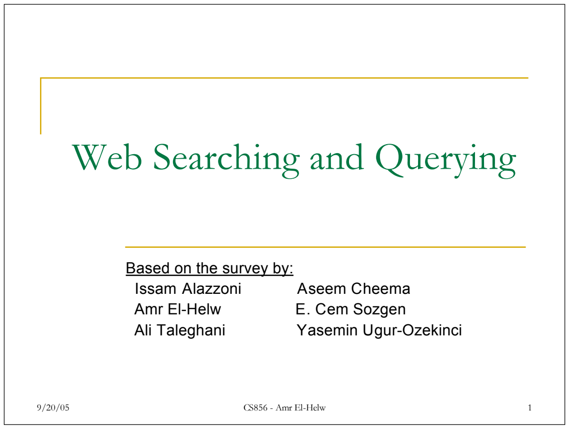
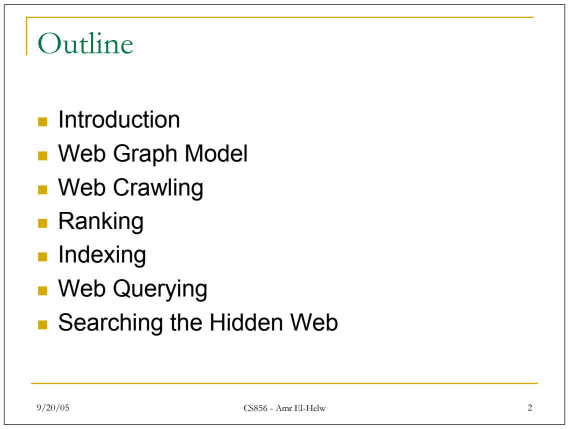


Web Searching and Querying

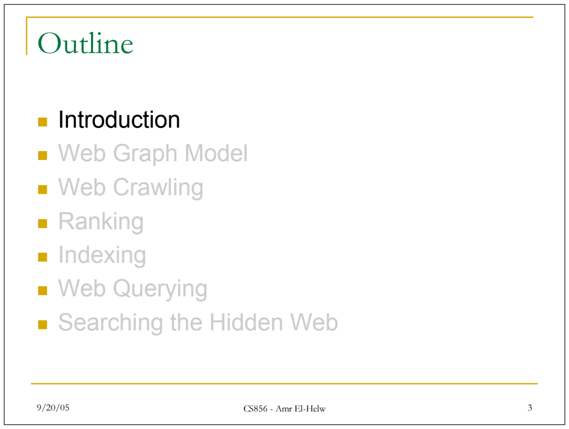
Based on the survey by:

Issam Alazzoni Aseem Cheema
Amr El-Helw E. Cem Sozgen
Ali Taleghani Yasemin Ugur-Ozekinci



Outline

- Introduction
- Web Graph Model
- Web Crawling
- Ranking
- Indexing
- Web Querying
- Searching the Hidden Web



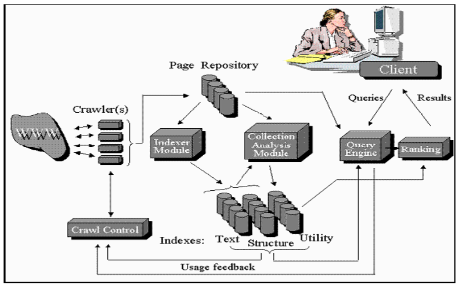
Outline

- Introduction
- Web Graph Model
- Web Crawling
- Ranking
- Indexing
- Web Querying
- Searching the Hidden Web

Introduction

- The Web has more than 3 billion HTML pages.
- Most Internet users gain access to the Web using search engines.
- 23% of Web pages change daily [3].
- 40% of commercial pages change daily [3].

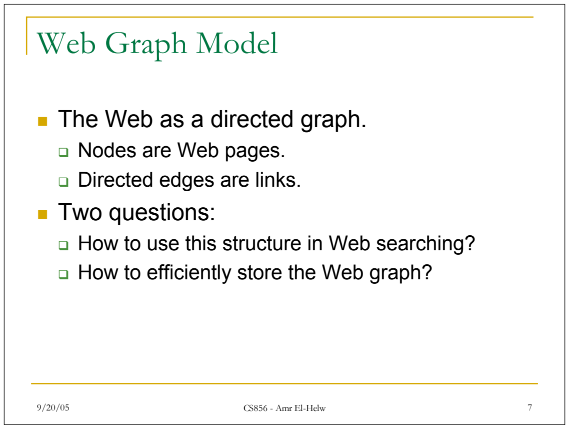
Introduction



Search Engine Architecture [3]

