Question Answering Techniques for the World Wide Web

Jimmy Lin and Boris Katz
MIT Artificial Intelligence Laboratory

Tutorial presentation at
The 11th Conference of the European Chapter of the
Association of Computational Linguistics (EACL-2003)

April 12, 2003

Abstract

Question answering systems have become increasingly popular because they deliver users short, succinct answers instead of overloading them with a large number of irrelevant documents. The vast amount of information readily available on the World Wide Web presents new opportunities and challenges for question answering. In order for question answering systems to benefit from this vast store of useful knowledge, they must cope with large volumes of useless data.

Many characteristics of the World Wide Web distinguish Web-based question answering from question answering on closed corpora such as newspaper texts. The Web is vastly larger in size and boasts incredible "data redundancy," which renders it amenable to statistical techniques for answer extraction. A data-driven approach can yield high levels of performance and nicely complements traditional question answering techniques driven by information extraction.

In addition to enormous amounts of unstructured text, the Web also contains pockets of structured and semistructured knowledge that can serve as a valuable resource for question answering. By organizing these resources and annotating them with natural language, we can successfully incorporate Web knowledge into question answering systems.

This tutorial surveys recent Web-based question answering technology, focusing on two separate paradigms: knowledge mining using statistical tools and knowledge annotation using database concepts. Both approaches can employ a wide spectrum of techniques ranging in linguistic sophistication from simple "bag-of-words" treatments to full syntactic parsing.

Introduction

- Why question answering?
  - Question answering provides intuitive information access
  - Computers should respond to human information needs with “just the right information”
- What role does the World Wide Web play in question answering?
  - The Web is an enormous store of human knowledge
  - This knowledge is a valuable resource for question answering

How can we effectively utilize the World Wide Web to answer natural language questions?

Different Types of Questions

What does Cog look like?

Who directed Gone with the Wind?

Gone with the Wind (1939) was directed by George Cukor, Victor Fleming, and Sam Wood.

How many cars left the garage yesterday between noon and 1pm?

What were the causes of the French Revolution?

"Factoid" Question Answering

- Modern systems are limited to answering fact-based questions
  - Answers are typically named-entities

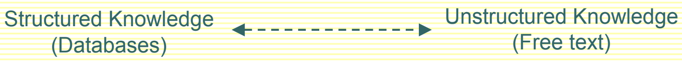
    Who discovered Oxygen?
    When did Hawaii become a state?
    Where is Ayer's Rock located?
    What team won the World Series in 1992?

- Future systems will move towards "harder questions", e.g.,
  - *Why* and *how* questions
  - Questions that require simple inferences

This tutorial focuses on using the Web to answer factoid questions…

Two Axes of Exploration

- Nature of the information
  - What type of information is the system utilizing to answer natural language questions?

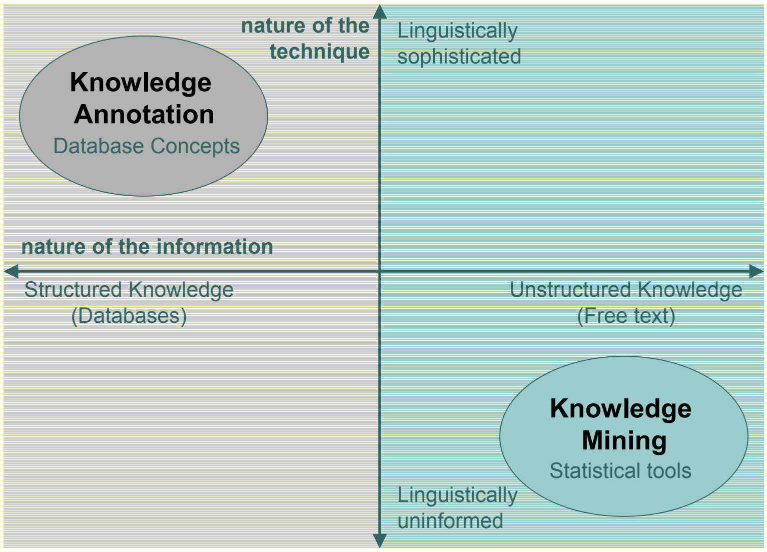


Structured Knowledge (Databases) ←- - - - - - - - - -→ Unstructured Knowledge (Free text)

- Nature of the technique
  - How linguistically sophisticated are the techniques employed to answer natural language questions?

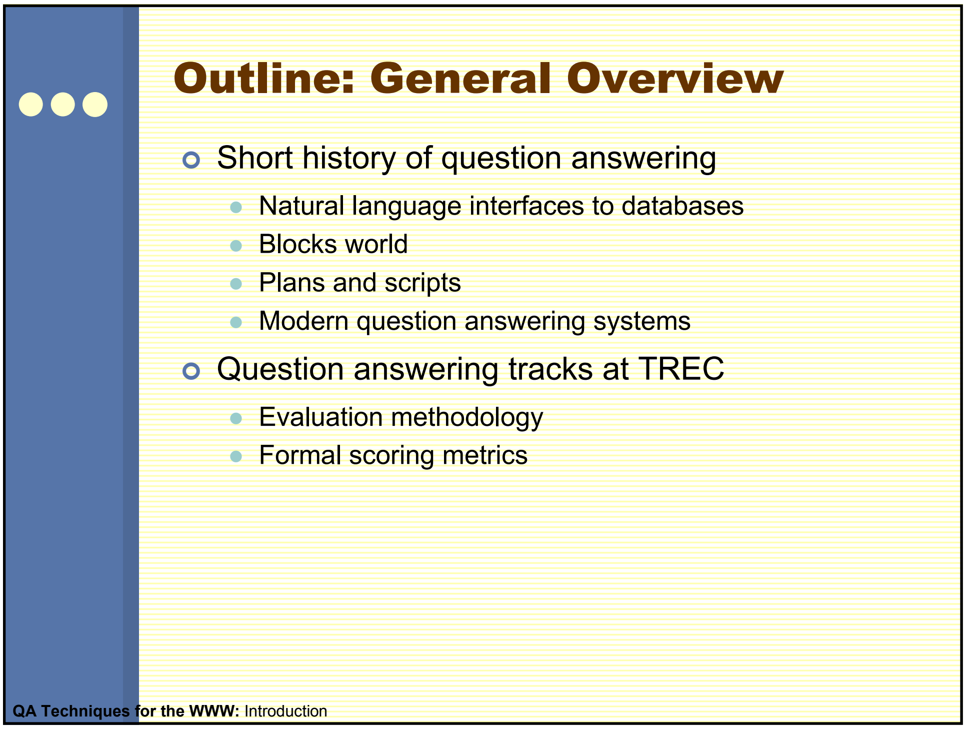
Linguistically Sophisticated (e.g., syntactic parsing) ←- - - - - - - - - -→ Linguistically Uninformed (e.g., *n*-gram generation)

Two Techniques for Web QA


nature of the technique
Linguistically sophisticated
Knowledge Annotation
Database Concepts
nature of the information
Structured Knowledge (Databases)
Unstructured Knowledge (Free text)
Knowledge Mining
Statistical tools
Linguistically uninformed

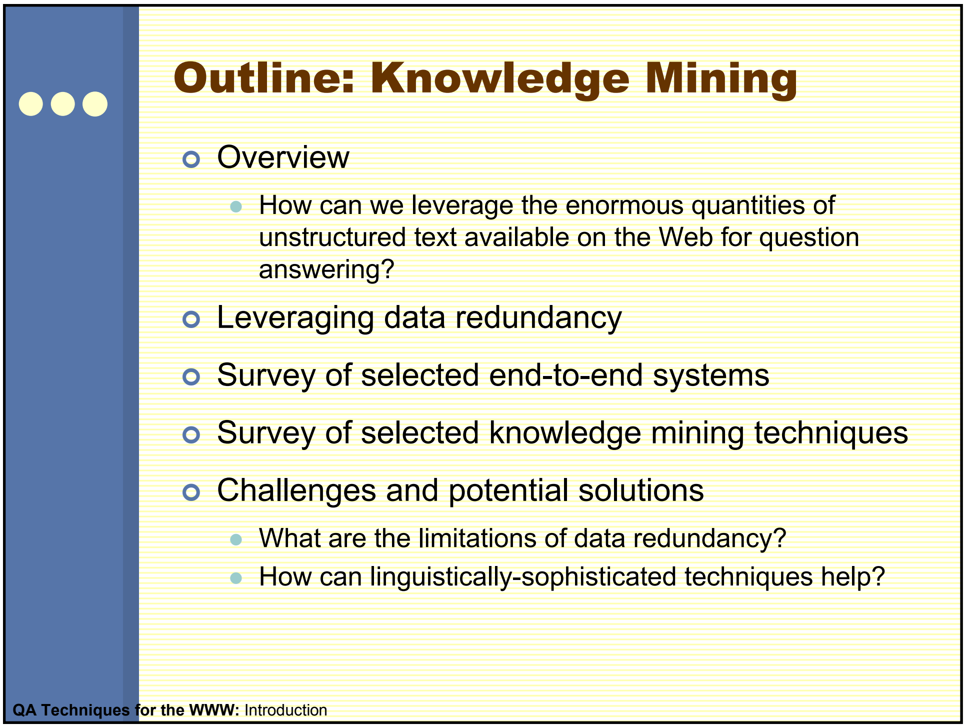
Outline: Top-Level

- **General Overview**: Origins of Web-based Question Answering
- **Knowledge Mining**: techniques that effectively employ unstructured text on the Web for question answering
- **Knowledge Annotation**: techniques that effectively employ structured and semistructured sources on the Web for question answering



