application of implicit, previously unknown, and potentially useful information from **web data**

Web Science: non-trivial processing and application of implicit, previously unknown, and potentially useful information from **web data**

Web data examples: webpages, blogs, tweets, amazon clicks, youtube comments, dr.dk updates, sourceforge downloads…

Web Science: non-trivial processing and **application** of implicit, previously unknown, and potentially useful information from **web data**

Web data examples: webpages, blogs, tweets, amazon clicks, youtube comments, dr.dk updates, sourceforge downloads…

Web Science: non-trivial processing and **application** of implicit, previously unknown, and potentially useful information from **web data**

Web data examples: webpages, blogs, tweets, amazon clicks, youtube comments, dr.dk updates, sourceforge downloads…

Application examples: opinion mining, trend detection, recommendation, web search, crowdsourcing, web analytics...

Web Science: **non-trivial** processing and **application** of implicit, previously unknown, and potentially useful information from **web data**

Web data examples: webpages, blogs, tweets, amazon clicks, youtube comments, dr.dk updates, sourceforge downloads…

Application examples: opinion mining, trend detection, recommendation, web search, crowdsourcing, web analytics...

Web Science: **non-trivial** processing and **application** of implicit, previously unknown, and potentially useful information from **web data**

Web data examples: webpages, blogs, tweets, amazon clicks, youtube comments, dr.dk updates, sourceforge downloads…

Application examples: opinion mining, trend detection, recommendation, web search, crowdsourcing, web analytics...

Comprises elements of: machine learning, information retrieval, natural language processing, data mining, databases, statistics...

Web Science: **non-trivial** processing and **application** of implicit, previously unknown, and potentially useful information from **web data**

Web data examples: webpages, blogs, tweets, amazon clicks, youtube comments, dr.dk updates, sourceforge downloads…

Application examples: opinion mining, trend detection, recommendation, web search, crowdsourcing, web analytics...

Comprises elements of: machine learning, information retrieval, natural language processing, data mining, databases, statistics...

Spans from back-end (algorithms) to front-end (visualisation)

Web Science comprises elements of: machine learning, information retrieval, natural language processing, data mining, databases, statistics...

Different scientific cultures

Web Science comprises elements of: machine learning, information retrieval, natural language processing, data mining, databases, statistics...

Different scientific cultures

- *Machine learning*: focus on “complex” methods, “small” data

Web Science comprises elements of: machine learning, information retrieval, natural language processing, data mining, databases, statistics...

Different scientific cultures

- *Machine learning*: focus on “complex” methods, “small” data
- *Information Retrieval*: focus on large-scale (non main-memory) data

Web Science comprises elements of: machine learning, information retrieval, natural language processing, data mining, databases, statistics...

Different scientific cultures

- *Machine learning*: focus on “complex” methods, “small” data
- *Information Retrieval*: focus on large-scale (non main-memory) data
- *Natural language processing*: focus on “linguistic” analyses of data (paradigmatic)
- *Data mining*: focus on discovering patterns in data

Web Science comprises elements of: machine learning, information retrieval, natural language processing, data mining, statistics...

Different scientific cultures

- *Machine learning*: focus on “complex” methods, “small” data
- *Information Retrieval*: focus on large-scale (non main-memory) data
- *Natural language processing*: focus on “linguistic” analyses of data (paradigmatic)
- *Data mining*: focus on discovering patterns in data
- *Visualisation*: focus on user-intuitive data overviewing
- *Data cleaning*: focus on detecting bogus data, e.g. age=150

Web

Web: not the same as the Internet

Web: not the same as the Internet

Internet: massive network of computers that can communicate (transfer data between them) in various languages called *protocols*.

Web: not the same as the Internet

Internet: massive network of computers that can communicate (transfer data between them) in various languages called *protocols*.

World Wide Web (or Web): information-sharing model built on top of the Internet, which uses the HTTP protocol to transfer data.

Web: not the same as the Internet

Internet: massive network of computers that can communicate (transfer data between them) in various languages called *protocols*.

World Wide Web (or Web): information-sharing model built on top of the Internet, which uses the HTTP protocol to transfer data.

Web: one of many ways to share information on the Internet. Other ways are email (SMTP protocol), instant messaging (SIMPLE), FTP, …

Web: not the same as the Internet

Internet: massive network of computers that can communicate (transfer data between them) in various languages called *protocols*.

World Wide Web (or Web): information-sharing model built on top of the Internet, which uses the HTTP protocol to transfer data.

Web: one of many ways to share information on the Internet. Other ways are email (SMTP protocol), instant messaging (SIMPLE), FTP, …

The Web is part of the Internet

Web as a graph

- A graph $G = (V, E)$ is defined by
  - a set V of vertices (nodes)
  - a set E of edges (links) connecting pairs of nodes

- The **Web page graph**
  - V is the set of pages
  - E is the set of hyperlinks

Web as a graph (directed)

- A graph G = (V, E) is defined by
  - a set V of vertices (nodes)
  - a set E of edges (links) connecting pairs of nodes

- The **Web page graph** (**directed**)
  - V is the set of pages
  - E is the set of hyperlinks (**inlinks & outlinks**)

Web as a graph (directed)

- A graph G = (V, E) is defined by
  - a set V of vertices (nodes)
  - a set E of edges (links) connecting pairs of nodes

- The **Web page graph** (directed)
  - V is the set of pages
  - E is the set of hyperlinks (inlinks & outlinks)

- Many more graphs can be defined, e.g. host graph, co-citation graph

Web as a graph (directed)

- A graph G = (V, E) is defined by
  - a set V of vertices (nodes)
  - a set E of edges (links) connecting pairs of nodes

- The **Web page graph** (directed)
  - V is the set of pages
  - E is the set of hyperlinks (inlinks & outlinks)

- Many more graphs can be defined, e.g. host graph, co-citation graph

- ~10-20 hyperlinks per page on average

Web as a graph (directed)

- A graph G = (V, E) is defined by
  - a set V of vertices (nodes)
  - a set E of edges (links) connecting pairs of nodes

- The **Web page graph** (directed)
  - V is the set of pages
  - E is the set of hyperlinks (inlinks & outlinks)

- Many more graphs can be defined, e.g. host graph, co-citation graph

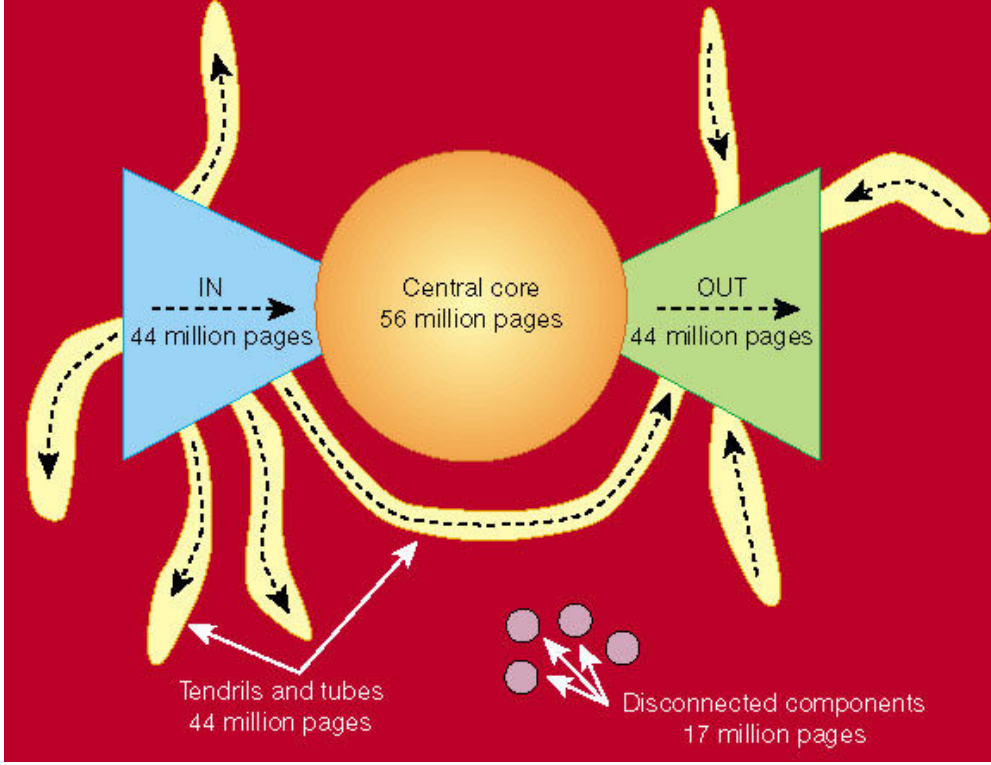
- ~10-20 hyperlinks per page on average

Power law distribution of hyperlinks:

- very few pages have the most hyperlinks
- vast majority of pages have very few hyperlinks

Bow-Tie Structure of the Web

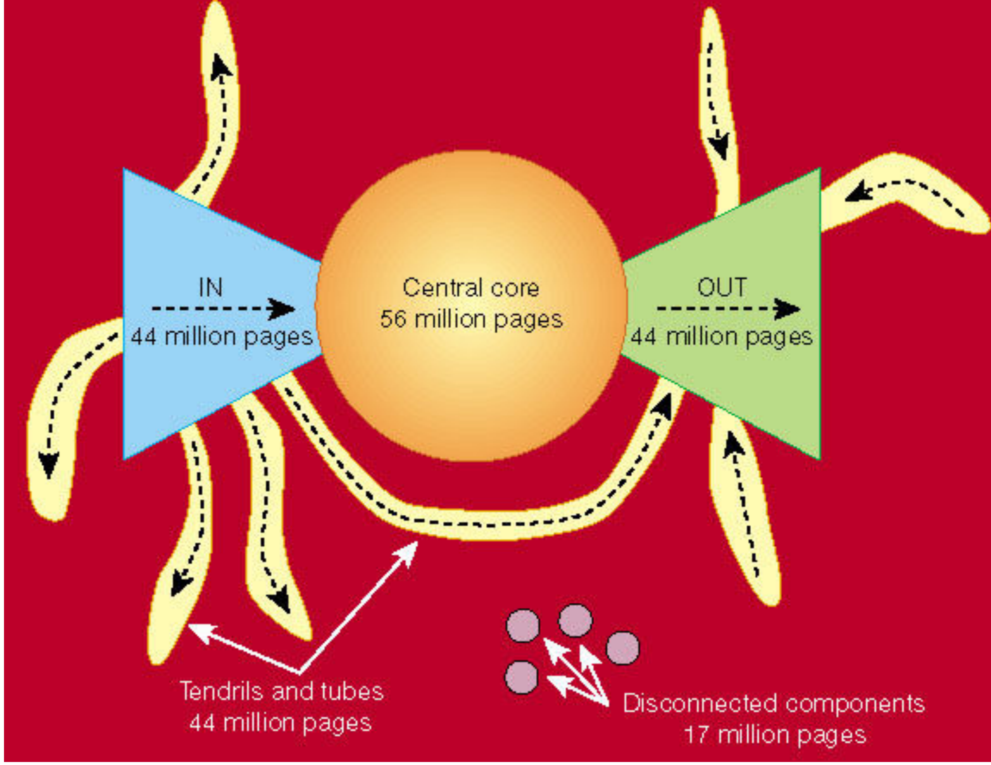
(Broder et al, 1999)



Bow-Tie Structure of the Web

(Broder et al, 1999)

Core: 27%



<u>Core</u>: SCC (strongly connected component) − can go from any node to any node via a directed path