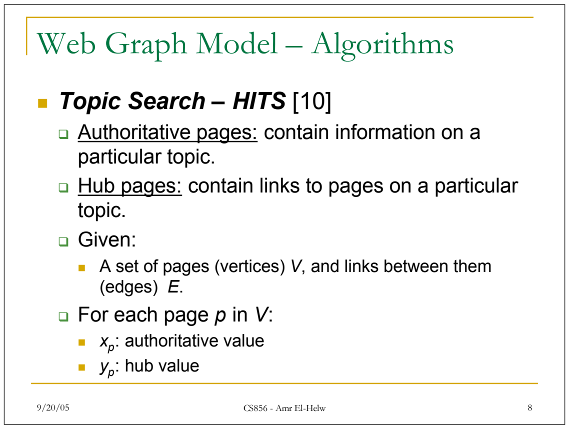
Outline

- Introduction
- Web Graph Model
- Web Crawling
- Ranking
- Indexing
- Web Querying
- Searching the Hidden Web



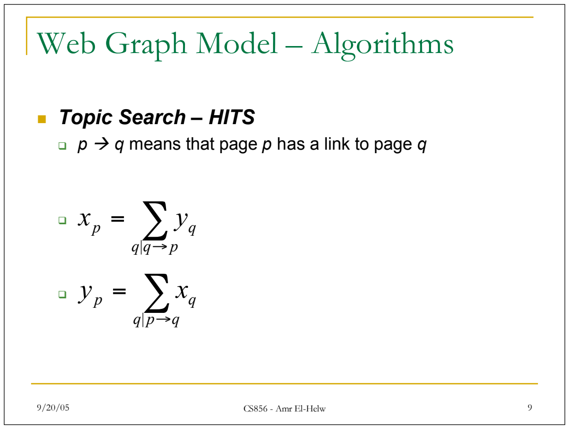
Web Graph Model

- The Web as a directed graph.
  - Nodes are Web pages.
  - Directed edges are links.
- Two questions:
  - How to use this structure in Web searching?
  - How to efficiently store the Web graph?



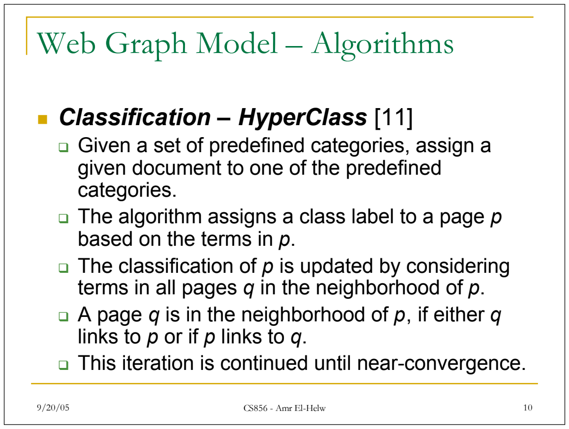
Web Graph Model – Algorithms

- ***Topic Search – HITS*** [10]
  - Authoritative pages: contain information on a particular topic.
  - Hub pages: contain links to pages on a particular topic.
  - Given:
    - A set of pages (vertices) *V*, and links between them (edges) *E*.
  - For each page *p* in *V*:
    - x_p: authoritative value
    - y_p: hub value



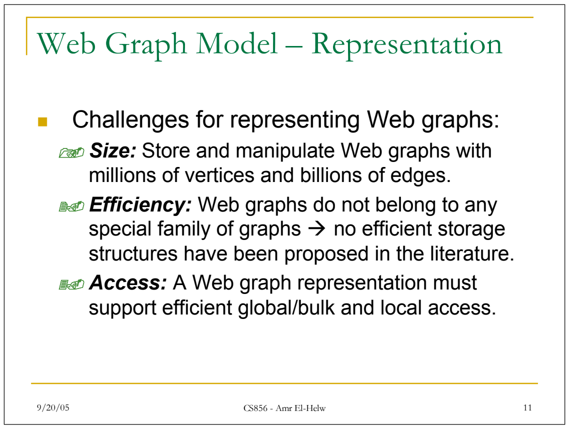
Web Graph Model – Algorithms

- ***Topic Search – HITS***
  - *p* → *q* means that page *p* has a link to page *q*
  - $x_p = \sum_{q|q \to p} y_q$
  - $y_p = \sum_{q|p \to q} x_q$