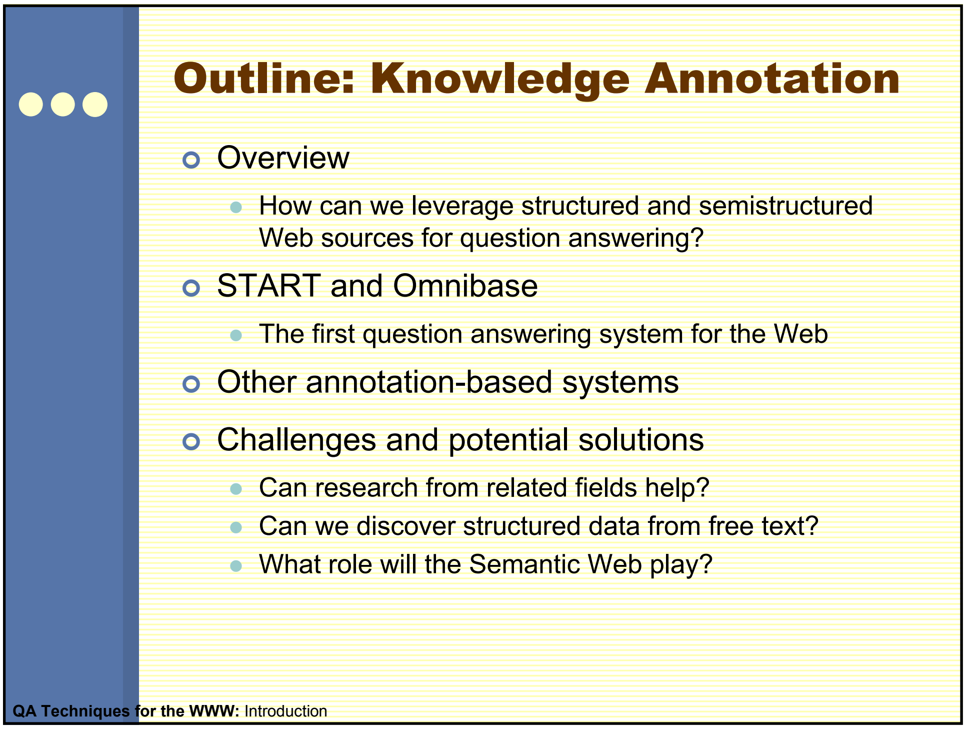
Outline: General Overview

- Short history of question answering
  - Natural language interfaces to databases
  - Blocks world
  - Plans and scripts
  - Modern question answering systems
- Question answering tracks at TREC
  - Evaluation methodology
  - Formal scoring metrics



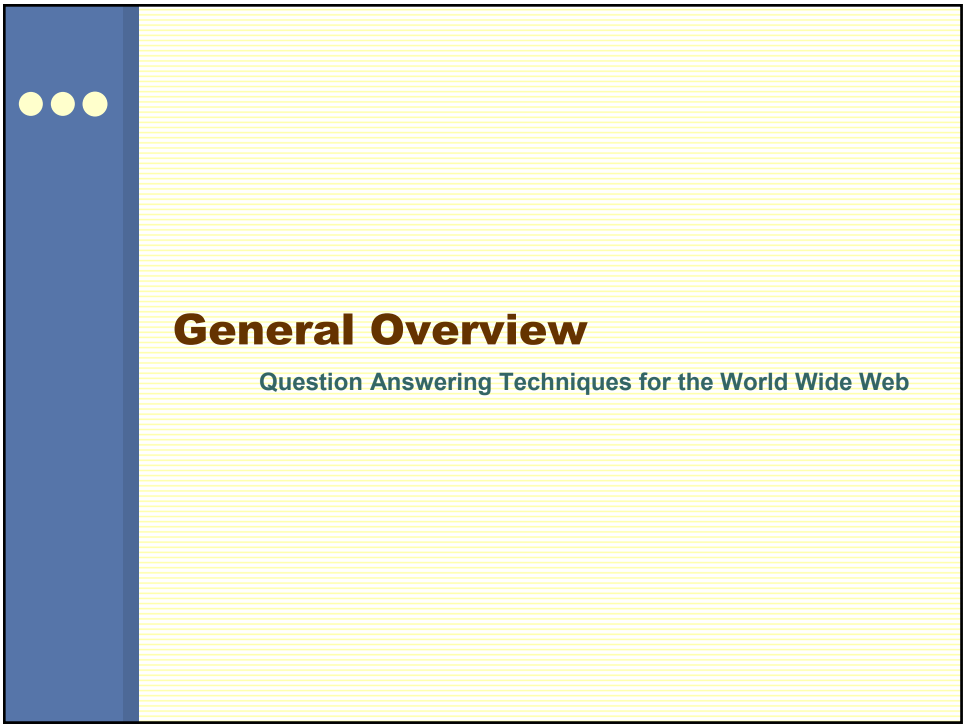
Outline: Knowledge Mining

- Overview
  - How can we leverage the enormous quantities of unstructured text available on the Web for question answering?
- Leveraging data redundancy
- Survey of selected end-to-end systems
- Survey of selected knowledge mining techniques
- Challenges and potential solutions
  - What are the limitations of data redundancy?
  - How can linguistically-sophisticated techniques help?



Outline: Knowledge Annotation

- Overview
  - How can we leverage structured and semistructured Web sources for question answering?
- START and Omnibase
  - The first question answering system for the Web
- Other annotation-based systems
- Challenges and potential solutions
  - Can research from related fields help?
  - Can we discover structured data from free text?
  - What role will the Semantic Web play?



General Overview

Question Answering Techniques for the World Wide Web

A Short History of QA

- Natural language interfaces to databases
- Blocks world
- Plans and scripts
- Emergence of the Web
- IR+IE-based QA and large-scale evaluation
- Re-discovery of the Web

NL Interfaces to Databases

- Natural language interfaces to relational databases
  - BASEBALL – baseball statistics [Green *et al.* 1961]

    Who did the Red Sox lose to on July 5?
    On how many days in July did eight teams play?

  - LUNAR – analysis of lunar rocks [Woods *et al.* 1972]

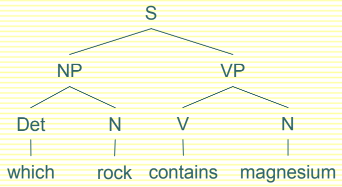
    What is the average concentration of aluminum in high alkali rocks?
    How many Brescias contain Olivine?

  - LIFER – personnel statistics [Hendrix 1977ab]

    What is the average salary of math department secretaries?
    How many professors are there in the compsci department?

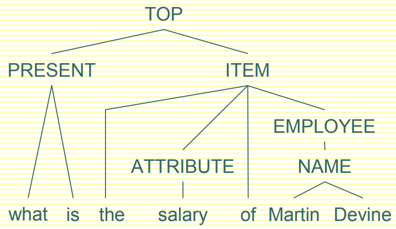
Typical Approaches

Direct Translation: determine mapping rules between syntactic structures and database queries (e.g., LUNAR)



(for_every X
(is_rock X)
(contains X magnesium)
(printout X))

Semantic Grammar: parse at the semantic level directly into database queries (e.g., LIFER)



Properties of Early NL Systems

- Often brittle and not scalable
  - Natural language understanding process was a mix of syntactic and semantic processing
  - Domain knowledge was often embedded implicitly in the parser
- Narrow and restricted domain
  - Users were often presumed to have some knowledge of underlying data tables
- Systems performed syntactic and semantic analysis of questions
  - Discourse modeling (e.g., anaphora, ellipsis) is easier in a narrow domain

Blocks World

- Interaction with a robotic arm in a world filled with colored blocks [Winograd 1972]
  - Not only answered questions, but also followed commands

    What is on top of the red brick?
    Is the blue cylinder larger than the one you are holding?
    Pick up the yellow brick underneath the green brick.
- The “blocks world” domain was a fertile ground for other research
  - Near-miss learning [Winston 1975]
  - Understanding line drawings [Waltz 1975]
  - Acquisition of problem solving strategies [Sussman 1973]

Plans and Scripts

- QUALM [Lehnert 1977,1981]
  - Application of scripts and plans for story comprehension
  - Very restrictive domain, e.g., restaurant scripts
  - Implementation status uncertain – difficult to separate discourse theory from working system
- UNIX Consultant [Wilensky 1982; Wilensky *et al.* 1989]
  - Allowed users to interact with UNIX, e.g., ask “How do I delete a file?”
  - User questions were translated into goals and matched with plans for achieving that goal: paradigm not suitable for general purpose question answering
  - Effectiveness and scalability of approach is unknown due to lack of rigorous evaluation

Emergence of the Web

- Before the Web…
  - Question answering systems had limited audience
  - All knowledge had to be hand-coded and specially prepared
- With the Web…
  - Millions can access question answering services
  - Question answering systems could take advantage of already-existing knowledge: “virtual collaboration”

START

MIT: [Katz 1988,1997; Katz *et al.* 2002a]

- The first question answering system for the World Wide Web
  - On-line and continuously operating since 1993
  - Has answered millions of questions from hundreds of thousands of users all over the world
  - Engages in “virtual collaboration” by utilizing knowledge freely available on the Web
- Introduced the knowledge annotation approach to question answering

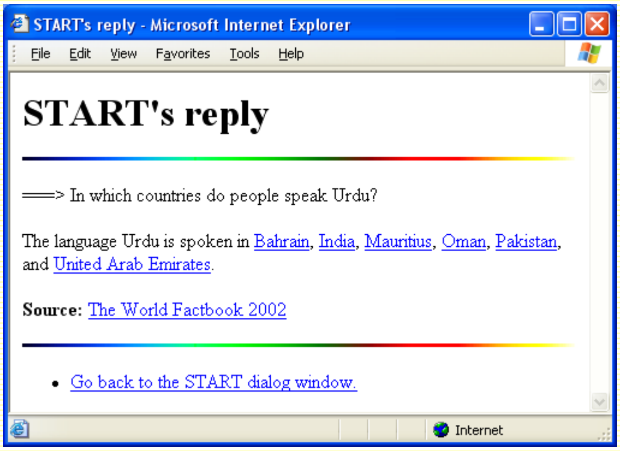
http://www.ai.mit.edu/projects/infolab

Additional START Applications

START is easily adaptable to different domains:

- Analogy/explanation-based learning [Winston *et al.* 1983]
- Answering questions from the GRE [Katz 1988]
- Answering questions in the JPL press room regarding the Voyager flyby of Neptune (1989) [Katz 1990]
- START Bosnia Server dedicated to the U.S. mission in Bosnia (1996)
- START Mars Server to inform the public about NASA's planetary missions (2001)
- START Museum Server for an ongoing exhibit at the MIT Museum (2001)

START in Action



START's reply

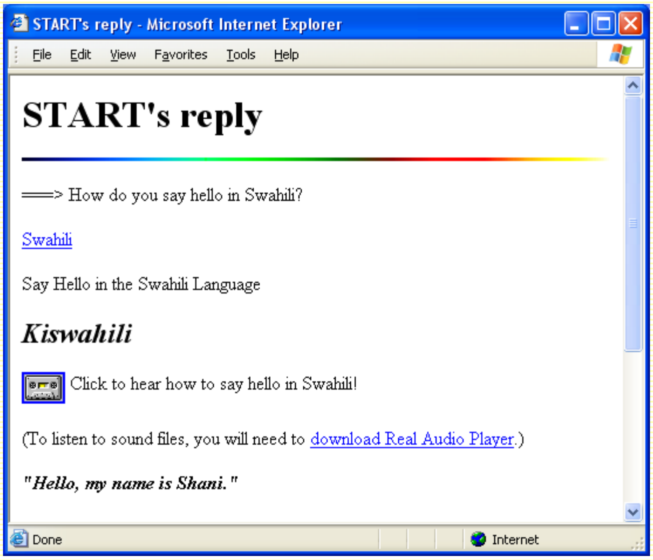
==> In which countries do people speak Urdu?