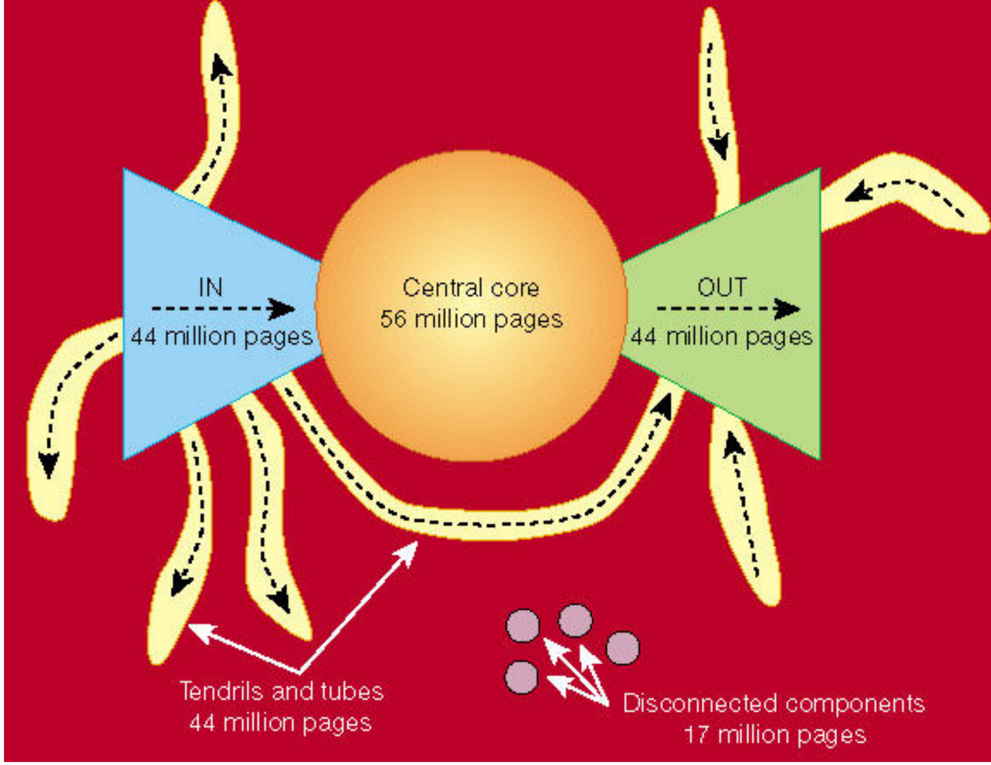
Bow-Tie Structure of the Web

(Broder et al, 1999)

Core: 27%

IN: 21%

OUT: 22%



Core: SCC (strongly connected component) − can go from any node to any node via a directed path

IN: can reach core, but cannot be reached from it

OUT: can be reached from core, but cannot reach it

Bow-Tie Structure of the Web

(Broder et al, 1999)

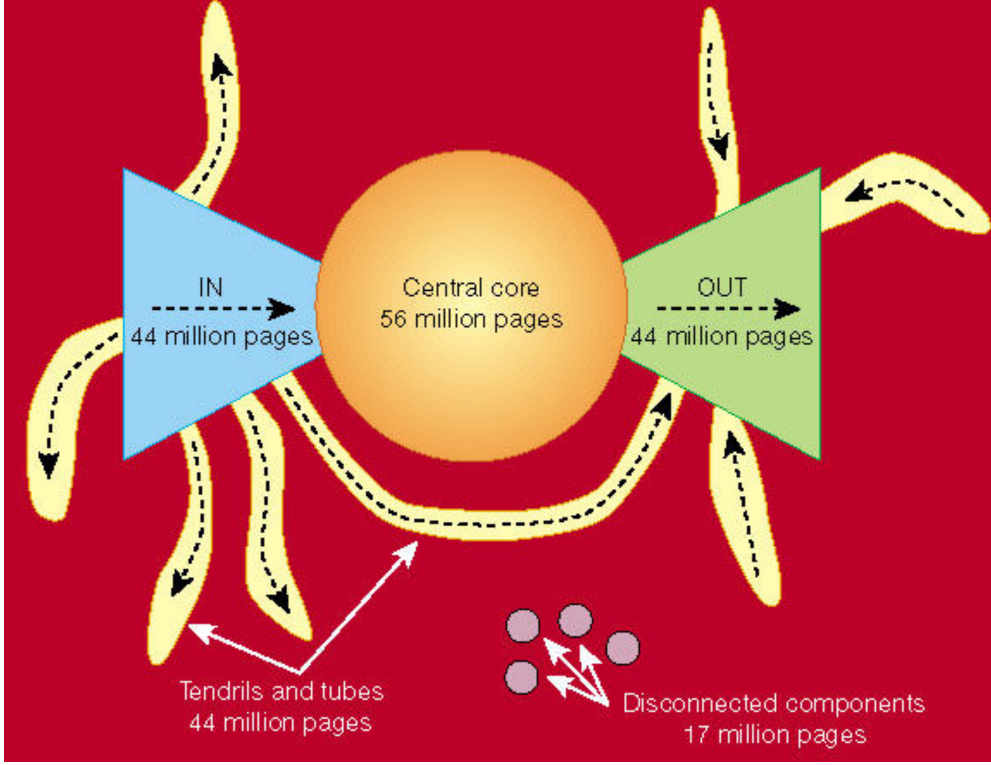
Core: 27%

IN: 21%

OUT: 22%

Tendrils: 22%

Disconnected: 8%



Core: SCC (strongly connected component) – can go from any node to any node via a directed path

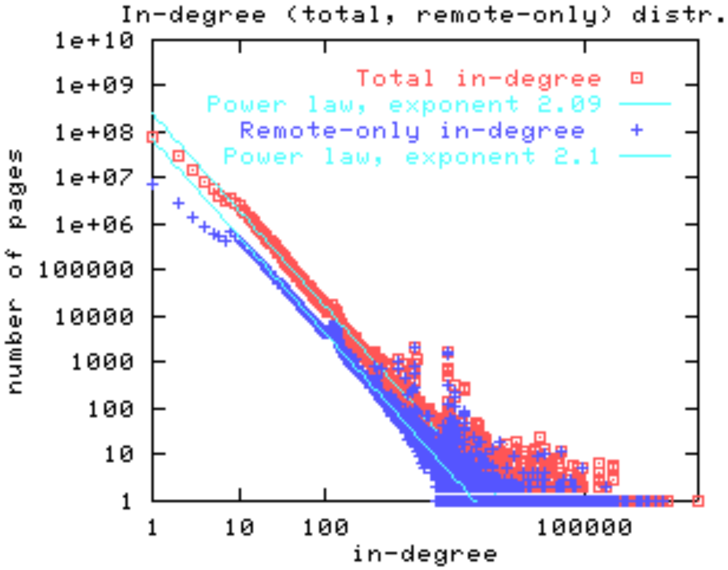
IN: can reach core, but cannot be reached from it

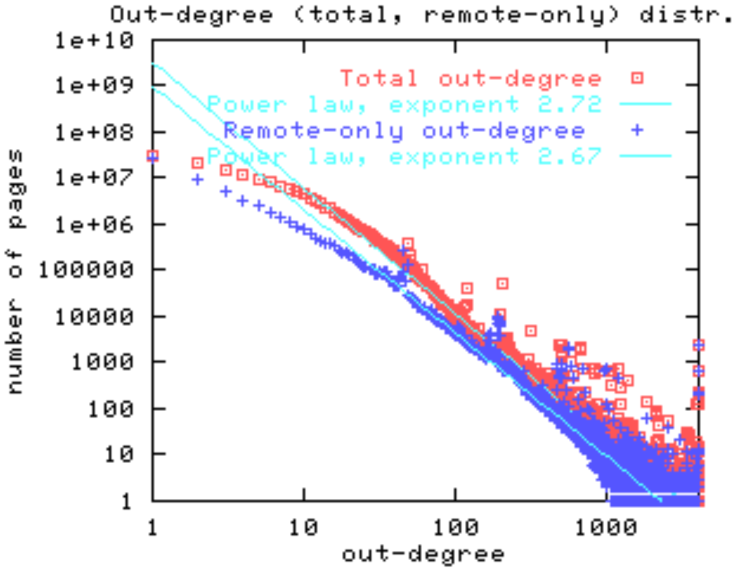
OUT: can be reached from core, but cannot reach it

Tendrils: (a) reachable from IN but cannot reach core OR/AND

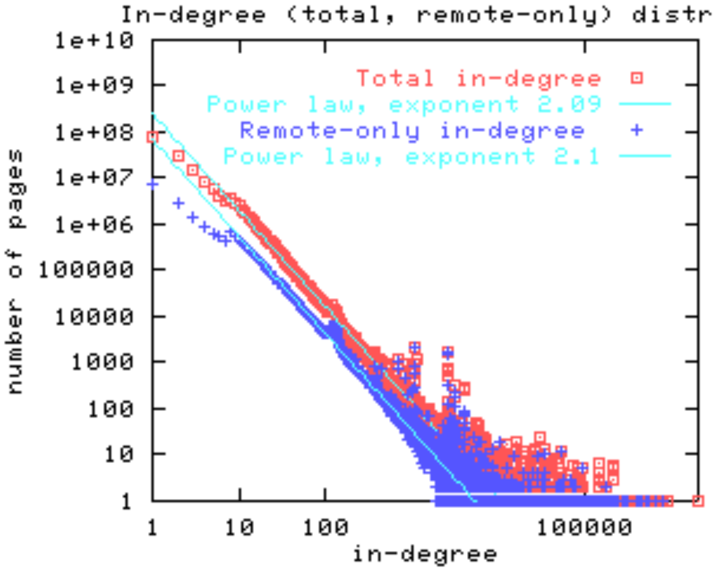
(b) can reach OUT but cannot be reached from it

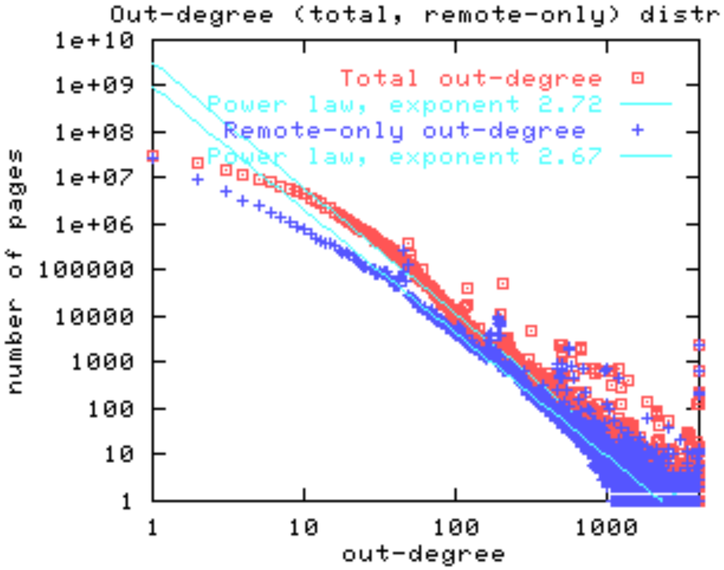
Disconnected: no path to core even if direction is ignored