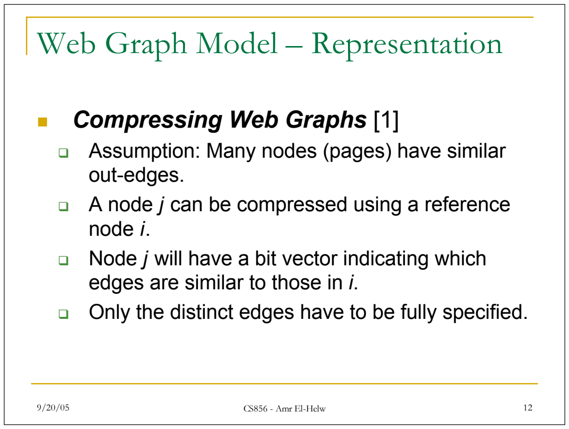
Web Graph Model – Algorithms

- ***Classification – HyperClass*** [11]
  - Given a set of predefined categories, assign a given document to one of the predefined categories.
  - The algorithm assigns a class label to a page *p* based on the terms in *p*.
  - The classification of *p* is updated by considering terms in all pages *q* in the neighborhood of *p*.
  - A page *q* is in the neighborhood of *p*, if either *q* links to *p* or if *p* links to *q*.
  - This iteration is continued until near-convergence.



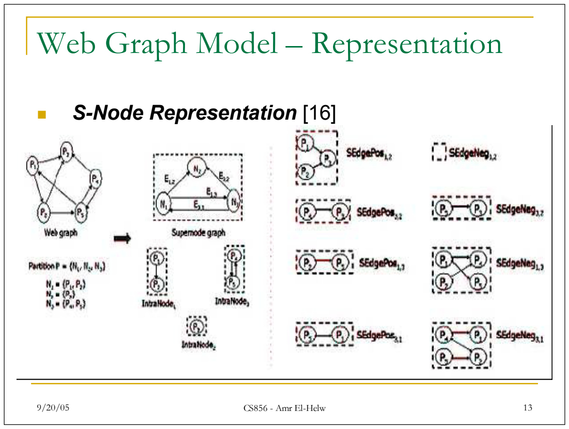
Web Graph Model – Representation

- Challenges for representing Web graphs:
  - ***Size:*** Store and manipulate Web graphs with millions of vertices and billions of edges.
  - ***Efficiency:*** Web graphs do not belong to any special family of graphs → no efficient storage structures have been proposed in the literature.
  - ***Access:*** A Web graph representation must support efficient global/bulk and local access.



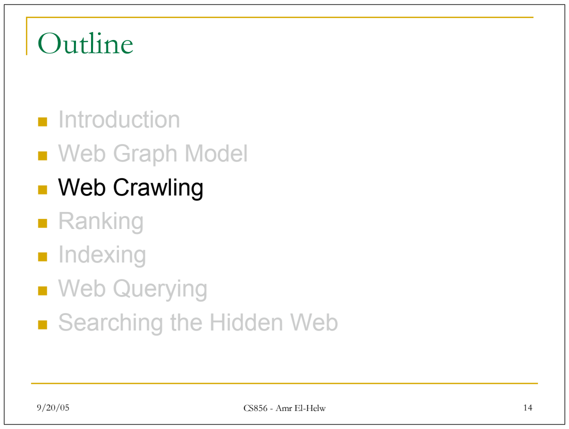
Web Graph Model – Representation

- ***Compressing Web Graphs*** [1]
  - Assumption: Many nodes (pages) have similar out-edges.
  - A node *j* can be compressed using a reference node *i*.
  - Node *j* will have a bit vector indicating which edges are similar to those in *i*.
  - Only the distinct edges have to be fully specified.



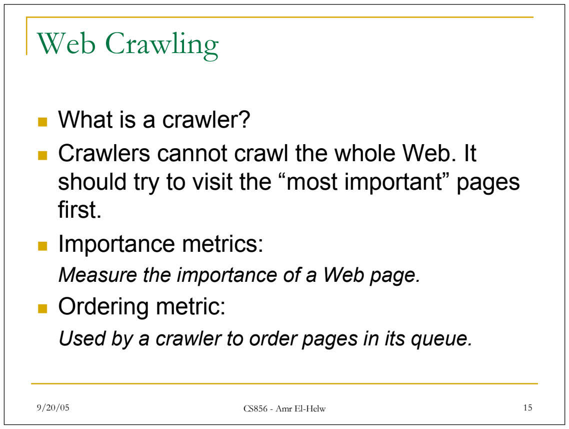
Web Graph Model – Representation

- ***S-Node Representation*** [16]