The language Urdu is spoken in Bahrain, India, Mauritius, Oman, Pakistan, and United Arab Emirates.

Source: The World Factbook 2002

- Go back to the START dialog window.

START in Action


START's reply - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Done
Internet

START's reply

===> How do you say hello in Swahili?

Swahili

Say Hello in the Swahili Language

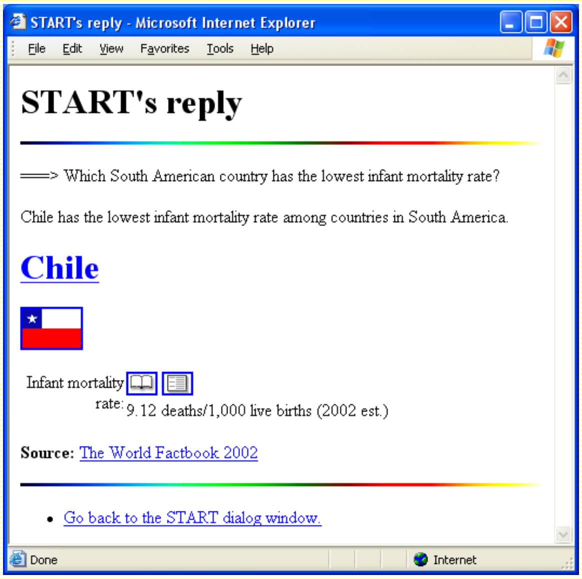
Kiswahili

Click to hear how to say hello in Swahili!

(To listen to sound files, you will need to download Real Audio Player.)

"Hello, my name is Shani."

START in Action


START's reply - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Done
Internet

START's reply

===> Which South American country has the lowest infant mortality rate?

Chile has the lowest infant mortality rate among countries in South America.

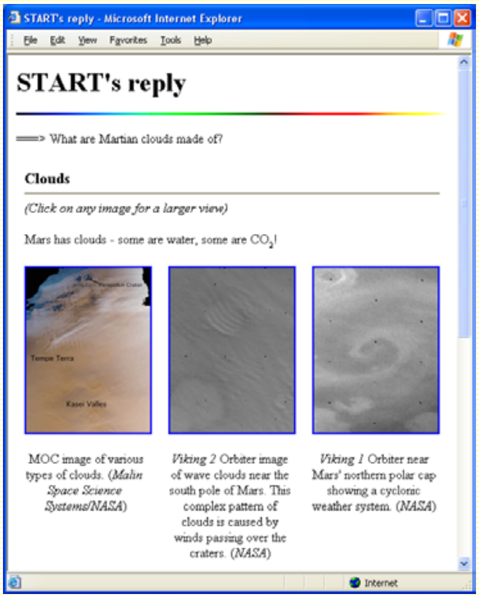
Chile

Infant mortality rate: 9.12 deaths/1,000 live births (2002 est.)

Source: The World Factbook 2002

- Go back to the START dialog window.

START in Action


START's reply - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Tempe Terra
Kasei Valles
Internet

START's reply

==> What are Martian clouds made of?

Clouds

(Click on any image for a larger view)

Mars has clouds - some are water, some are CO_2!

MOC image of various types of clouds. (*Malin Space Science Systems/NASA*)

Viking 2 Orbiter image of wave clouds near the south pole of Mars. This complex pattern of clouds is caused by winds passing over the craters. (*NASA*)

Viking 1 Orbiter near Mars' northern polar cap showing a cyclonic weather system. (*NASA*)

Related Strands: IR and IE

- Information retrieval has a long history
  - Origins can be traced back to Vannevar Bush (1945)
  - Active field since mid-1950s
  - Primary focus on document retrieval
  - Finer-grained IR: emergence of passage retrieval techniques in early 1990s
- Information extraction seeks to “distill” information from large numbers of documents
  - Concerned with filling in pre-specified templates with participating entities
  - Started in the late 1980s with the Message Understanding Conferences (MUCs)

IR+IE-based QA

- Recent question answering systems are based on information retrieval and information extraction
  - Answers are extracted from closed corpora, e.g., newspaper and encyclopedia articles
  - Techniques range in sophistication from simple keyword matching to some parsing
- Formal, large-scale evaluations began with the TREC QA tracks
  - Facilitated rapid dissemination of results and formation of a community
  - Dramatically increased speed at which new techniques have been adopted

Re-discovery of the Web

- IR+IE-based systems focus on answering questions from a closed corpus
  - Artifact of the TREC setup
- Recently, researchers have discovered a wealth of resource on the Web
  - Vast amounts of unstructured free text
  - Pockets of structured and semistructured sources
- This is where we are today…

How can we effectively utilize the Web to answer natural language questions?

The Short Answer

- **Knowledge Mining:** techniques that effectively employ unstructured text on the Web for question answering
- **Knowledge Annotation:** techniques that effectively employ structured and semistructured sources on the Web for question answering

General Overview:

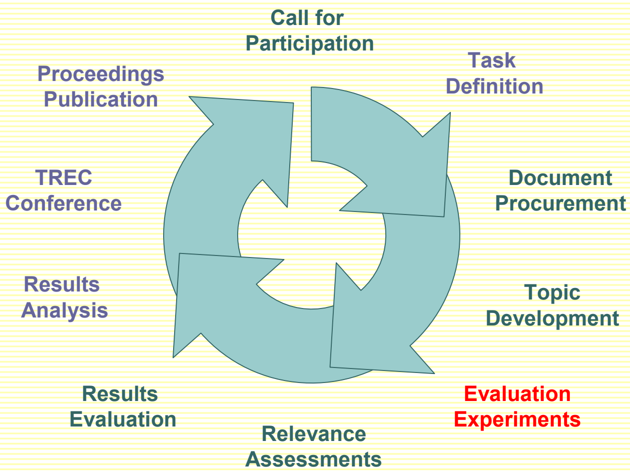
TREC Question Answering Tracks

Question Answering Techniques for the World Wide Web

TREC QA Tracks

- Question answering track at the Text Retrieval Conference (TREC)
  - Large-scale evaluation of question answering
  - Sponsored by NIST (with later support from ARDA)
  - Uses formal evaluation methodologies from information retrieval
- Formal evaluation is a part of a larger “community process”

The TREC Cycle



TREC QA Tracks

- TREC-8 QA Track [Voorhees and Tice 1999,2000b]
  - 200 questions: backformulations of the corpus
  - Systems could return up to five answers

    answer = [answer string, docid]
  - Two test conditions: 50-byte or 250-byte answer strings
  - MRR scoring metric
- TREC-9 QA Track [Voorhees and Tice 2000a]
  - 693 questions: from search engine logs
  - Systems could return up to five answers

    answer = [answer string, docid]
  - Two test conditions: 50-byte or 250-byte answer strings
  - MRR scoring metric

TREC QA Tracks

- TREC 2001 QA Track [Voorhees 2001,2002a]
  - 500 questions: from search engine logs
  - Systems could return up to five answers

    answer = [answer string, docid]
  - 50-byte answers only
  - Approximately a quarter of the questions were definition questions (unintentional)
- TREC 2002 QA Track [Voorhees 2002b]
  - 500 questions: from search engine logs
  - Each system could only return one answer per question

    answer = [exact answer string, docid]
  - All answers were sorted by decreasing confidence
  - Introduction of “exact answers” and CWS metric

Evaluation Metrics

- Mean Reciprocal Rank (MRR) (through TREC 2001)
  - Reciprocal rank = inverse of rank at which first correct answer was found: {1, 0.5, 0.33, 0.25, 0.2, 0}
  - MRR = average over all questions
  - Judgments: correct, unsupported, incorrect

Correct: answer string answers the question in a "responsive" fashion and is supported by the document

Unsupported: answer string is correct but the document does not support the answer

Incorrect: answer string does not answer the question

  - Strict score: unsupported counts as incorrect
  - Lenient score: unsupported counts as correct

Evaluation Metrics

- Confidence-Weighted Score (CWS) (TREC 2002)
  - Evaluates how well "systems know what they know"