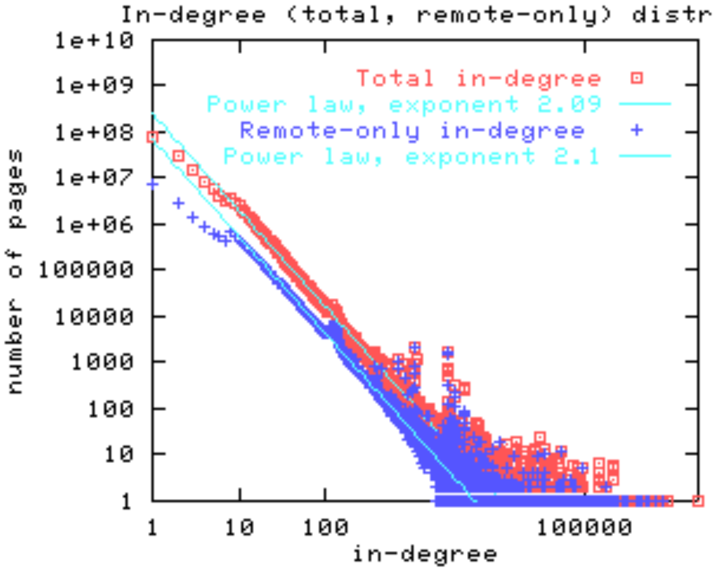
In & out degree (number of links) distribution: power-law with exponent 2.1 and 2.7

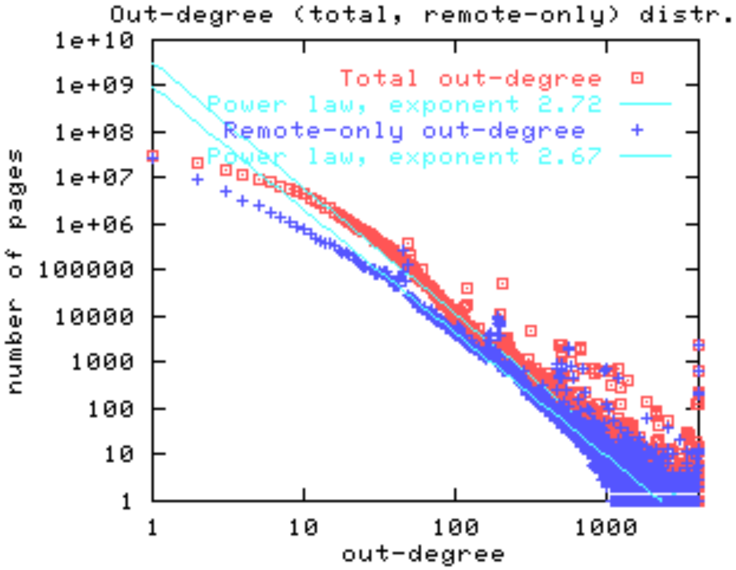




In & out degree (number of links) distribution: power-law with exponent 2.1 and 2.7

Exponent > 2: the expected value of the degree is a constant (not growing with the number of nodes)

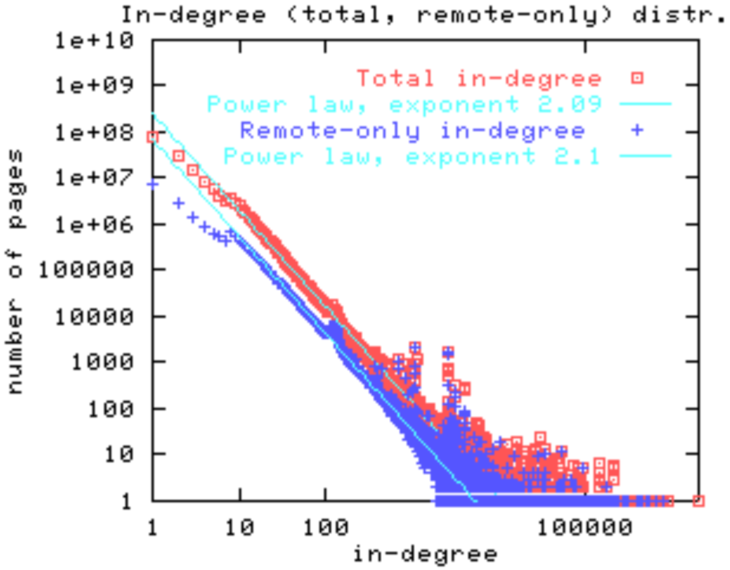


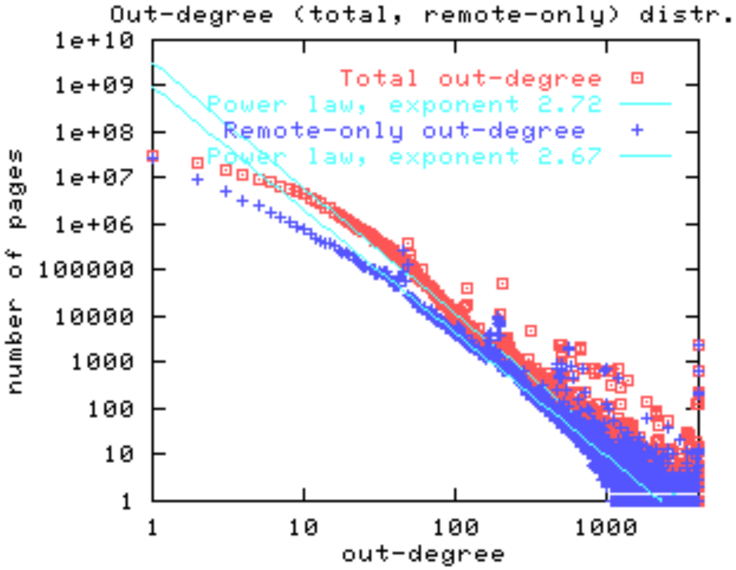


In & out degree (number of links) distribution: power-law with exponent 2.1 and 2.7

Exponent > 2: the expected value of the degree is a constant (not growing with the number of nodes)

Therefore, the expected number of links is linear in the number of nodes





In & out degree (number of links) distribution: power-law with exponent 2.1 and 2.7

Exponent > 2: the expected value of the degree is a constant (not growing with the number of nodes)

Therefore, the expected number of links is linear in the number of nodes

Good news (because we cannot handle anything more than linear)

Why do we care about the Web graph?

- Exploit the Web structure for
  - crawlers
  - search and link analysis
  - spam detection
  - community discovery
  - classification/organization

Why do we care about the Web graph?

- Exploit the Web structure for
  - crawlers
  - search and link analysis
  - spam detection
  - community discovery
  - classification/organization

- Predict the Web future
  - mathematical models
  - algorithm analysis
  - sociological understanding
  - new business opportunities
  - new politics

Why do we care about the Web graph?

- Exploit the Web structure for
  - crawlers
  - search and link analysis
  - spam detection
  - community discovery
  - classification/organization

- Predict the Web future
  - mathematical models
  - algorithm analysis
  - sociological understanding
  - new business opportunities
  - new politics

- Largest human artifact ever created (?)

Web data: webpages, blogs, tweets, amazon clicks, youtube comments, dr.dk updates, sourceforge downloads, satellite data…

Big data: term coined by META (now Gartner) in 2001

Web data: webpages, blogs, tweets, amazon clicks, youtube comments, dr.dk updates, sourceforge downloads, satellite data…

Big data: term coined by META (now Gartner) in 2001

“Big Data problem”: The rate of data accumulation is rising faster than our cognitive capacity to analyse increasingly large datasets to make decisions

Web data: webpages, blogs, tweets, amazon clicks, youtube comments, dr.dk updates, sourceforge downloads, satellite data…

Big data: term coined by META (now Gartner) in 2001

“Big Data problem”: The rate of data accumulation is rising faster than our cognitive capacity to analyse increasingly large datasets to make decisions

Challenges:

- Volume, Variety, Veracity, Velocity (the 4 Vs of data)
- Situation, Scale, Semantics, Sequence (the 4 Ss of data)

Volume, Variety, Veracity, Velocity (the 4 Vs of data)

Volume: substantially large-scale & increasing.

Volume, Variety, Veracity, Velocity (the 4 Vs of data)

Volume: substantially large-scale & increasing. Examples:

- 90% of data has accumulated in the last 2 years (Ghavami, 2016)

Volume, Variety, Veracity, Velocity (the 4 Vs of data)

Volume: substantially large-scale & increasing. Examples:

- 90% of data has accumulated in the last 2 years (Ghavami, 2016)
- Self-driving cars will generate 2PB of data every year

Volume, Variety, Veracity, Velocity (the 4 Vs of data)

Volume: substantially large-scale & increasing. Examples:

- 90% of data has accumulated in the last 2 years (Ghavami, 2016)
- Self-driving cars will generate 2PB of data every year
- The entire writings of all humankind from the beginning of history up to now in all languages is ~50 petabytes

Volume, Variety, Veracity, Velocity (the 4 Vs of data)

Volume: substantially large-scale & increasing. Examples:

- 90% of data has accumulated in the last 2 years (Ghavami, 2016)
- Self-driving cars will generate 2PB of data every year
- The entire writings of all humankind from the beginning of history up to now in all languages is ~50 petabytes
- Data volume in 2020 will be 40ZB (International Data Corporation)

Gigabyte − 1K Megabytes	A movie of TV quality
Terabyte − 1K Gigabytes	All x-ray films in a large hospital
Petabyte − 1K Terabytes	Half of all US academic research libraries
Exabyte − 1K Petabytes	Data generated from SKA telescope per day
Zetabyte − 1K Exabytes	All worldwide data generated by June 2012
Yottabyte − 1K Zetabytes	1YB=1000^8 bytes

Volume, Variety, Veracity, Velocity (the 4 Vs of data)

Variety:

- Data previously confined to paper are now digital & new forms of data, previously non-existent

Volume, Variety, Veracity, Velocity (the 4 Vs of data)

Variety:

- Data previously confined to paper are now digital & new forms of data, previously non-existent
- Both human and machine generated

Volume, Variety, Veracity, Velocity (the 4 Vs of data)

Variety:

- Data previously confined to paper are now digital & new forms of data, previously non-existent
- Both human and machine generated
- Almost all devices will soon generate their own data. E.g., sensors, smart pumps, ventilators, audio recordings of patient-doctor sessions, videos captured during surgery, colour images of wounds, customer sentiment, social media, genetic sequence, …

Internet of Things (IoT): all devices communicate freely with each other through the Internet

Volume, Variety, Veracity, Velocity (the 4 Vs of data)

Variety:

- Data previously confined to paper are now digital & new forms of data, previously non-existent
- Both human and machine generated
- Almost all devices will soon generate their own data. E.g., sensors, smart pumps, ventilators, audio recordings of patient-doctor sessions, videos captured during surgery, colour images of wounds, customer sentiment, social media, genetic sequence, …

Internet of Things (IoT): all devices communicate freely with each other through the Internet

- Heterogeneous: .pdf, .txt, .html, .css, .xml, .gif, .mp3, .mp4, …
- Semi/un/structured: database, excel, free text,…

Volume, Variety, Veracity, Velocity (the 4 Vs of data)

Variety:

- Data previously confined to paper are now digital & new forms of data, previously non-existent
- Both human and machine generated
- Almost all devices will soon generate their own data. E.g., sensors, smart pumps, ventilators, audio recordings of patient-doctor sessions, videos captured during surgery, colour images of wounds, customer sentiment, social media, genetic sequence, …

Internet of Things (IoT): all devices communicate freely with each other through the Internet

- Heterogeneous: .pdf, .txt, .html, .css, .xml, .gif, .mp3, .mp4, …
- Semi/un/structured: database, excel, free text,…
- By 2020, 85% of all data will be in new data types and formats (International Data Corporation)

Volume, Variety, Veracity, Velocity (the 4 Vs of data)

Veracity:

- Uncertainty (unknown/unreliable sources…)

Volume, Variety, Veracity, Velocity (the 4 Vs of data)

Veracity:

- Uncertainty (unknown/unreliable sources…)
- Noise-prone: intention (e.g. spam), form (e.g. typos, chat lingo), content (e.g. bias, factual error, fake news VS sarcasm), source (e.g. near-duplicate ~40%, corrupt OCR), …

Volume, Variety, Veracity, Velocity (the 4 Vs of data)

Veracity:

- Uncertainty (unknown/unreliable sources…)
- Noise-prone: intention (e.g. spam), form (e.g. typos, chat lingo), content (e.g. bias, factual error, fake news VS sarcasm), source (e.g. near-duplicate ~40%, corrupt OCR), …

Velocity:

- Dynamic: updated at various frequencies, often without warning (e.g. no timestamp on the update)

Volume, Variety, Veracity, Velocity (the 4 Vs of data)

Veracity:

- Uncertainty (unknown/unreliable sources…)
- Noise-prone: intention (e.g. spam), form (e.g. typos, chat lingo), content (e.g. bias, factual error, fake news VS sarcasm), source (e.g. near-duplicate ~40%, corrupt OCR), …

Velocity:

- Dynamic: updated at various frequencies, often without warning (e.g. no timestamp on the update)
- Time-series (some): from near real-time (e.g. instant messaging) to periodicity (e.g. web search query logs)

Situation, Scale, Semantics, Sequence (the 4 Ss of data)

Situation, Scale, Semantics, Sequence (the 4 Ss of data)

- **Situation**: context of data measurement. E.g. Blood pressure value when at rest VS standing up VS after climbing stairs

Situation, Scale, Semantics, Sequence (the 4 Ss of data)

- **Situation**: context of data measurement. E.g. Blood pressure value when at rest VS standing up VS after climbing stairs

- **Scale**: data with limited range VS wide range. A slight change can be significant in limited range data, but should be ignored in wide range data.