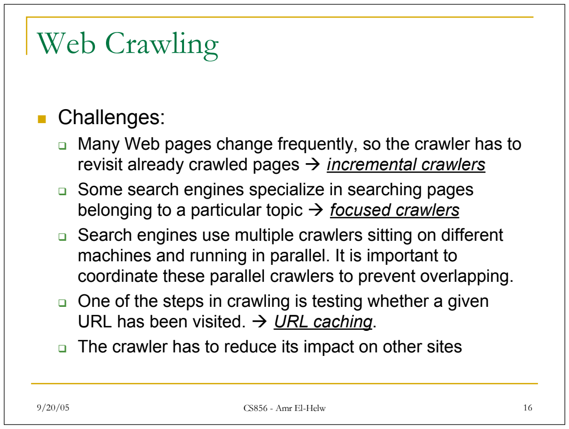
Outline

- Introduction
- Web Graph Model
- Web Crawling
- Ranking
- Indexing
- Web Querying
- Searching the Hidden Web



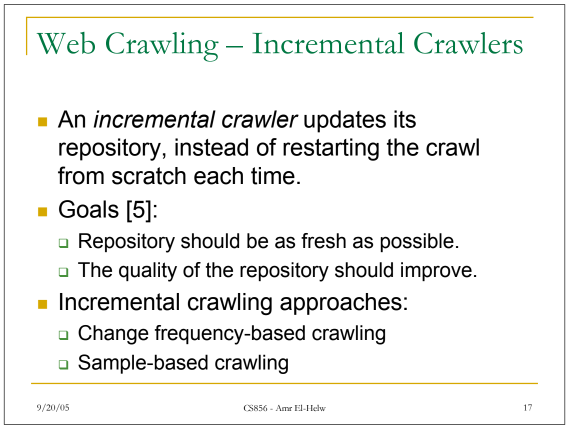
Web Crawling

- What is a crawler?
- Crawlers cannot crawl the whole Web. It should try to visit the "most important" pages first.
- Importance metrics:
  Measure the importance of a Web page.
- Ordering metric:
  Used by a crawler to order pages in its queue.



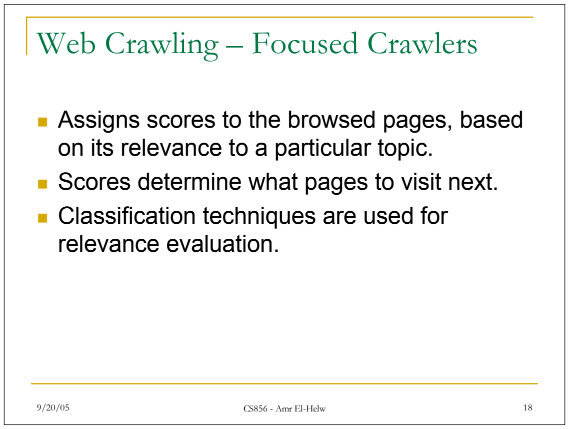
Web Crawling

- Challenges:
  - Many Web pages change frequently, so the crawler has to revisit already crawled pages → *incremental crawlers*
  - Some search engines specialize in searching pages belonging to a particular topic → *focused crawlers*
  - Search engines use multiple crawlers sitting on different machines and running in parallel. It is important to coordinate these parallel crawlers to prevent overlapping.
  - One of the steps in crawling is testing whether a given URL has been visited. → *URL caching*.
  - The crawler has to reduce its impact on other sites



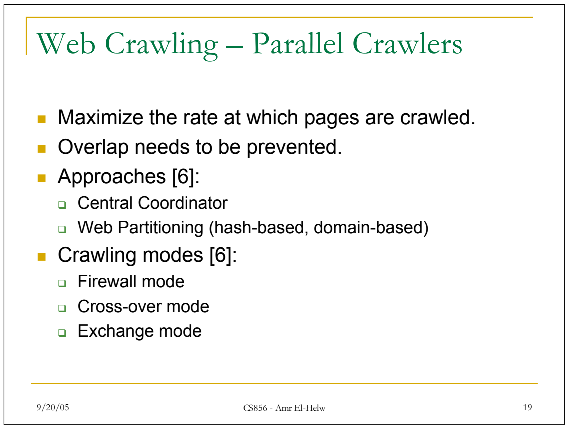
Web Crawling – Incremental Crawlers

- An *incremental crawler* updates its repository, instead of restarting the crawl from scratch each time.
- Goals [5]:
  - Repository should be as fresh as possible.
  - The quality of the repository should improve.
- Incremental crawling approaches:
  - Change frequency-based crawling
  - Sample-based crawling