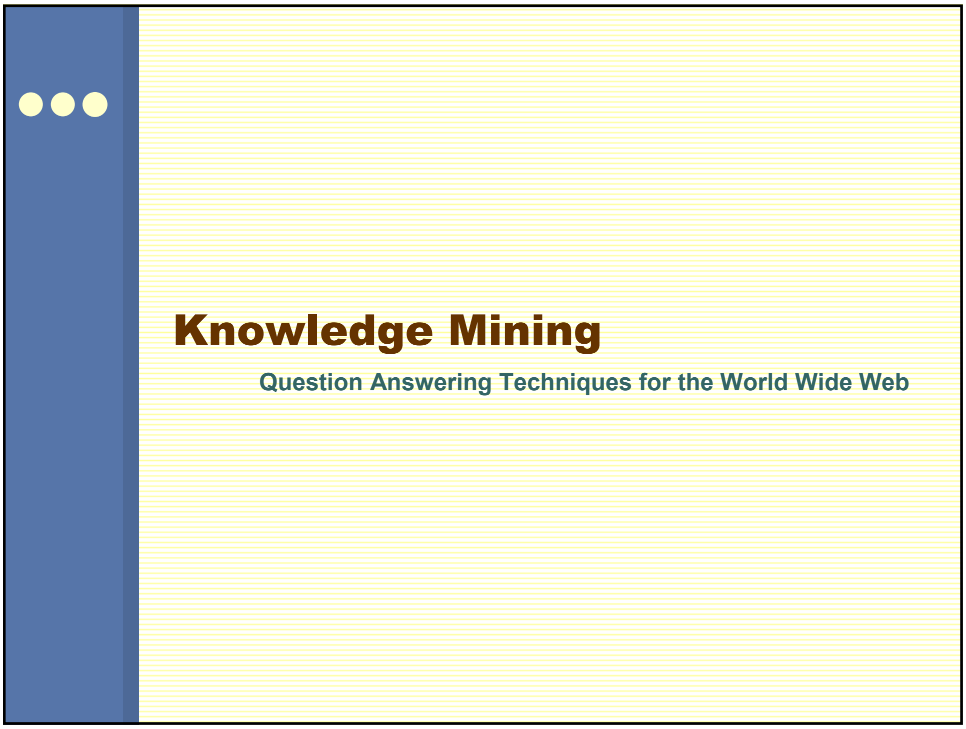
$$\frac{\sum_{i=1}^{Q} i_c / i}{Q}$$

i_c = number of correct answers in first *i* questions
Q = total number of questions

  - Judgments: correct, unsupported, inexact, wrong

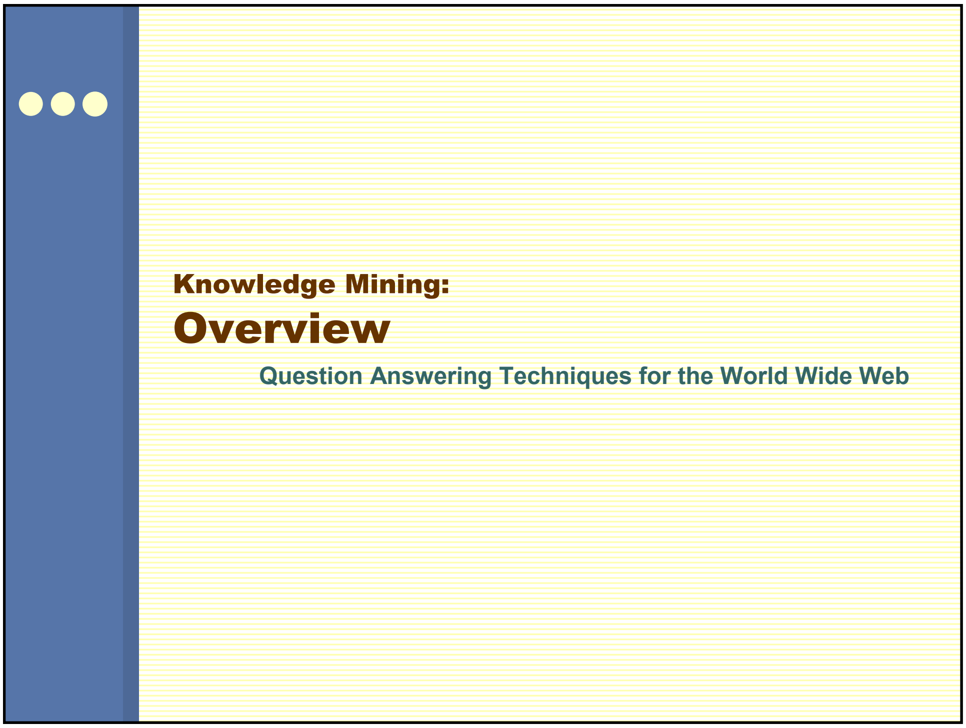
Exact answers:
Mississippi
the Mississippi
the Mississippi River
Mississippi River
mississippi

Inexact answers:
At 2,348 miles the Mississippi River is the longest river in the US.
2,348; Mississippi
Missipp



Knowledge Mining

Question Answering Techniques for the World Wide Web



Knowledge Mining:
Overview

Question Answering Techniques for the World Wide Web

Knowledge Mining

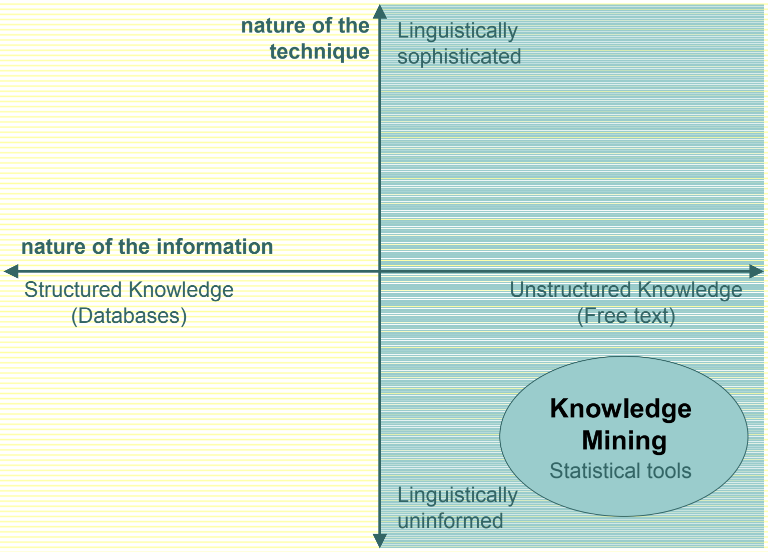
- **Definition:** techniques that effectively employ unstructured text on the Web for question answering
- **Key Ideas:**
  - Leverage data redundancy
  - Use simple statistical techniques to bridge question and answer gap
  - Use linguistically-sophisticated techniques to improve answer quality

Key Questions

- How is the Web different from a closed corpus?
- How can we quantify and leverage data redundancy?
- How can data-driven approaches help solve some NLP challenges?
- How do we make the most out of existing search engines?

How can we effectively employ unstructured text on the Web for question answering?

Knowledge Mining



nature of the technique

Linguistically sophisticated

nature of the information

Structured Knowledge (Databases)

Unstructured Knowledge (Free text)

Knowledge Mining

Statistical tools

Linguistically uninformed

“Knowledge” and “Data” Mining

How is knowledge mining related to data mining?

Knowledge Mining

- Answers specific natural language questions
- Benefits from well-specified input and output
- Primarily utilizes textual sources

Data Mining

- Discovers interesting patterns and trends
- Often suffers from vague goals
- Utilizes a variety of data from text to numerical databases

Similarities:

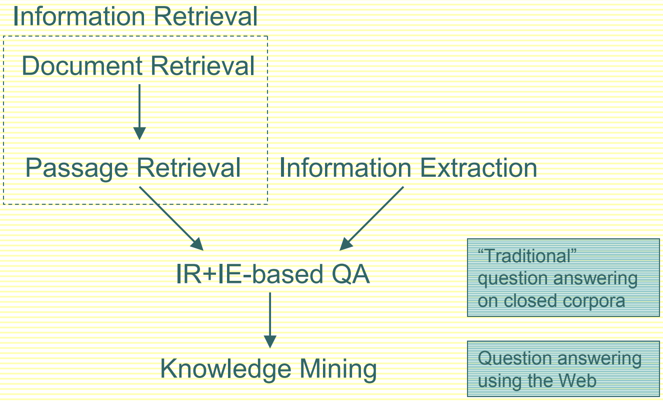
- Both are driven by enormous quantities of data
- Both leverage statistical and data-driven techniques

Present and Future

- Current state of knowledge mining:
  - Most research activity concentrated in the last two years
  - Good performance using statistical techniques
- Future of knowledge mining:
  - Build on statistical techniques
  - Overcome brittleness of current natural language techniques
  - Address remaining challenges with linguistic knowledge
  - Selectively employ linguistic analysis: use it only in beneficial situations

Origins of Knowledge Mining

The origins of knowledge mining lie in information retrieval and information extraction



Information Retrieval

Document Retrieval

Passage Retrieval

Information Extraction

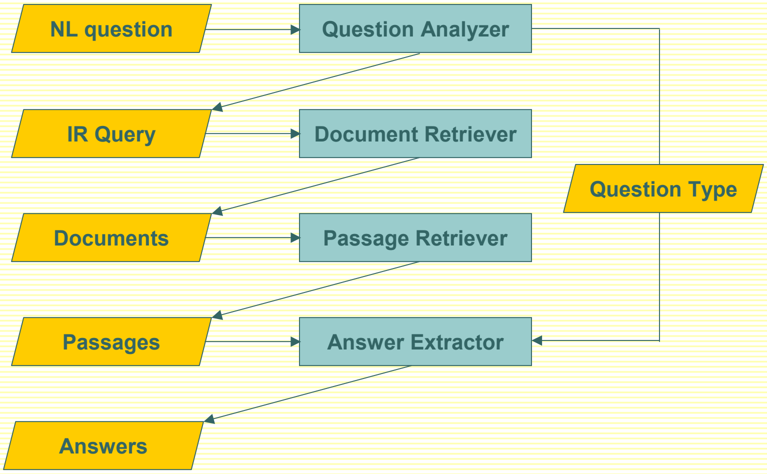
IR+IE-based QA

Knowledge Mining

"Traditional" question answering on closed corpora

Question answering using the Web

“Traditional” IR+IE-based QA



“Traditional” IR+IE-based QA

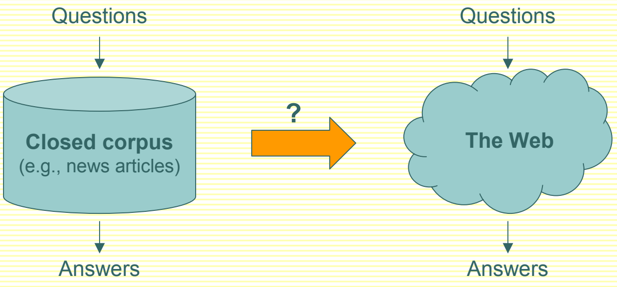
- Question Analyzer Input = natural language question
  - Determines expected answer type
  - Generates query for IR engine
- Document Retriever Input = IR query
  - Narrows corpus down to a smaller set of potentially relevant documents
- Passage Retrieval Input = set of documents
  - Narrows documents down to a set of passages for additional processing
- Answer Extractor Input = set of passages + question type
  - Extracts the final answer to the question
  - Typically matches entities from passages against the expected answer type
  - May employ more linguistically-sophisticated processing

References: IR+IE-based QA

- General Survey [Hirschman and Gaizauskas 2001]
- Sample Systems
  - Cymfony at TREC-8 [Srihari and Li 1999]
    - Three-level information extraction architecture
  - IBM at TREC-9 (and later versions) [Prager *et al.* 1999]
    - Predictive annotations: perform named-entity detection at time of index creation
  - FALCON (and later versions) [Harabagiu *et al.* 2000a]
    - Employs question/answer logic unification and feedback loops
- Tutorials [Harabagiu and Moldovan 2001, 2002]

Just Another Corpus?