Situation, Scale, Semantics, Sequence (the 4 Ss of data)

- **Situation**: context of data measurement. E.g. Blood pressure value when at rest VS standing up VS after climbing stairs

- **Scale**: data with limited range VS wide range. A slight change can be significant in limited range data, but should be ignored in wide range data.

- **Semantics**: circa 80% of data is unstructured → extracting pertinent terms from unstructured data is a challenge

- **Sequence** (same as Velocity): sequential or time series data

What is the size of the Web?

Surprisingly hard to answer

What is the size of the Web?

Surprisingly hard to answer

- Naïve solution: keep crawling until the whole graph has been explored
- Extremely simple but wrong solution: crawling is complicated
  - Spamming, duplicates, mirrors, …

What is the size of the Web?

Surprisingly hard to answer

- Naïve solution: keep crawling until the whole graph has been explored
- Extremely simple but wrong solution: crawling is complicated
  - Spamming, duplicates, mirrors, …
- Simple example of a complication: Soft 404
  - If a page does not exist, the server is supposed to return an error code = “404”
  - Many servers do not return an error code, but keep the visitor on site, or simply send the visitor to the home page

- The Web that we see is what the web crawler discovers

- The Web that we see is what the web crawler discovers
- We need large crawls in order to make meaningful measurements
- The measurements are still biased by
  - the crawling policy
  - size limitations of the crawl
  - perturbations of the "natural" process of birth and death of nodes and links

- The Web that we see is what the web crawler discovers
- We need large crawls in order to make meaningful measurements
- The measurements are still biased by
  - the crawling policy
  - size limitations of the crawl
  - perturbations of the "natural" process of birth and death of nodes and links

Estimates:

1999: 800 million pages (Lawrence and Giles)

2008: 1 trillion pages (https://googleblog.blogspot.dk/2008/07/we-knew-web-was-big.html)

The *deep* (or *hidden* or *invisible)* Web contains 400-550 times more information than the known Web [Bergman, 2001: “The deep web: surfacing hidden value”]

Today’s lecture

- ~~What is Web Science~~
- ~~What is the Web~~
- ~~What is the Internet~~
- ~~Web graph~~
- ~~Main challenges of web data processing~~
- Web crawling

Crawler (a.k.a. spider, bot, worm, ant, scutter, harvester …)

- Program that automatically locates, fetches and stores webpages efficiently & methodically

http://www.google.com/support/enterprise/static/gsa/docs/admin/70/gsa_doc_set/admin_crawl/introduction.html

Crawler (a.k.a. spider, bot, worm, ant, scutter, harvester …)

- Program that automatically locates, fetches and stores webpages efficiently & methodically

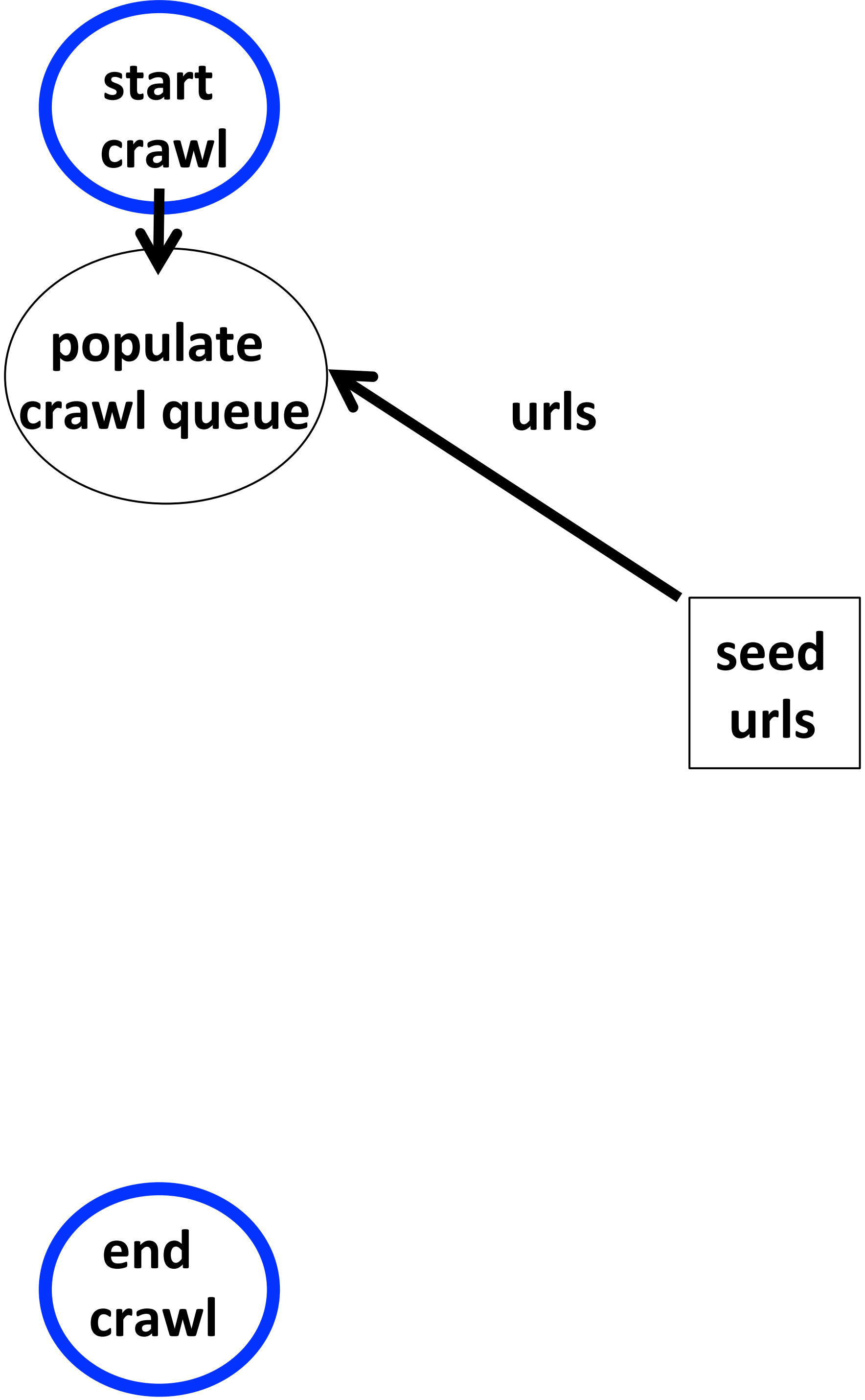
Aim: gather as many useful webpages as possible

How: following hyperlinks

http://www.google.com/support/enterprise/static/gsa/docs/admin/70/gsa_doc_set/admin_crawl/introduction.html