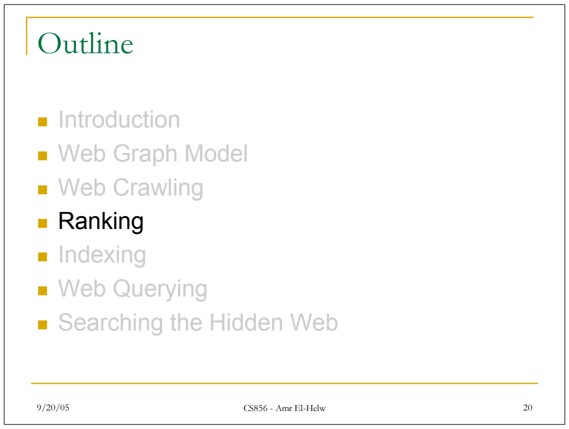
Web Crawling – Focused Crawlers

- Assigns scores to the browsed pages, based on its relevance to a particular topic.
- Scores determine what pages to visit next.
- Classification techniques are used for relevance evaluation.



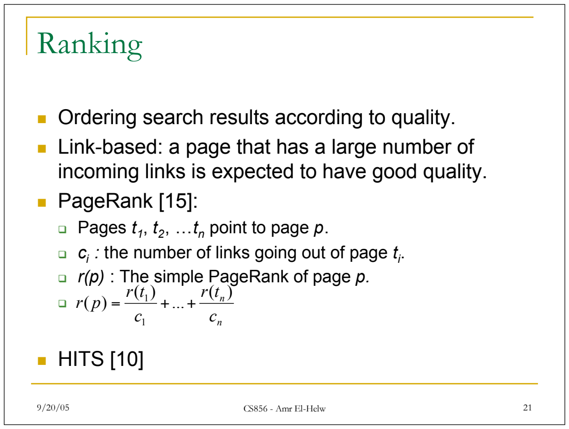
Web Crawling – Parallel Crawlers

- Maximize the rate at which pages are crawled.
- Overlap needs to be prevented.
- Approaches [6]:
  - Central Coordinator
  - Web Partitioning (hash-based, domain-based)
- Crawling modes [6]:
  - Firewall mode
  - Cross-over mode
  - Exchange mode



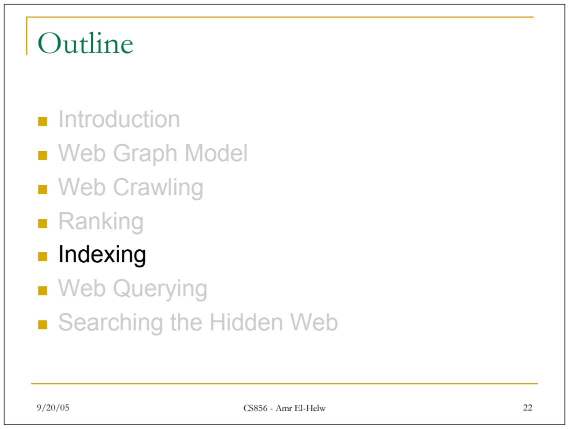
Outline

- Introduction
- Web Graph Model
- Web Crawling
- Ranking
- Indexing
- Web Querying
- Searching the Hidden Web



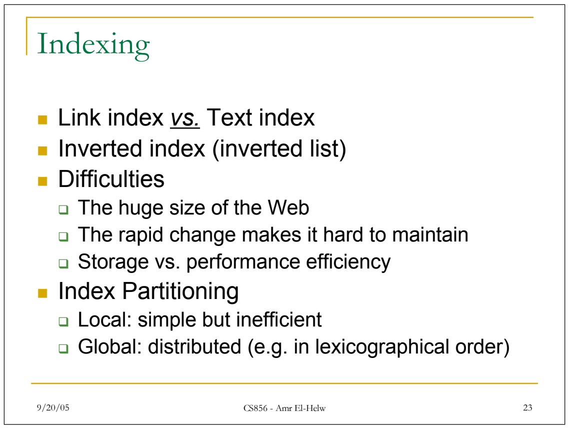
Ranking

- Ordering search results according to quality.
- Link-based: a page that has a large number of incoming links is expected to have good quality.
- PageRank [15]:
  - Pages t_1, t_2, …t_n point to page p.
  - c_i : the number of links going out of page t_i.
  - $r(p)$: The simple PageRank of page p.
  - $r(p) = \frac{r(t_1)}{c_1} + \ldots + \frac{r(t_n)}{c_n}$
- HITS [10]