- Is the Web just another corpus?
- Can we simply apply traditional IR+IE-based question answering techniques on the Web?



Questions

Closed corpus
(e.g., news articles)

Answers

?

Questions

The Web

Answers

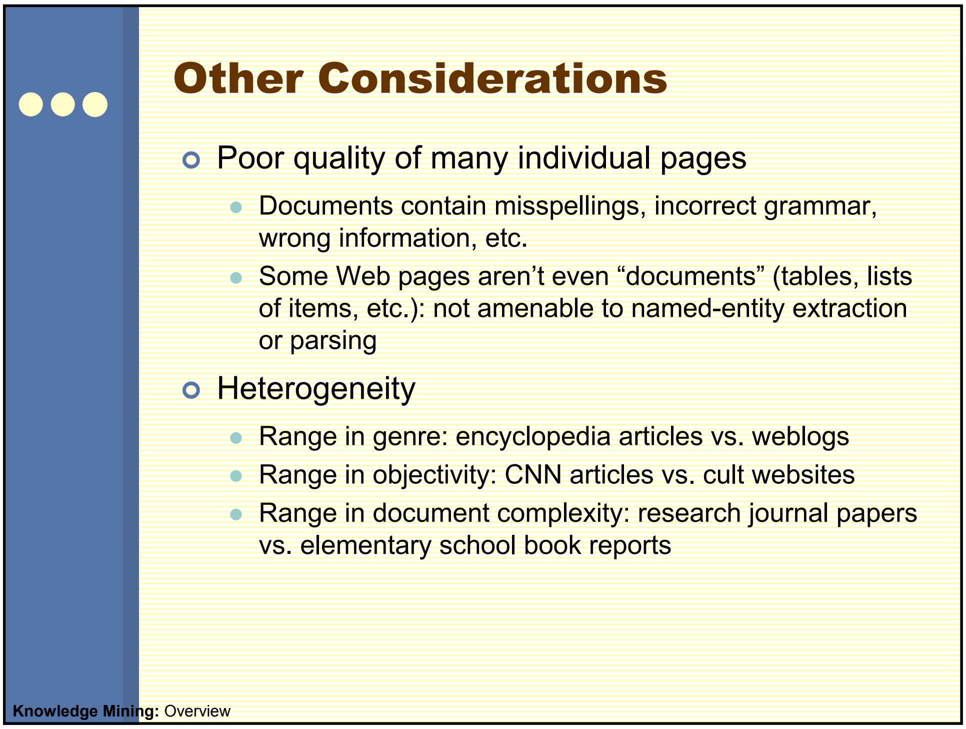
Not Just Another Corpus…

- The Web is qualitatively different from a closed corpus
- Many IR+IE-based question answering techniques will still be effective
- But we need a different set of techniques to capitalize on the Web as a document collection

Size and Data Redundancy

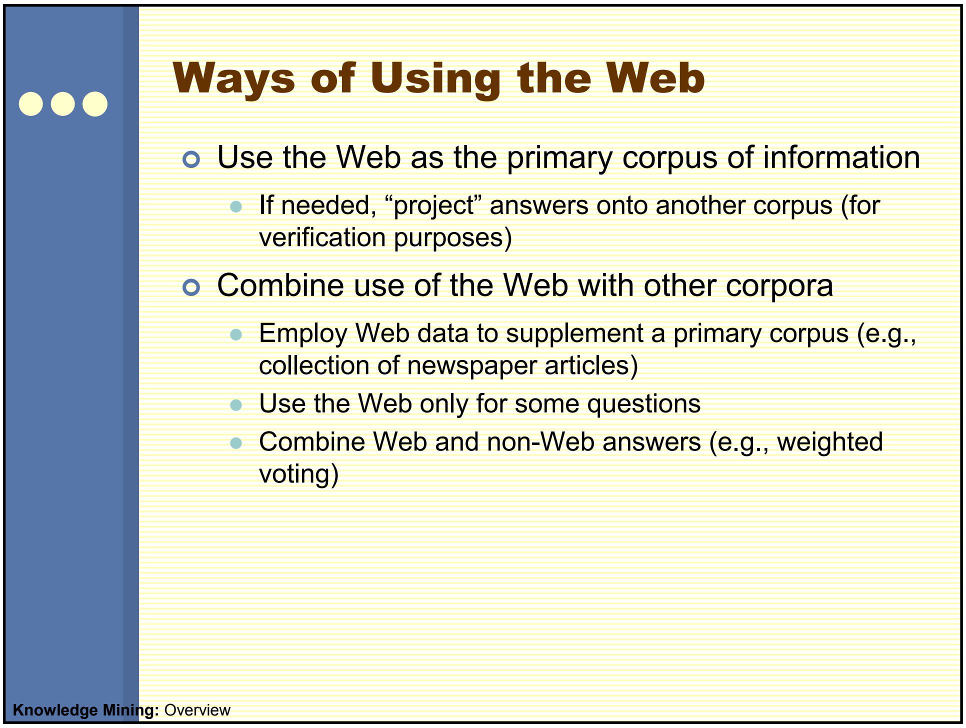
- How big?
  - Tens of terabytes? No agreed upon methodology to even measure it
  - Google indexes over 3 billion Web pages (early 2003)
- Size introduces engineering issues
  - Use existing search engines? Limited control over search results
  - Crawl the Web? Very resource intensive
- **Size gives rise to data redundancy**
  - Knowledge stated multiple times…

    in multiple documents
    in multiple formulations



Other Considerations

- Poor quality of many individual pages
  - Documents contain misspellings, incorrect grammar, wrong information, etc.
  - Some Web pages aren’t even “documents” (tables, lists of items, etc.): not amenable to named-entity extraction or parsing
- Heterogeneity
  - Range in genre: encyclopedia articles vs. weblogs
  - Range in objectivity: CNN articles vs. cult websites
  - Range in document complexity: research journal papers vs. elementary school book reports



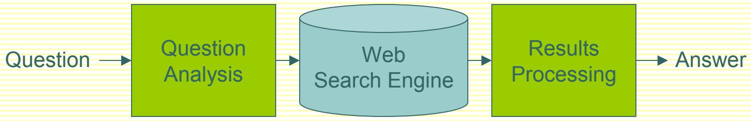
Ways of Using the Web

- Use the Web as the primary corpus of information
  - If needed, “project” answers onto another corpus (for verification purposes)
- Combine use of the Web with other corpora
  - Employ Web data to supplement a primary corpus (e.g., collection of newspaper articles)
  - Use the Web only for some questions
  - Combine Web and non-Web answers (e.g., weighted voting)

Capitalizing on Search Engines

Data redundancy would be useless unless we could easily access all that data…

- Leverage existing information retrieval infrastructure [Brin and Page 1998]
  - The engineering task of indexing and retrieving terabyte-sized document collections has been solved
- Existing search engines are "good enough"
  - Build systems on top of commercial search engines, e.g., Google, FAST, AltaVista, Teoma, etc.



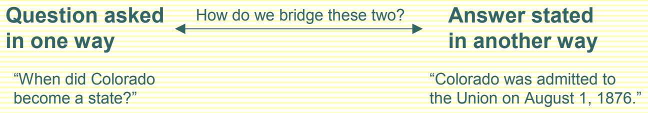
Knowledge Mining:

Leveraging Data Redundancy

Question Answering Techniques for the World Wide Web

Leveraging Data Redundancy

- Take advantage of different reformulations
  - The expressiveness of natural language allows us to say the same thing in multiple ways
  - This poses a problem for question answering



Question asked in one way

How do we bridge these two?

Answer stated in another way

"When did Colorado become a state?"

"Colorado was admitted to the Union on August 1, 1876."

  - With data redundancy, it is likely that answers will be stated in the same way the question was asked
- Cope with poor document quality
  - When many documents are analyzed, wrong answers become "noise"

Leveraging Data Redundancy

Data Redundancy = Surrogate for sophisticated NLP
Obvious reformulations of questions can be easily found

Who killed Abraham Lincoln?

(1) John Wilkes Booth killed Abraham Lincoln.
(2) John Wilkes Booth altered history with a bullet. He will forever be known as the man who ended Abraham Lincoln's life.

When did Wilt Chamberlain score 100 points?

(1) Wilt Chamberlain scored 100 points on March 2, 1962 against the New York Knicks.
(2) On December 8, 1961, Wilt Chamberlain scored 78 points in a triple overtime game. It was a new NBA record, but Warriors coach Frank McGuire didn't expect it to last long, saying, "He'll get 100 points someday." McGuire's prediction came true just a few months later in a game against the New York Knicks on March 2.