start crawl

end crawl



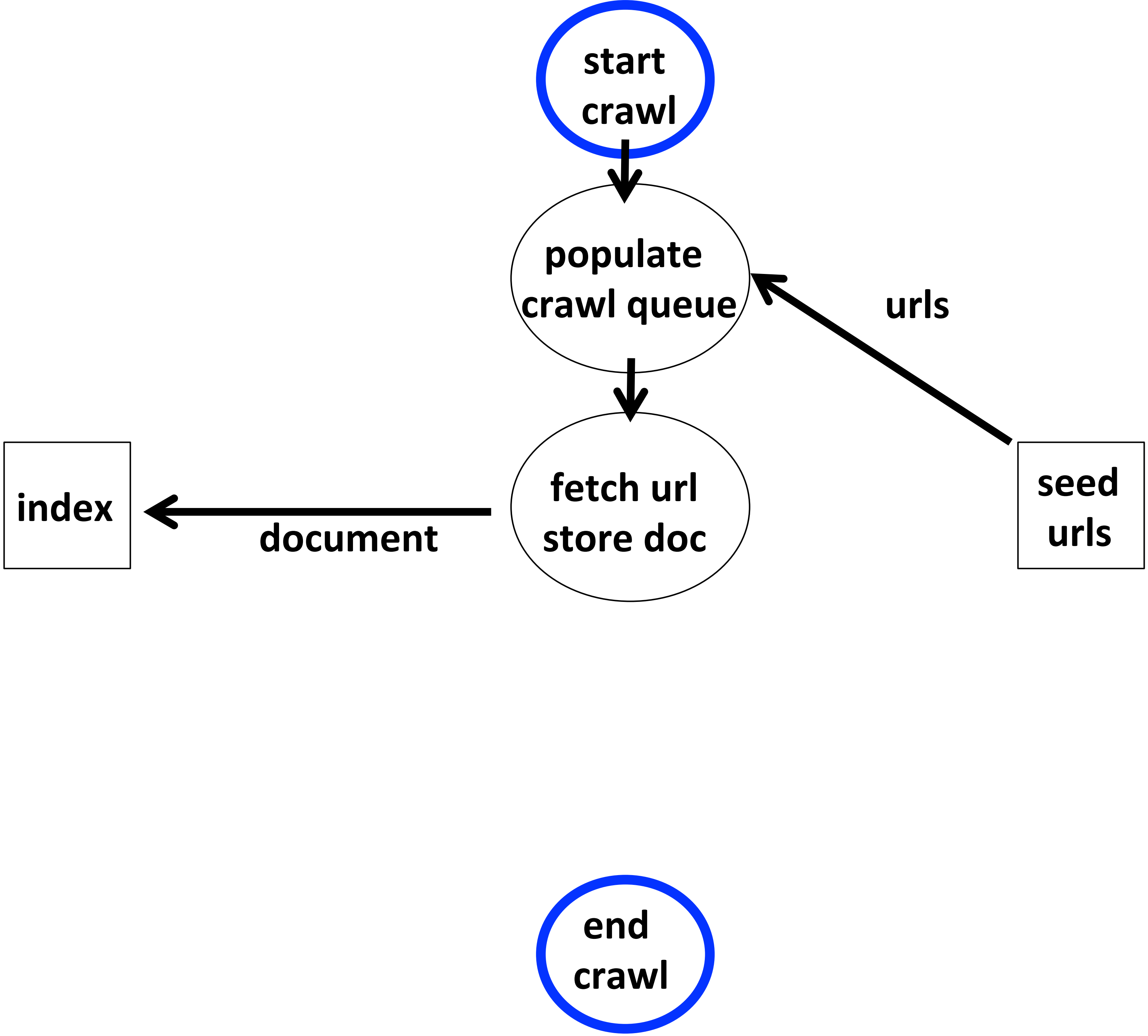
start crawl

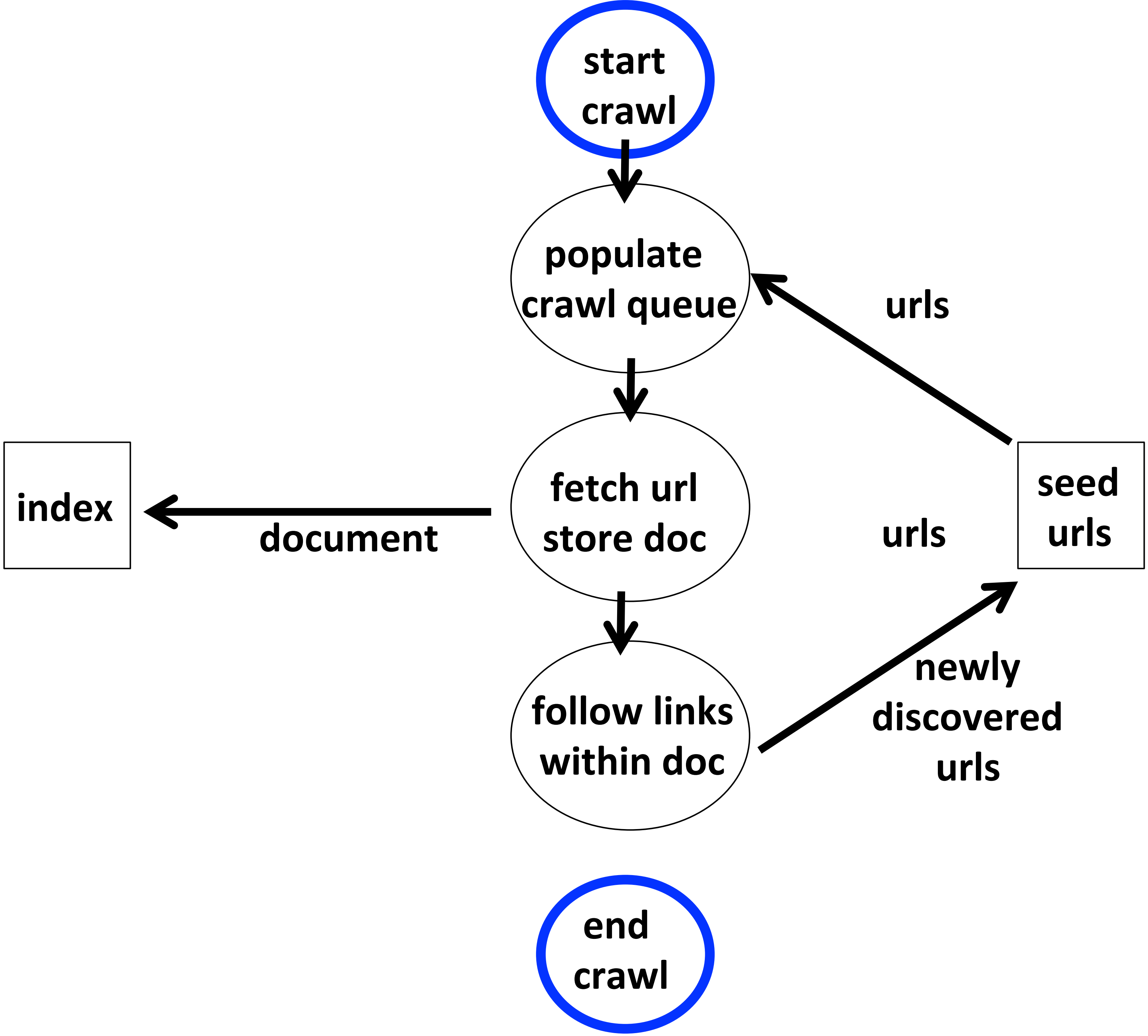
populate crawl queue

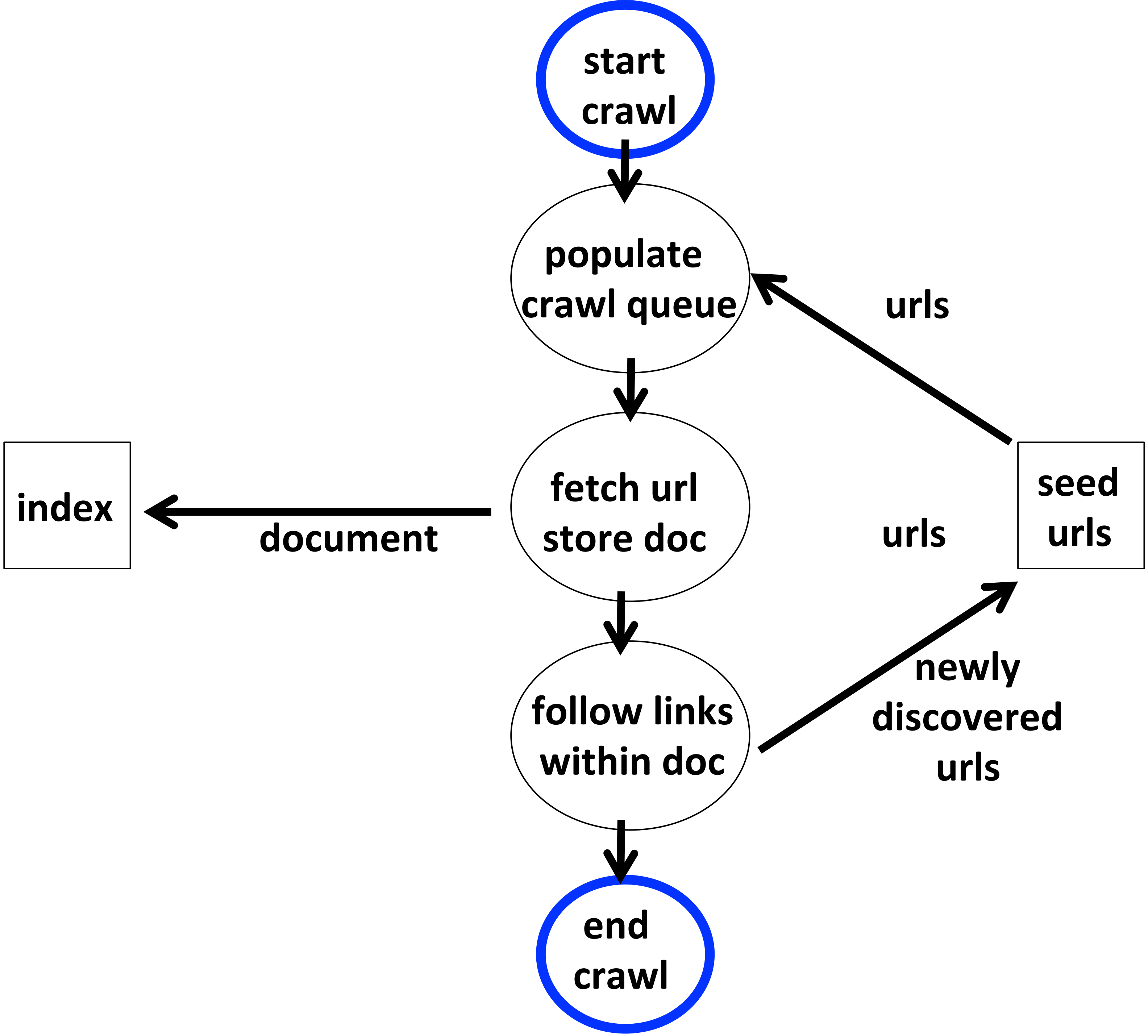
urls

seed urls

end crawl


start crawl
populate crawl queue
urls
fetch url store doc
index
document
seed urls
end crawl


start crawl
populate crawl queue
urls
fetch url store doc
index
document
urls
seed urls
follow links within doc
newly discovered urls
end crawl



start crawl

populate crawl queue

urls

fetch url store doc

index

document

urls

seed urls

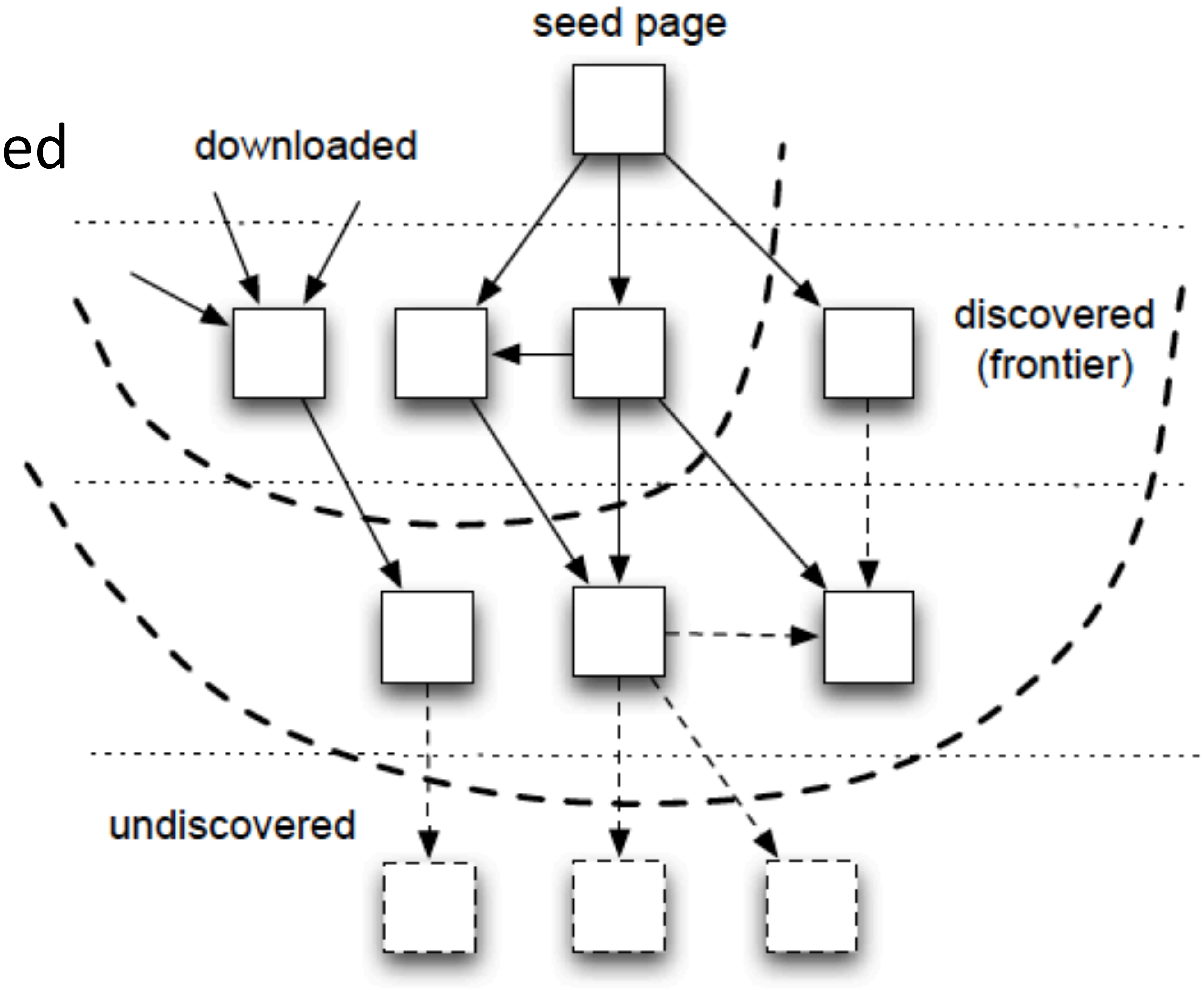
follow links within doc

newly discovered urls

end crawl

Crawling divides the web into 3 sets:

1. Downloaded
2. Discovered
3. Undiscovered



Crawl stops when crawl queue is exhausted & ***subject to policies***

Crawl stops when crawl queue is exhausted & ***subject to policies***

Crawling policies:

1. **Selection policy**: which urls to crawl
2. **Re-visit policy**: when to re-crawl the same url
3. **Politeness policy**: how aggressive the crawl is

Selection & Revisit are *URL Prioritisation Policies*

A crawler can only download tiny fraction of webpages each time, so it needs to *prioritise its downloads*

Selection & Revisit are *URL Prioritisation Policies*

A crawler can only download tiny fraction of webpages each time, so it needs to *prioritise its downloads*

A crawler maintains two separate queues for prioritizing the download of URLs

- **Discovery queue** (selection policy):
- **Refreshing queue** (re-visit policy)

Selection & Revisit are *URL Prioritisation Policies*

A crawler can only download tiny fraction of webpages each time, so it needs to *prioritise its downloads*

A crawler maintains two separate queues for prioritizing the download of URLs

- **Discovery queue** (selection policy):
  - downloads pages pointed by already discovered links
  - tries to increase ***coverage***
- **Refreshing queue** (re-visit policy)

Selection & Revisit are *URL Prioritisation Policies*

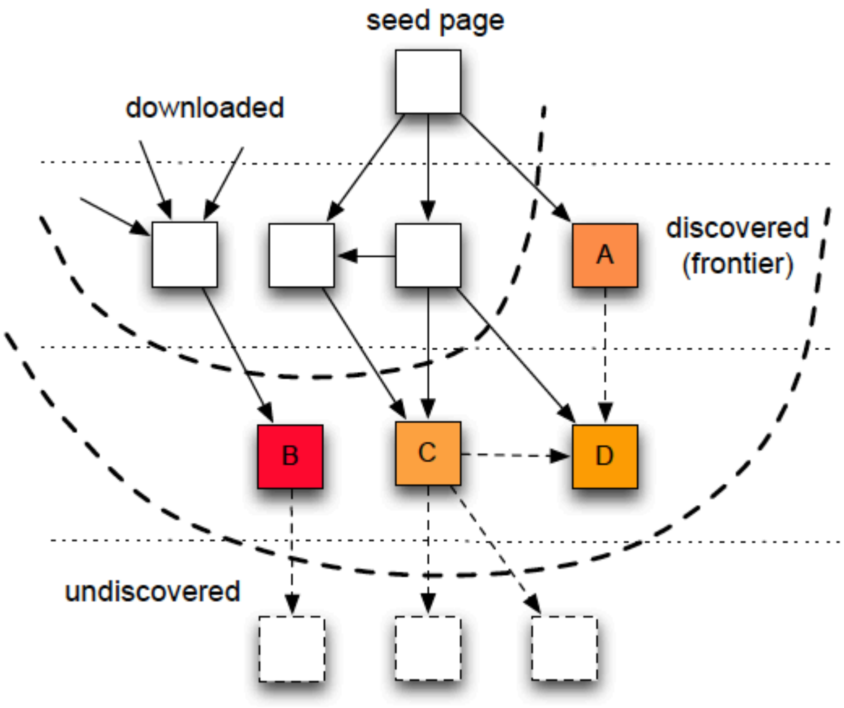
A crawler can only download tiny fraction of webpages each time, so it needs to *prioritise its downloads*

A crawler maintains two separate queues for prioritizing the download of URLs

- **Discovery queue** (selection policy):
  - downloads pages pointed by already discovered links
  - tries to increase ***coverage***
- **Refreshing queue** (re-visit policy)
  - Re-downloads already downloaded pages
  - Tries to increase ***freshness***

URL prioritization (Discovery):

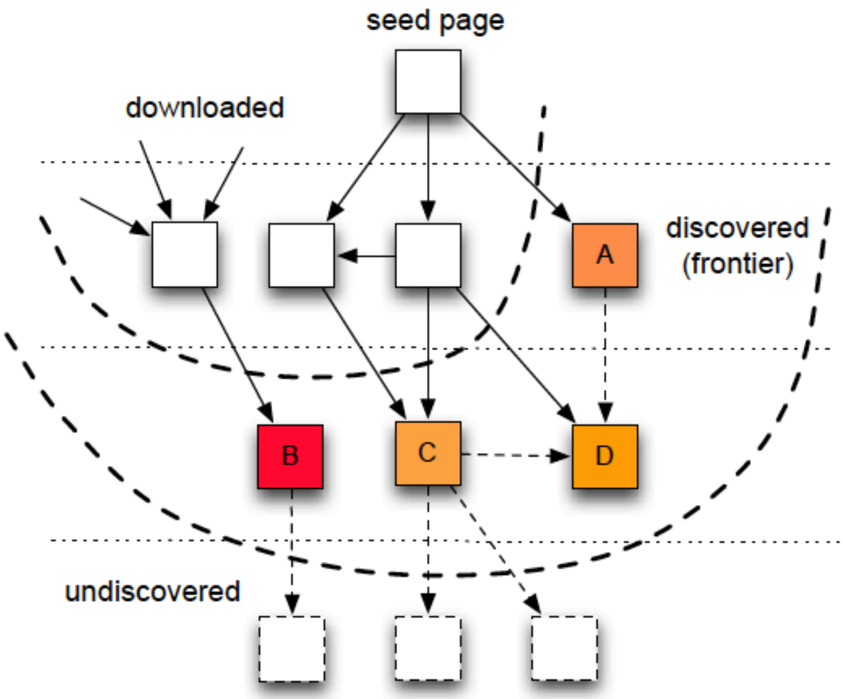
- Random (A,B,C,D)
- Breadth-first (A)
- In-degree (C)
- PageRank (B)



(more intense red color indicates higher PageRank)

URL prioritization (Discovery):

- Random (A,B,C,D)
- Breadth-first (A)
- In-degree (C)
- PageRank (B)



(more intense red color indicates higher PageRank)

URL prioritization (Refreshing):

- Random
- PageRank
- User feedback/interest
- Age
- Longevity

Why focus on ***discovery*** and ***refreshing***?

- Because the web is ***dynamic***: crawling tiny fraction can take months. By the time crawling is done, new information has been added, updated or deleted. Need to re-crawl.

Why focus on ***discovery*** and ***refreshing***?

- Because the web is ***dynamic***: crawling tiny fraction can take months. By the time crawling is done, new information has been added, updated or deleted. Need to re-crawl.
- Because for a search engine there is a ***cost*** associated with missing webpages or having outdated information

Why focus on ***discovery*** and ***refreshing***?

- Because the web is ***dynamic***: crawling tiny fraction can take months. By the time crawling is done, new information has been added, updated or deleted. Need to re-crawl.
- Because for a search engine there is a ***cost*** associated with missing webpages or having outdated information

  – Crawling metrics measure this cost

Coverage & Freshness are crawling metrics

Coverage & Freshness are crawling metrics

Quality metrics:

Performance metric:

Coverage & Freshness are crawling metrics

Quality metrics:

- Coverage: % of the Web discovered or downloaded by the crawler
- Freshness: measure of staleness of the local copy a page relative to the page's copy on the Web

Performance metric:

Coverage & Freshness are crawling metrics

Quality metrics:

- Coverage: % of the Web discovered or downloaded by the crawler
- Freshness: measure of staleness of the local copy a page relative to the page’s copy on the Web

Performance metric:

- Throughput: content download rate in bytes per unit of time

Example of Freshness

Freshness *F* of a webpage *p* stored in the index at time *t* (binary measure)

$$F_p(t) = \begin{cases} 1, \text{ if } p \text{ is equal to the stored copy at time } t \\ 0, \text{ otherwise} \end{cases}$$

Example of Freshness

Freshness *F* of a webpage *p* stored in the index at time *t* (binary measure)

$$F_p(t) = \begin{cases} 1, \text{ if } p \text{ is equal to the stored copy at time } t \\ 0, \text{ otherwise} \end{cases}$$

Age *A* of a webpage *p* stored in the index at time *t*

$$A_p(t) = \begin{cases} 0, \text{ if } p \text{ is not modified} \\ t - \text{modification time of } p\text{, otherwise} \end{cases}$$

Example of Freshness

Freshness *F* of a webpage *p* stored in the index at time *t* (binary measure) **(1 is best)**

$$F_p(t) = \begin{cases} 1, \text{ if } p \text{ is equal to the stored copy at time } t \\ 0, \text{ otherwise} \end{cases}$$

Age *A* of a webpage *p* stored in the index at time *t* **(0 is best)**

$$A_p(t) = \begin{cases} 0, \text{ if } p \text{ is not modified} \\ t - \text{modification time of } p, \text{ otherwise} \end{cases}$$

Crawling politeness policy: how aggressive the crawl is

Crawling politeness policy: how aggressive the crawl is

Why? Crawlers get data very fast & in great depth → **crippling impact on website performance** e.g. if crawler sends multiple requests per sec, or downloads large files, server may not keep up with user requests

Crawling politeness policy: how aggressive the crawl is

Why? Crawlers get data very fast & in great depth → **crippling impact on website performance** e.g. if crawler sends multiple requests per sec, or downloads large files, server may not keep up with user requests

- **Network resources**: considerable bandwidth for a long period of time
- **Server overload**: if frequency of accesses to server is high

Crawling politeness policy: how aggressive the crawl is

Why? Crawlers get data very fast & in great depth → **crippling impact on website performance** e.g. if crawler sends multiple requests per sec, or downloads large files, server may not keep up with user requests

- **Network resources**: considerable bandwidth for a long period of time
- **Server overload**: if frequency of accesses to server is high

Poorly written crawlers may crash servers or routers or may download webpages they cannot handle

Partial solutions:

- A polite crawler puts a delay between two consecutive downloads from the same server (common: 20 seconds)

Partial solutions:

- A polite crawler puts a delay between two consecutive downloads from the same server (common: 20 seconds)
- A polite crawler closes the connection after the webpage is downloaded from the server

Partial solutions:

- A polite crawler puts a delay between two consecutive downloads from the same server (common: 20 seconds)
- A polite crawler closes the connection after the webpage is downloaded from the server
- A polite crawler respects the ***robots exclusion protocol***
  - Created by webpage administrators to indicate which parts of their servers should and/or should not be crawled
  - robots.txt: standard from the early days of the web