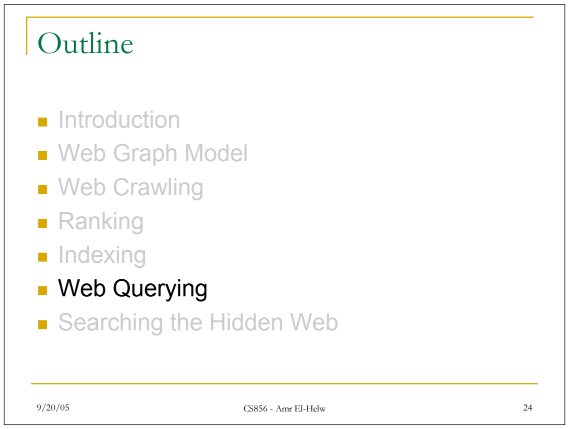
Outline

- Introduction
- Web Graph Model
- Web Crawling
- Ranking
- Indexing
- Web Querying
- Searching the Hidden Web



Indexing

- Link index *vs.* Text index
- Inverted index (inverted list)
- Difficulties
  - The huge size of the Web
  - The rapid change makes it hard to maintain
  - Storage vs. performance efficiency
- Index Partitioning
  - Local: simple but inefficient
  - Global: distributed (e.g. in lexicographical order)



Outline

- Introduction
- Web Graph Model
- Web Crawling
- Ranking
- Indexing
- Web Querying
- Searching the Hidden Web

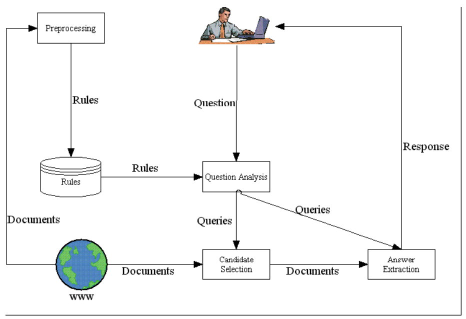
Web Querying

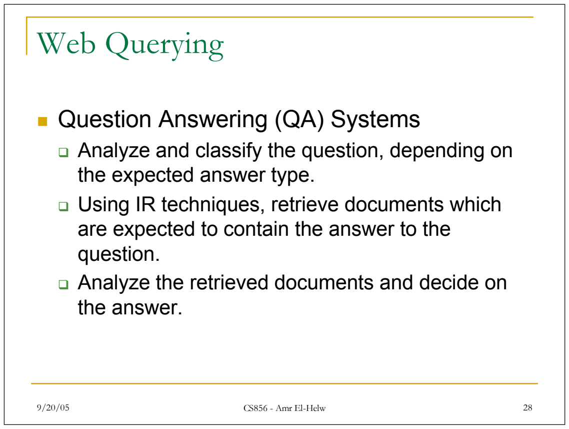
- Why Web Querying?
  - It is not always easy to express information requests using keywords.
  - Search engines do not make use of *Web topology* and *document structure* in queries.
- Early Web Query Approaches
  - Structured (Similar to DBMSs): Data model + Query Language
  - Semi-structured: e.g. Object Exchange Model (OEM)

Web Querying

- Question Answering (QA) Systems
  - Finding answers to natural language questions, e.g. *What is Computer?*
  - Analyze the question and try to guess what type of information that is required.
  - Not only locate relevant documents but also extract answers from them.
  - Examples: WebQA [14], Mulder [12], Tritus [2] and Start [9].