Leveraging Data Redundancy

Data Redundancy can overcome poor document quality
Lots of wrong answers, but even more correct answers

What's the rainiest place in the world?

(1) Blah blah **Seattle** blah blah **Hawaii** blah blah blah blah blah
(2) Blah **Sahara Desert** blah blah blah blah blah blah blah **Amazon**
(3) Blah blah blah blah blah blah blah **Mount Waiale'ale in Hawaii** blah
(4) Blah blah blah **Hawaii** blah blah blah blah **Amazon** blah blah
(5) Blah **Mount Waiale'ale** blah blah blah blah blah blah blah blah blah

What is the furthest planet in the Solar System?

(1) Blah **Pluto** blah blah blah blah **Planet X** blah blah
(2) Blah blah blah blah **Pluto** blah blah blah blah blah blah blah blah
(3) Blah blah blah **Planet X** blah blah blah blah blah blah blah **Pluto**
(4) Blah **Pluto** blah blah blah blah blah blah blah blah **Pluto** blah blah

General Principles

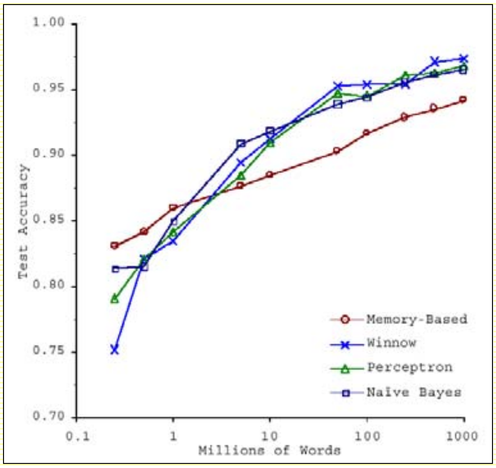
- Match answers using surface patterns
  - Apply regular expressions over textual snippets to extract answers
  - Bypass linguistically sophisticated techniques, e.g., parsing
- Rely on statistics and data redundancy
  - Expect many occurrences of the answer mixed in with many occurrences of wrong, misleading, or lower quality answers
  - Develop techniques for filtering, sorting large numbers of candidates

Can we "quantify" data redundancy?

Leveraging Massive Data Sets

[Banko and Brill 2001]

Grammar Correction: {two, to, too} {principle, principal}



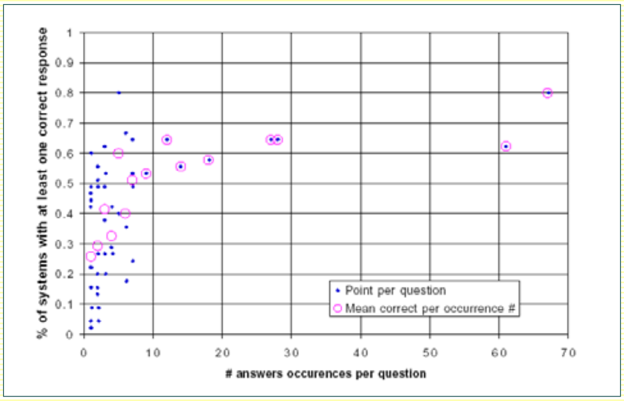
Observations: Banko and Brill

- For some applications, learning technique is less important than amount of training data
  - In the limit (i.e., infinite data), performance of different algorithms converges
  - It doesn’t matter if the data is (somewhat) noisy
  - Why compare performance of learning algorithms on (relatively) small corpora?
- In many applications, data is free!
- Throwing more data at a problem is sometimes the easiest solution (hence, we should try it first)

Effects of Data Redundancy

[Breck *et al.* 2001; Light *et al.* 2001]

Are questions with more answer occurrences "easier"?
Examined the effect of answer occurrences on question answering performance (on TREC-8 results)

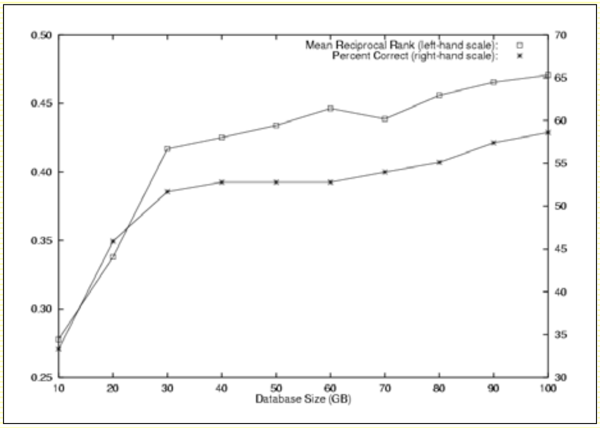


~27% of systems produced a correct answer for questions with 1 answer occurrence.
~50% of systems produced a correct answer for questions with 7 answer occurrences.

Effects of Data Redundancy

[Clarke *et al.* 2001a]

How does corpus size affect performance?
Selected 87 "people" questions from TREC-9; Tested effect of corpus size on passage retrieval algorithm (using 100GB TREC Web Corpus)



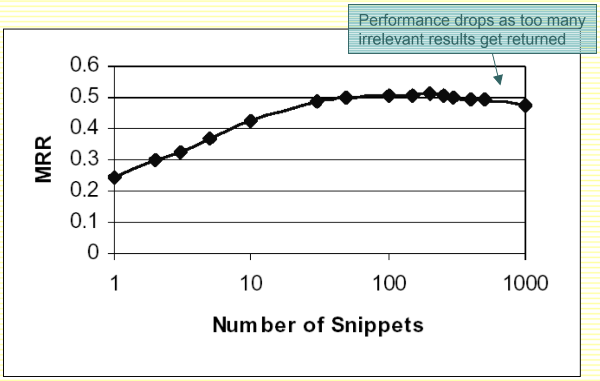
Conclusion: having more data improves performance

Effects of Data Redundancy

[Dumais *et al.* 2002]

How many search engine results should be used?
Plotted performance of a question answering system against the number of search engine snippets used

# Snippets	MRR
1	0.243
5	0.370
10	0.423
50	0.501
200	0.514



Performance drops as too many irrelevant results get returned

MRR as a function of number of snippets returned from the search engine. (TREC-9, q201-700)

Knowledge Mining:
System Survey

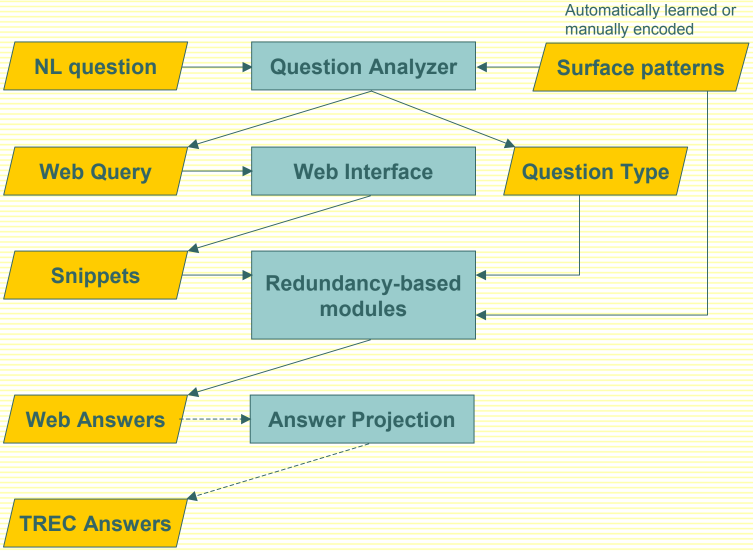
Question Answering Techniques for the World Wide Web

Knowledge Mining: Systems

- Ionaut (AT&T Research)
- MULDER (University of Washington)
- AskMSR (Microsoft Research)
- InsightSoft-M (Moscow, Russia)
- MultiText (University of Waterloo)
- Shapaqa (Tilburg University)
- Aranea (MIT)
- TextMap (USC/ISI)
- LAMP (National University of Singapore)
- NSIR (University of Michigan)
- PRIS (National University of Singapore)
- AnswerBus (University of Michigan)

Selected systems, apologies for any omissions

"Generic System"



Common Techniques

- Match answers using surface patterns
  - Apply regular expressions over textual snippets to extract answers

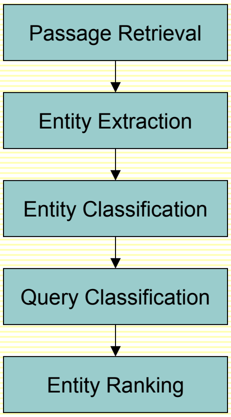
Surface patterns may also help in generating queries; they are either learned automatically or entered manually

- Leverage statistics and multiple answer occurrences
  - Generate n-grams from snippets
  - Vote, tile, filter, etc.
- Apply information extraction technology
  - Ensure that candidates match expected answer type

Ionaut

AT&T Research: [Abney *et al.* 2000]

Application of IR+IE-based question answering paradigm on documents gathered from a Web crawl


Passage Retrieval
Entity Extraction
Entity Classification
Query Classification
Entity Ranking

http://www.ionaut.com:8400/

Ionaut: Overview

- Passage Retrieval
  - SMART IR System [Salton 1971; Buckley and Lewit 1985]
  - Segment documents into three-sentence passages
- Entity Extraction
  - Cass partial parser [Abney 1996]
- Entity Classification
  - Proper names: person, location, organization
  - Dates
  - Quantities
  - Durations, linear measures