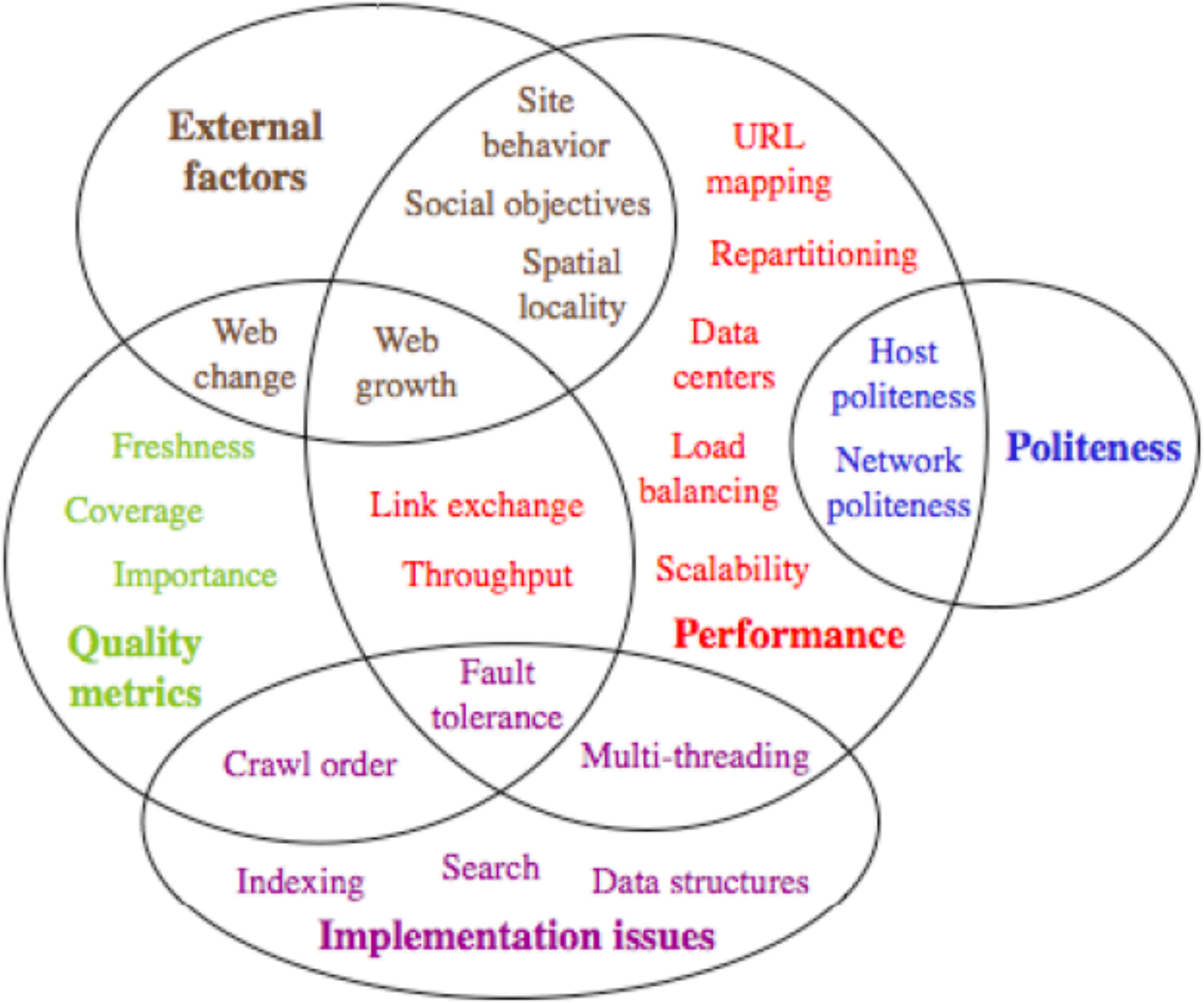
Partial solutions:

- A polite crawler puts a delay between two consecutive downloads from the same server (common: 20 seconds)
- A polite crawler closes the connection after the webpage is downloaded from the server
- A polite crawler respects the ***robots exclusion protocol***
  - Created by webpage administrators to indicate which parts of their servers should and/or should not be crawled
  - robots.txt: standard from the early days of the web
  - Crawlers often cache robots.txt files for efficiency

How Google handles such exclusion protocols (in BNF):

https://developers.google.com/webmasters/control-crawl-index/docs/robots_txt

Concepts related to web crawling



Crawlers: central part of search engines

- Details of their algorithms & architecture are kept as **business secrets** (lack of detail in published designs)

Why?

- **Competition**: prevent others to reproduce the work
- **Spamming risks**: emerging concerns about spammers taking advantage of the crawling process to spread spam

http://www.google.com/insidesearch/howsearchworks/crawling-indexing.html

Crawling architectures

- **Single computer**

- **Parallel**

- **Geographically distributed**

Crawling architectures

- **Single computer**
  - CPU, RAM, and disk becomes bottleneck
  - Not scalable
- **Parallel**
- **Geographically distributed**

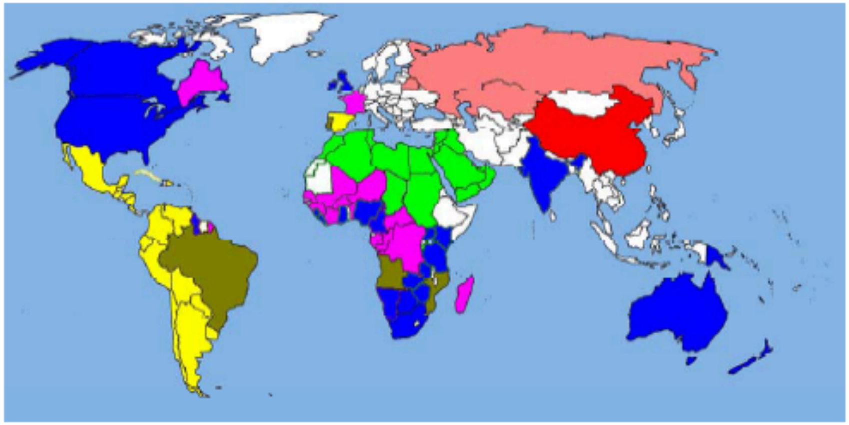
Crawling architectures

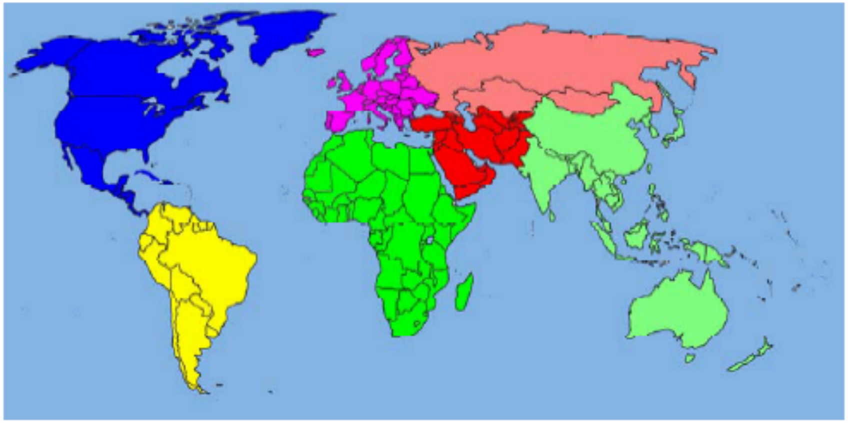
- **Single computer**
  - CPU, RAM, and disk becomes bottleneck
  - Not scalable
- **Parallel**
  - Multiple computers, single data centre
  - Scalable
- **Geographically distributed**

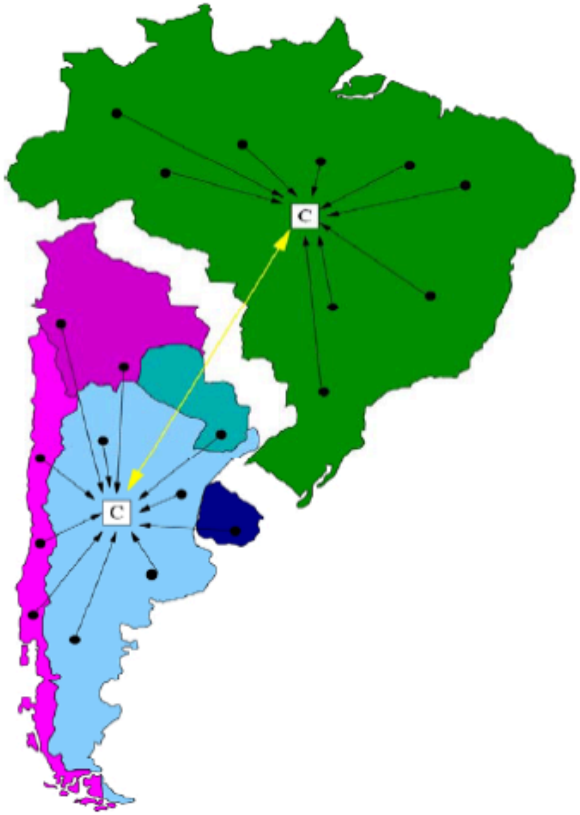
Crawling architectures

- **Single computer**
  - CPU, RAM, and disk becomes bottleneck
  - Not scalable
- **Parallel**
  - Multiple computers, single data centre
  - Scalable
- **Geographically distributed**
  - Multiple computers, multiple data centres
  - Scalable
  - Reduces network latency

Geographically distributed Web crawling







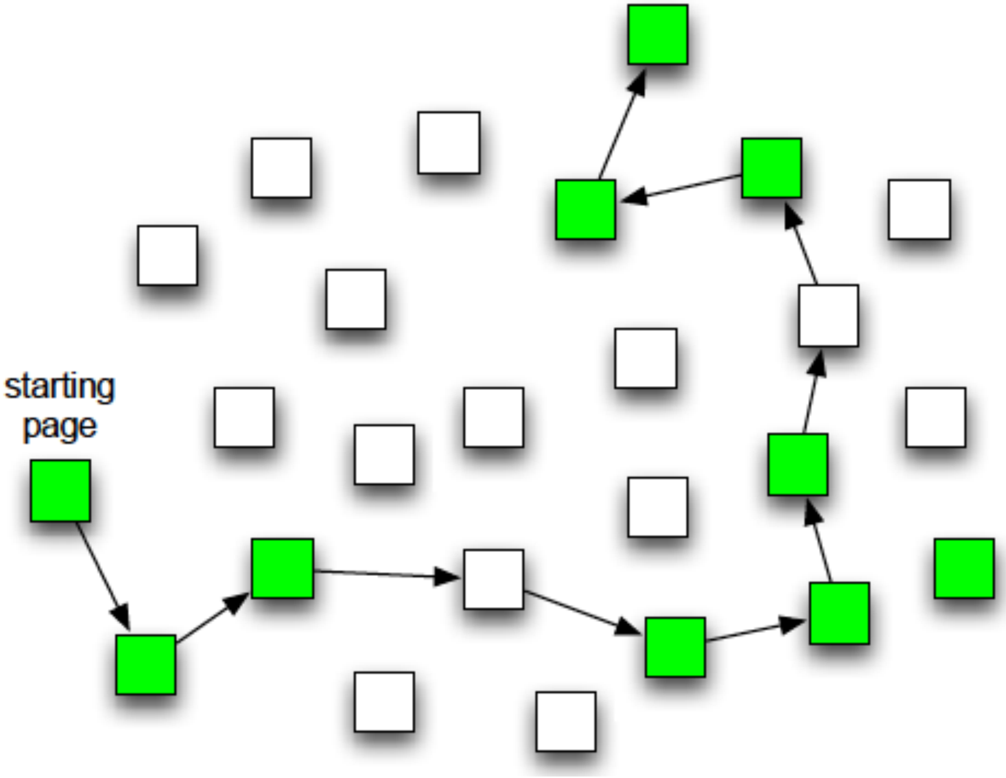
Geographically distributed Web crawling

Benefits:

- Higher crawling throughput
  - Geographical proximity
  - Lower crawling latency
- Increased availability
  - Continuity of business
- Better coupling with distributed indexing/search
  - Reduced data migration

Focused Web Crawling

Goal: locate and download a large proportion of web pages that match a given target theme as early as possible