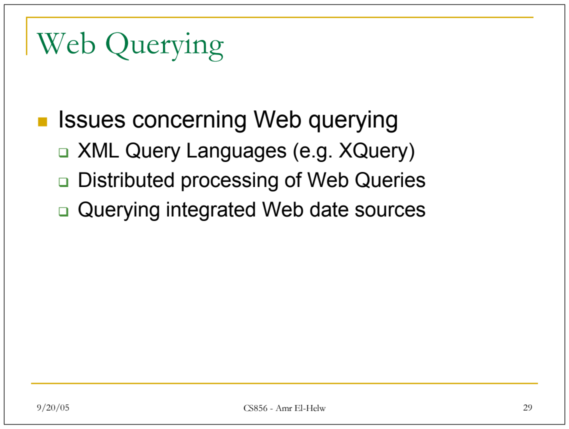
Web Querying





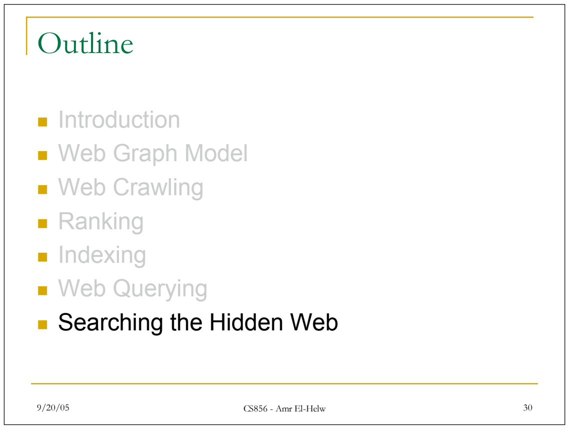
Web Querying

- Question Answering (QA) Systems
  - Analyze and classify the question, depending on the expected answer type.
  - Using IR techniques, retrieve documents which are expected to contain the answer to the question.
  - Analyze the retrieved documents and decide on the answer.



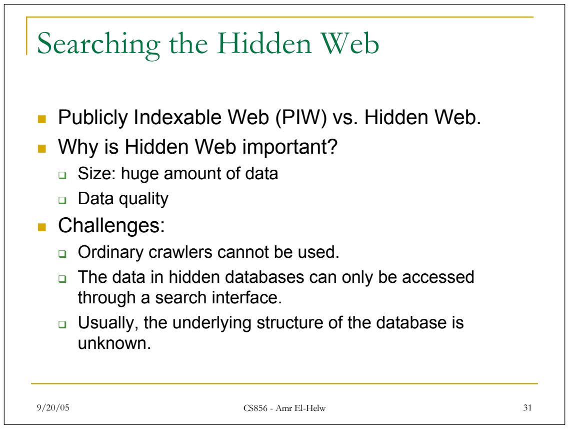
Web Querying

- Issues concerning Web querying
  - XML Query Languages (e.g. XQuery)
  - Distributed processing of Web Queries
  - Querying integrated Web date sources



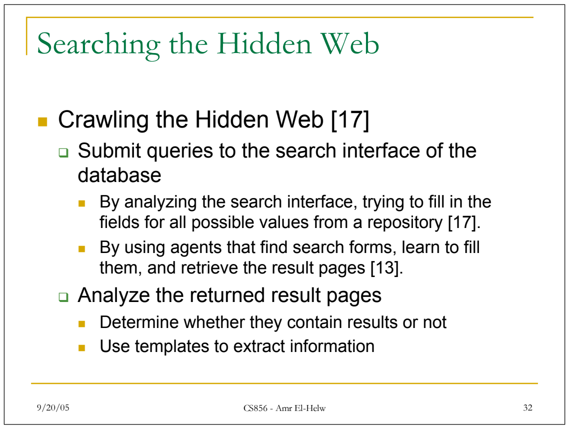
Outline

- Introduction
- Web Graph Model
- Web Crawling
- Ranking
- Indexing
- Web Querying
- Searching the Hidden Web



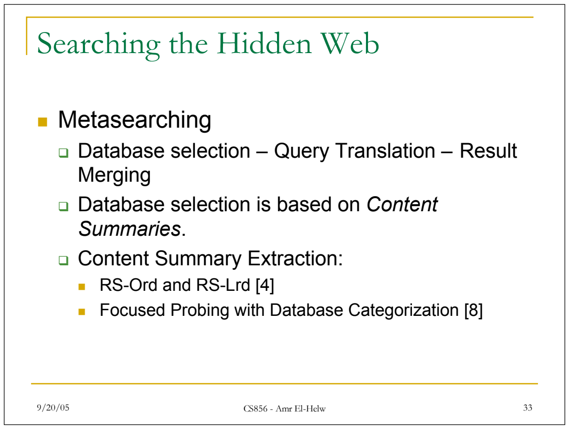
Searching the Hidden Web

- Publicly Indexable Web (PIW) vs. Hidden Web.
- Why is Hidden Web important?
  - Size: huge amount of data
  - Data quality
- Challenges:
  - Ordinary crawlers cannot be used.
  - The data in hidden databases can only be accessed through a search interface.
  - Usually, the underlying structure of the database is unknown.