Ionaut: Overview

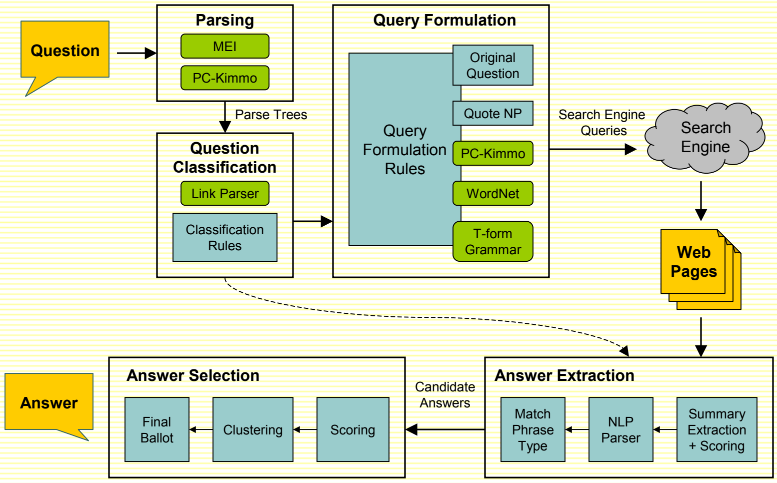
- Query Classification: 8 hand-crafted rules
  - Who, whom → Person
  - Where, whence, whither → Location
  - When → Date
  - And other simple rules
- Criteria for Entity Ranking:
  - Match between query classification and entity classification
  - Frequency of entity
  - Position of entity within retrieved passages

Ionaut: Evaluation

- End-to-end performance: TREC-8 (informal)
  - Exact answer: 46% answer in top 5, 0.356 MRR
  - 50-byte: 39% answer in top 5, 0.261 MRR
  - 250-byte: 68% answer in top 5, 0.545 MRR
- Error analysis
  - Good performance on person, location, date, and quantity (60%)
  - Poor performance on other types

MULDER

U. Washington: [Kwok *et al.* 2001]



Question

Parsing

MEI

PC-Kimmo

Parse Trees

Question Classification

Link Parser

Classification Rules

Query Formulation

Query Formulation Rules

Original Question

Quote NP

PC-Kimmo

WordNet

T-form Grammar

Search Engine Queries

Search Engine

Web Pages

Answer Extraction

Summary Extraction + Scoring

NLP Parser

Match Phrase Type

Candidate Answers

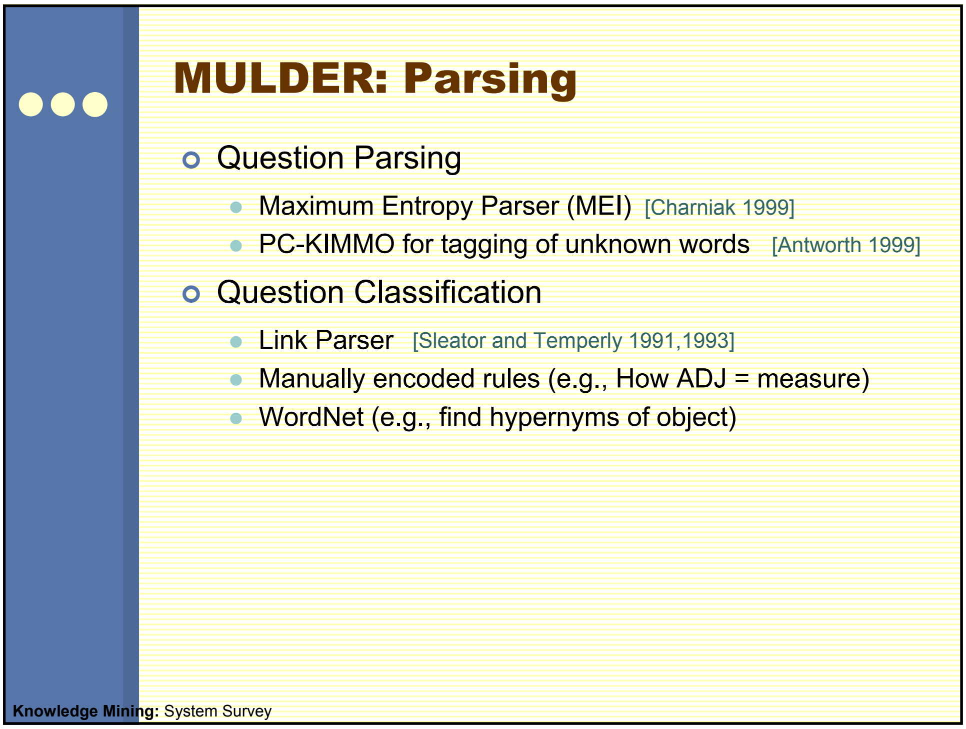
Answer Selection

Scoring

Clustering

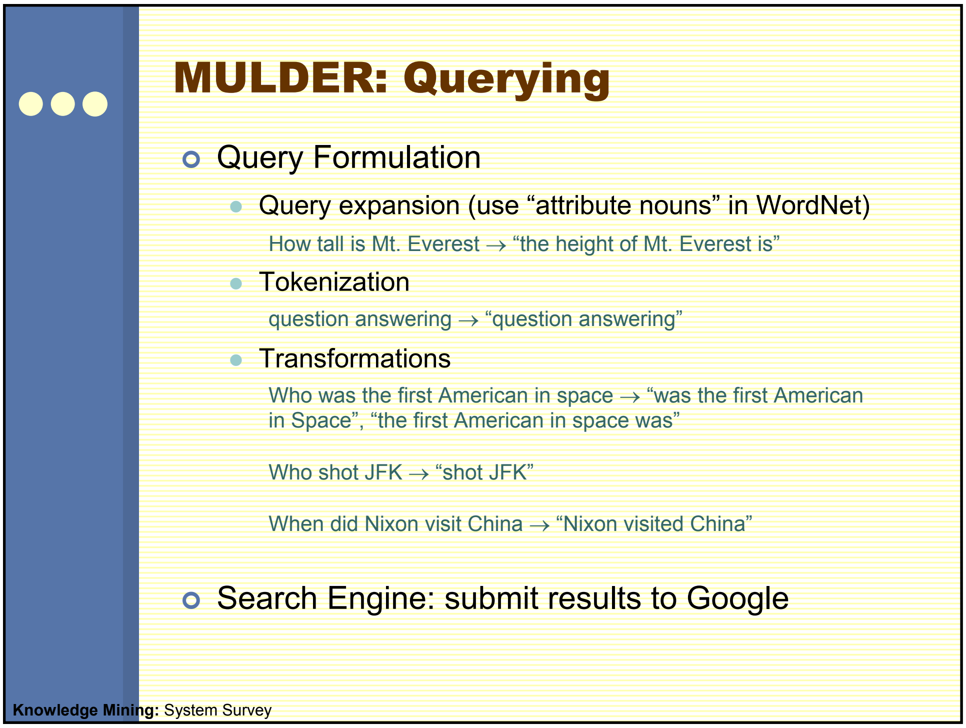
Final Ballot

Answer



MULDER: Parsing

- Question Parsing
  - Maximum Entropy Parser (MEI) [Charniak 1999]
  - PC-KIMMO for tagging of unknown words [Antworth 1999]
- Question Classification
  - Link Parser [Sleator and Temperly 1991,1993]
  - Manually encoded rules (e.g., How ADJ = measure)
  - WordNet (e.g., find hypernyms of object)



MULDER: Querying

- Query Formulation
  - Query expansion (use “attribute nouns” in WordNet)

    How tall is Mt. Everest → “the height of Mt. Everest is”
  - Tokenization

    question answering → “question answering”
  - Transformations

    Who was the first American in space → “was the first American in Space”, “the first American in space was”

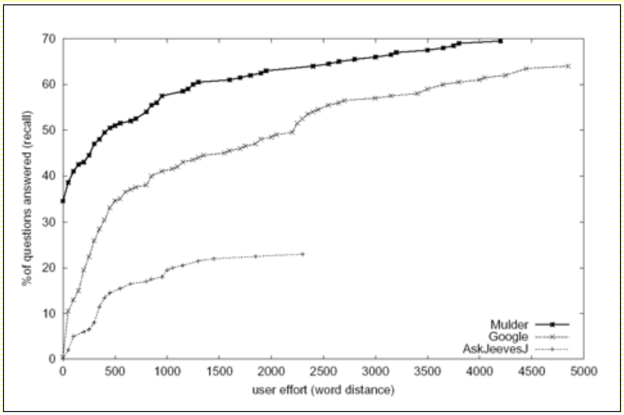
    Who shot JFK → “shot JFK”

    When did Nixon visit China → “Nixon visited China”
- Search Engine: submit results to Google

MULDER: Answer Extraction

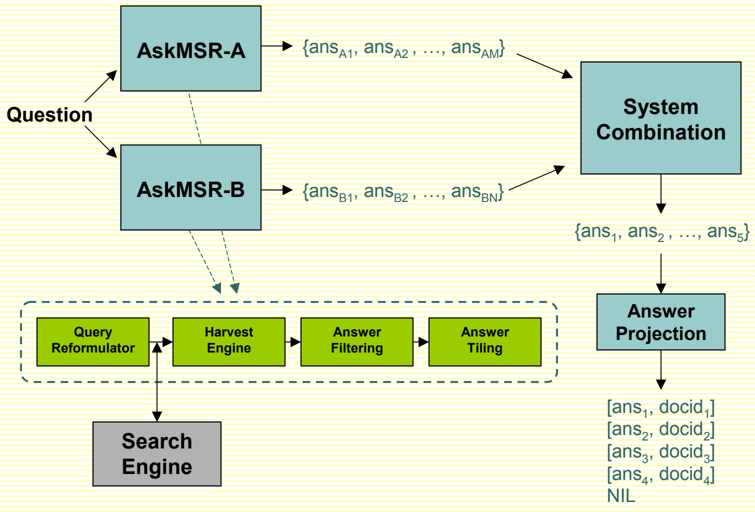
- Answer Extraction: extract summaries directly from Web pages
  - Locate regions with keywords
  - Score regions by keyword density and keyword *idf* values
  - Select top regions and parse them with MEI
  - Extract phrases of the expected answer type
- Answer Selection: score candidates based on
  - Simple frequency – voting
  - Closeness to keywords in the neighborhood

MULDER: Evaluation



- Evaluation on TREC-8 (200 questions)
  - Did not use MRR metric: results not directly comparable
  - “User effort”: how much text users must read in order to find the correct answer

AskMSR [Brill *et al.* 2001; Banko *et al.* 2002; Brill *et al.* 2002]



AskMSR: N-Gram Harvesting

Use text patterns derived from question to extract sequences of tokens that are likely to contain the answer

Question: Who is Bill Gates married to?