Focused Web Crawling

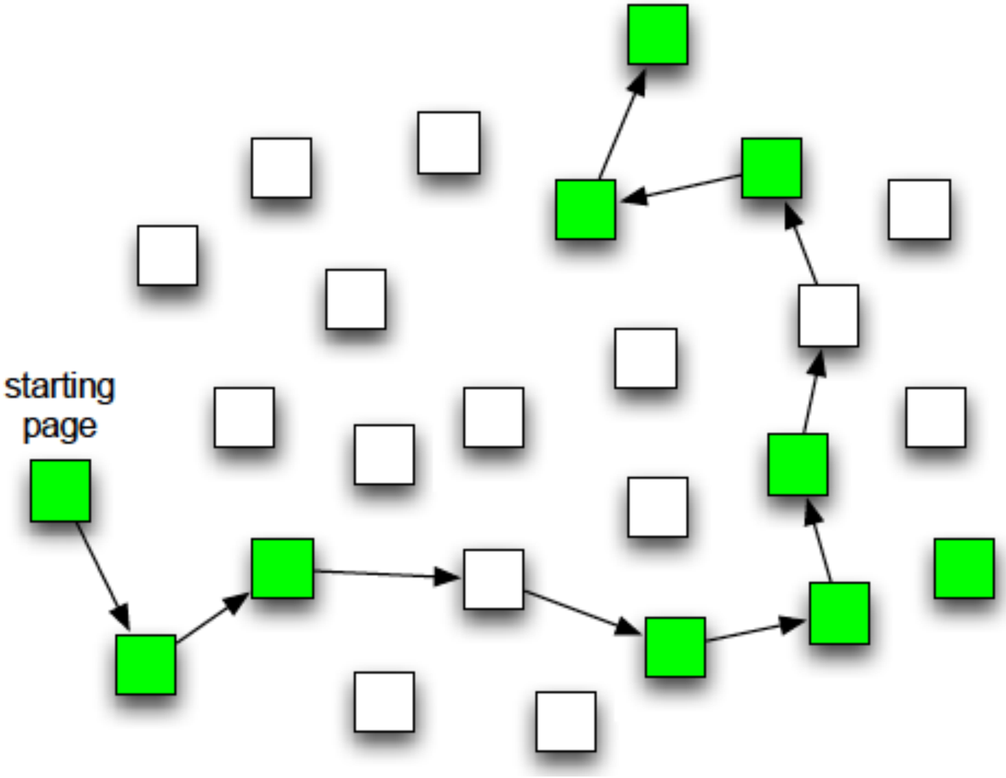
Goal: locate and download a large proportion of web pages that match a given target theme as early as possible

Example themes:

- Topic (nuclear energy)
- Media type (forums)
- Demographics (kids)

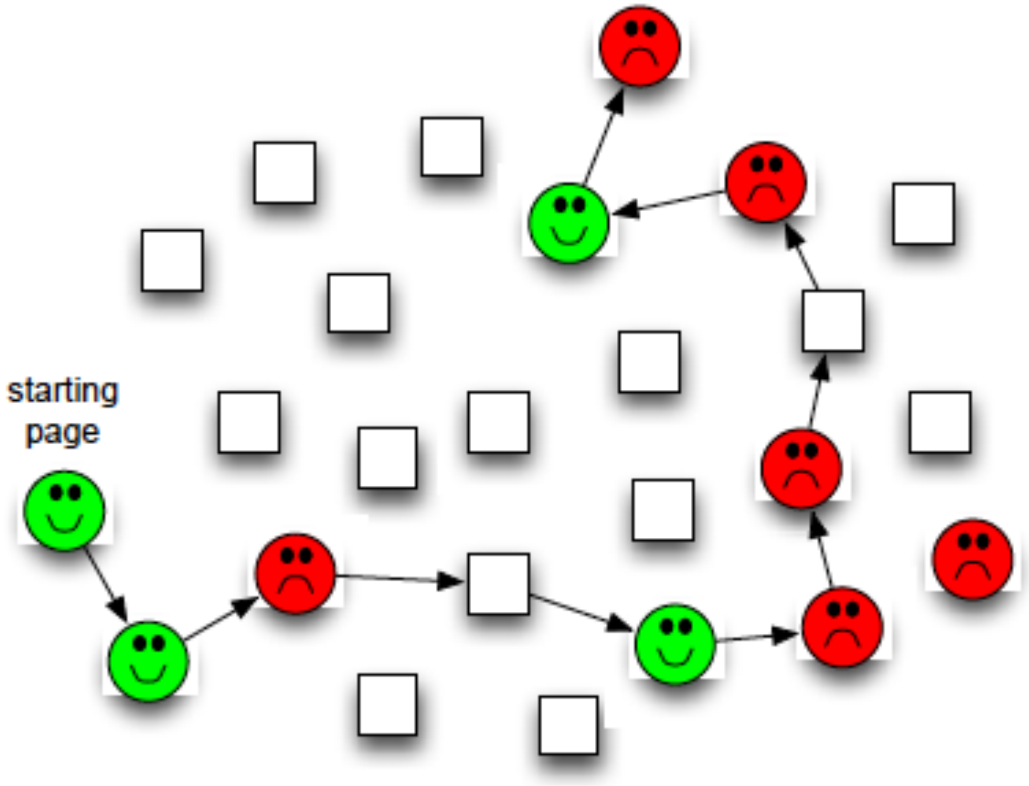
Strategies:

- URL patterns
- Referring page content
- Local graph structure



Sentiment Focused Web Crawling

Goal: locate and download a large proportion of web pages that contain positive or negative sentiments (opinionated content) as early as possible



Research Problem: Hidden Web Crawling

Hidden Web: web pages that a crawler cannot access by simply following link structure

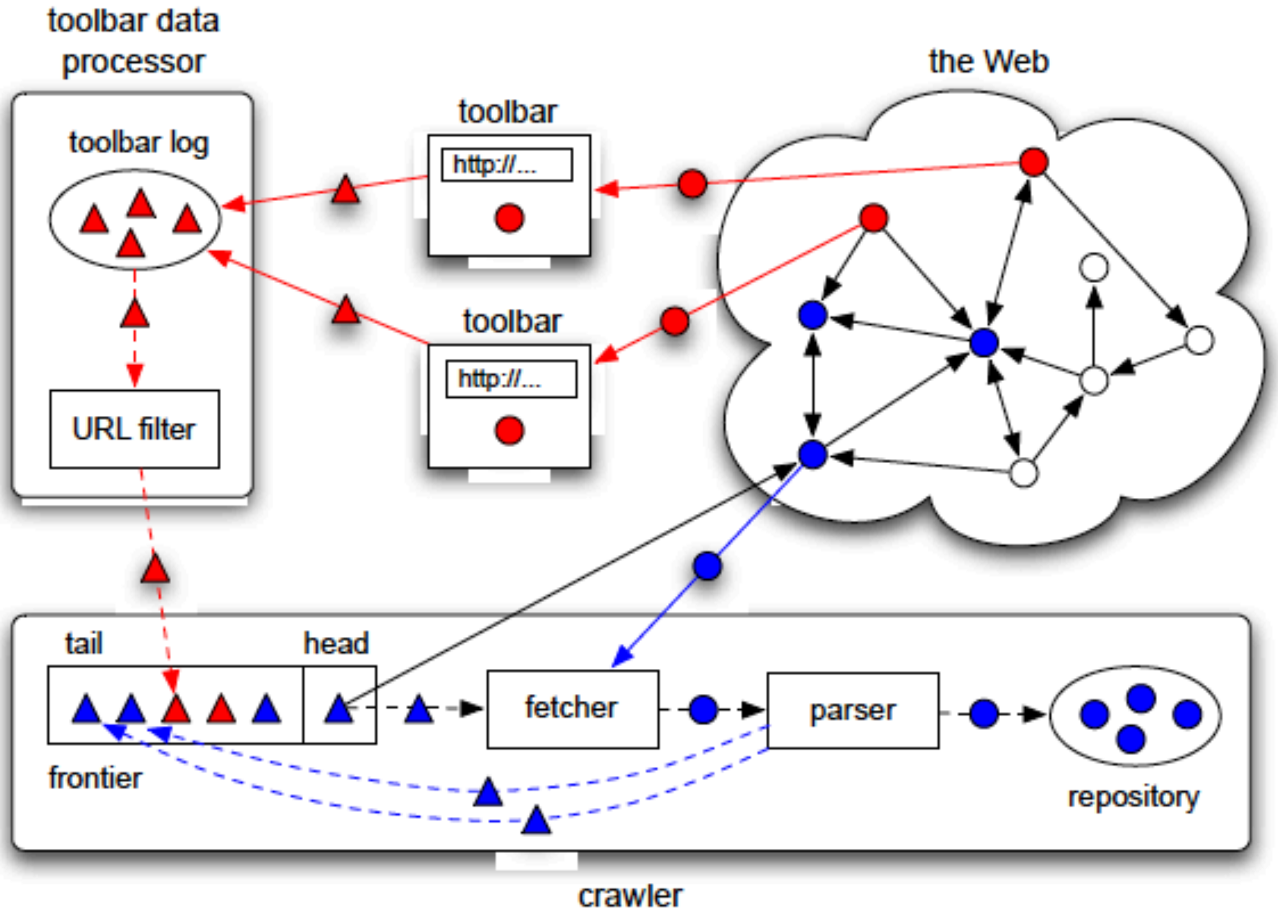
Examples:

- Unlinked pages
- Private sites
- Scripted content
- Dynamic content
- …

Hidden Web Crawling → Passive discovery

URL discovery by external agents: toolbar logs, email messages, tweets, …

Benefits: improved coverage, early discovery



Published web crawler architectures

- Bingbot: Microsoft’s Bing web crawler
- FAST craweler: Used by Fast Search & Transfer
- Googlebot: Web crawler of Google
- PolyBot: a distributed web crawler
- RBSE: The first published web crawler
- WebFountain: A distributed web crawler
- Web RACE: a crawling and caching module
- Yahoo Slurp: web crawler used by Yahoo search

Open source web crawlers

- DataparkSearch: GNU General Public License (GPL)
- GRUB: open source distributed crawler of Wikia Search
- Heritrix: Internet Archives crawler
- ICDL Crawler: cross-platform web crawler
- Norconex HTTP Collector: licensed under GPL
- Nutch: Apache License
- Open Search Server: GPL License
- PHP-Crawler: BSD license
- Scrapy: BSD license
- Seeks: Affero GPL

Today's lecture

- Course administration

- What is Web Science
- What is the Web
- What is the Internet
- Web graph
- Main challenges of web data processing
- Web crawling

References (in addition to Absalon readings) & sources:

- Chapter 13 from the book *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. By David Easley and Jon Kleinberg. Cambridge University Press, 2010.
  Complete preprint on-line at http://www.cs.cornell.edu/home/kleinber/networks-book/
- Chapter 1 from the book *Big Data Analytics Methods: Modern Analytics Techniques for the 21st Century*. By Peter Ghavami. Amazon, 2016.
- S. Lawrence and C. L. Giles. *Accessibility of Information on the Web*. Nature, 400, 107-109, 1999.
- T. Berners-Lee, W. Hall, J. A. Hendler, K. O'Hara, N. Shadbolt and D. J. Weitzner. "*A Framework for Web Science*", Foundations and Trends® in Web Science: Vol. 1: No. 1, pp 1-130, 2006.
- All pictures retrieved with Google for noncommercial reuse

Seminal readings on crawling:

Cho, Garcia-Molina, and Page, "Efficient crawling through URL ordering”, WWW, 1998.

Heydon and Najork, "Mercator: a scalable, extensible web crawler", WWW, 1999.

Chakrabarti, van den Berg, and Dom, "Focused crawling: a new approach to topic-specific web resource discovery", Computer Networks, 1999.

Najork and Wiener, "Breadth-first crawling yields high-quality pages“, WWW, 2001.

Cho and Garcia-Molina, "Parallel crawlers", WWW, 2002.

Cho and Garcia-Molina, "Effective page refresh policies for web crawlers”, ACM TDS, 2003.

Lee, Leonard, Wang, and Loguinov, "IRLbot: Scaling to 6 billion pages and beyond", ACM TWEB, 2009.