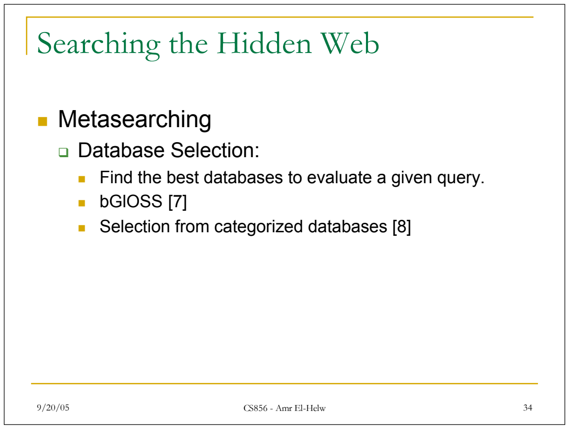
Searching the Hidden Web

- Crawling the Hidden Web [17]
  - Submit queries to the search interface of the database
    - By analyzing the search interface, trying to fill in the fields for all possible values from a repository [17].
    - By using agents that find search forms, learn to fill them, and retrieve the result pages [13].
  - Analyze the returned result pages
    - Determine whether they contain results or not
    - Use templates to extract information



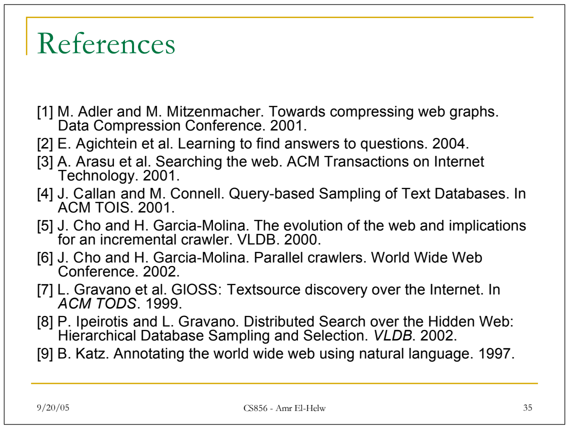
Searching the Hidden Web

- Metasearching
  - Database selection – Query Translation – Result Merging
  - Database selection is based on *Content Summaries*.
  - Content Summary Extraction:
    - RS-Ord and RS-Lrd [4]
    - Focused Probing with Database Categorization [8]



Searching the Hidden Web

- Metasearching
  - Database Selection:
    - Find the best databases to evaluate a given query.
    - bGlOSS [7]
    - Selection from categorized databases [8]



References

[1] M. Adler and M. Mitzenmacher. Towards compressing web graphs. Data Compression Conference. 2001.

[2] E. Agichtein et al. Learning to find answers to questions. 2004.

[3] A. Arasu et al. Searching the web. ACM Transactions on Internet Technology. 2001.

[4] J. Callan and M. Connell. Query-based Sampling of Text Databases. In ACM TOIS. 2001.

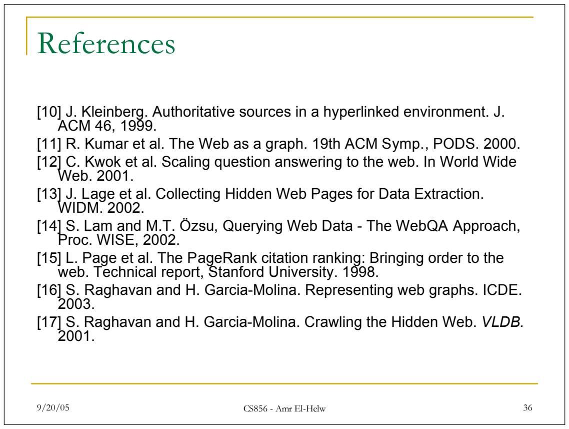
[5] J. Cho and H. Garcia-Molina. The evolution of the web and implications for an incremental crawler. VLDB. 2000.

[6] J. Cho and H. Garcia-Molina. Parallel crawlers. World Wide Web Conference. 2002.

[7] L. Gravano et al. GlOSS: Textsource discovery over the Internet. In *ACM TODS*. 1999.

[8] P. Ipeirotis and L. Gravano. Distributed Search over the Hidden Web: Hierarchical Database Sampling and Selection. *VLDB*. 2002.

[9] B. Katz. Annotating the world wide web using natural language. 1997.



References

[10] J. Kleinberg. Authoritative sources in a hyperlinked environment. J. ACM 46, 1999.

[11] R. Kumar et al. The Web as a graph. 19th ACM Symp., PODS. 2000.

[12] C. Kwok et al. Scaling question answering to the web. In World Wide Web. 2001.

[13] J. Lage et al. Collecting Hidden Web Pages for Data Extraction. WIDM. 2002.

[14] S. Lam and M.T. Özsu, Querying Web Data - The WebQA Approach, Proc. WISE, 2002.

[15] L. Page et al. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford University. 1998.

[16] S. Raghavan and H. Garcia-Molina. Representing web graphs. ICDE. 2003.

[17] S. Raghavan and H. Garcia-Molina. Crawling the Hidden Web. *VLDB*. 2001.