Look five tokens to the right

<“Bill Gates is married to”, right, 5>

... It is now the largest software company in the world. Today, Bill Gates is married to co-worker Melinda French. They live together in a house in the Redmond ...

... I also found out that Bill Gates is married to Melinda French Gates and they have a daughter named Jennifer Katharine Gates and a son named Rory John Gates. I ...

... of Microsoft, and they both developed Microsoft. * Presently Bill Gates is married to Melinda French Gates. They have two children: a daughter, Jennifer, and a ...

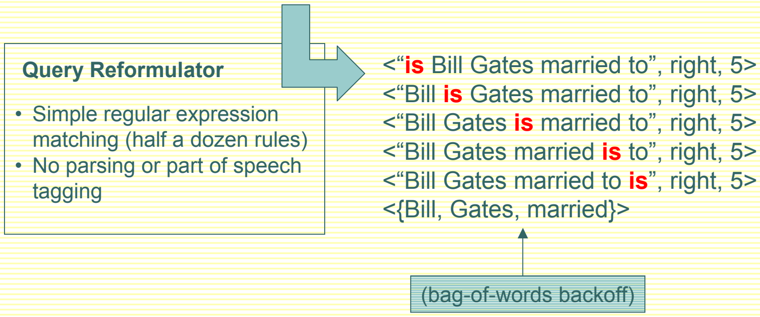
Generate N-Grams from Google summary snippets (bypassing original Web pages)

co-worker, co-worker Melinda, co-worker Melinda French, Melinda, Melinda French, Melinda French they, French, French they, French they live…

AskMSR: Query Reformulation

- Transform English questions into search engine queries
- Anticipate possible answer fragments

Question: Who is Bill Gates married to?



Query Reformulator

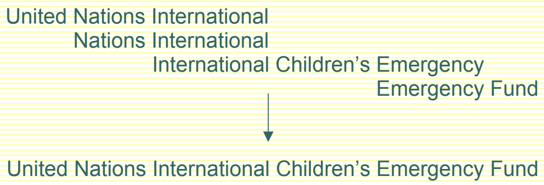
- Simple regular expression matching (half a dozen rules)
- No parsing or part of speech tagging

<“**is** Bill Gates married to”, right, 5>
<“Bill **is** Gates married to”, right, 5>
<“Bill Gates **is** married to”, right, 5>
<“Bill Gates married **is** to”, right, 5>
<“Bill Gates married to **is**”, right, 5>
<{Bill, Gates, married}>

(bag-of-words backoff)

AskMSR: Filter/Vote/Tile

- **Answer Filtering**: filter by question type
  - Simple regular expressions, e.g., for dates
- **Answer Voting**: score candidates by frequency of occurrence
- **Answer Tiling**: combine shorter candidates into longer candidates



United Nations International
Nations International
International Children’s Emergency
Emergency Fund

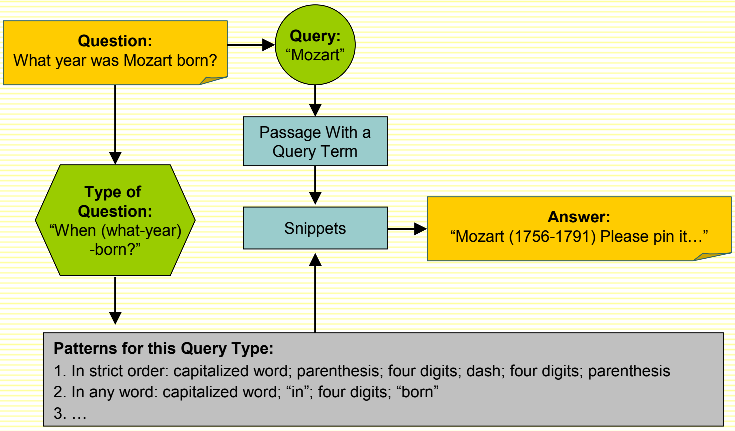
United Nations International Children’s Emergency Fund

AskMSR: Performance

- End-to-end performance: TREC-2001 (official)
  - MRR: 0.347 (strict), 0.434 (lenient)
- Lenient score is 25% higher than strict score
- Answer projection = weakest link
  - For 20% of correct answers, no adequate supporting document could be found
- Observations and questions
  - First question answering system to truly embrace data redundancy: simple counting of *n*-grams
  - How would MULDER and AskMSR compare?

InsightSoft-M [Soubbotin and Soubbotin 2001,2002]

Application of surface pattern matching techniques directly on the TREC corpus


Question:
What year was Mozart born?
Query:
“Mozart”
Passage With a Query Term
Type of Question:
“When (what-year) -born?”
Snippets
Answer:
“Mozart (1756-1791) Please pin it…”

Patterns for this Query Type:
1. In strict order: capitalized word; parenthesis; four digits; dash; four digits; parenthesis
2. In any word: capitalized word; “in”; four digits; “born”
3. …

InsightSoft-M: Patterns

Some patterns for “What is” questions:

<A; is/are;[a/an/the]; X>
<X; is/are;[a/an/the]; A>
Example: “Michigan's state flower is the apple blossom”
(23 correct responses in TREC 2001)

<A; comma; [a/an/the]; X; [comma/period]>
<X; comma; [a/an/the]; A; [comma/ period]>
Example: "Moulin Rouge, a cabaret "
(26 correct responses)

<A; [comma]; or; X; [comma]>
Example: "shaman, or tribal magician,“
(12 correct responses)

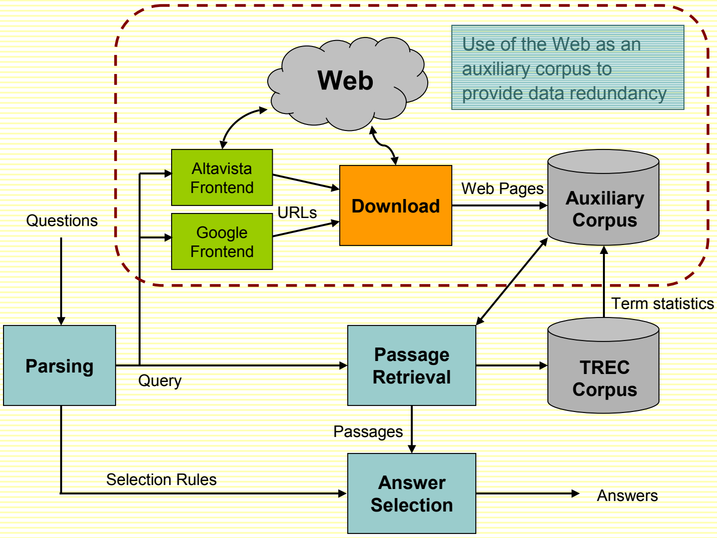
<A; [comma]; [also] called; X [comma]>
< X; [comma]; [also] called; A [comma]>
<X; is called; A> <A; is called; X>
Example: "naturally occurring gas called methane“
(10 correct responses)

InsightSoft-M: Evaluation

- End-to-end performance:
  - TREC 2001: MRR 0.676 (strict) 0.686 (lenient)
  - TREC 2002: CWS 0.691, 54.2% correct
- Observations:
  - Unclear how precision of patterns is controlled
  - Although the system used only the TREC corpus, it demonstrates the power of surface pattern matching

MultiText

U. Waterloo: [Clarke *et al.* 2001b, 2002]



MultiText: TREC 2001

- Download top 200 Web documents to create an auxiliary corpus
- Select 40 passages from Web documents to supplement passages from TREC corpus
- Candidate term weighting:

$$w_t = c_t \log\left(N / f_t\right)$$

N = sum of lengths of all documents in the corpus
f_t = number of occurrences of *t* in corpus
c_t = number of distinct passages in which *t* occurs

"Redundancy factor" where Web passages help

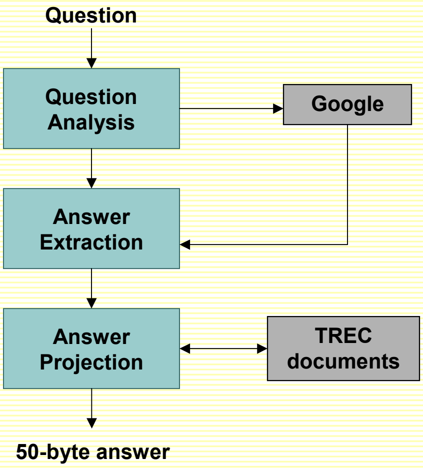
- End-to-end performance: TREC 2001 (official)
  - MRR 0.434 (strict) 0.457 (lenient)
  - Web redundancy contributed to 25% of performance

MultiText: TREC 2002

- Same basic setup as MultiText in TREC 2001
- Two sources of Web data:
  - One terabyte crawl of the Web from mid-2001
  - AltaVista
- End-to-end performance: TREC 2002 (official)
  - 36.8% correct, CWS 0.512
  - Impact of AltaVista not significant (compared to using 1TB of crawled data)

Shapaqa

ILK, Tilburg University: [Buchholz 2001]



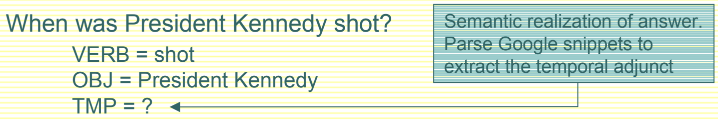
Analyze Google snippets for semantic roles. Match semantic role from question with those extracted from Google snippets.

Return most frequently-occurring answer

Find Web answer that occurs in TREC sentences (from NIST documents)

Shapaqa: Overview

- Extracts answers by determining the semantic role the answer is likely to play
  - **SBJ** (subject), **OBJ** (object), **LGC** (logical subjects of passive verbs), **LOC** (locative adjunct), **TMP** (temporal adjunct), **PRP** (adjust of purpose and reason), **MNR** (manner adjunct), **OTH** (unspecified relation between verb and PP)
  - Does not utilize named-entity detection



When was President Kennedy shot?

VERB = shot

OBJ = President Kennedy

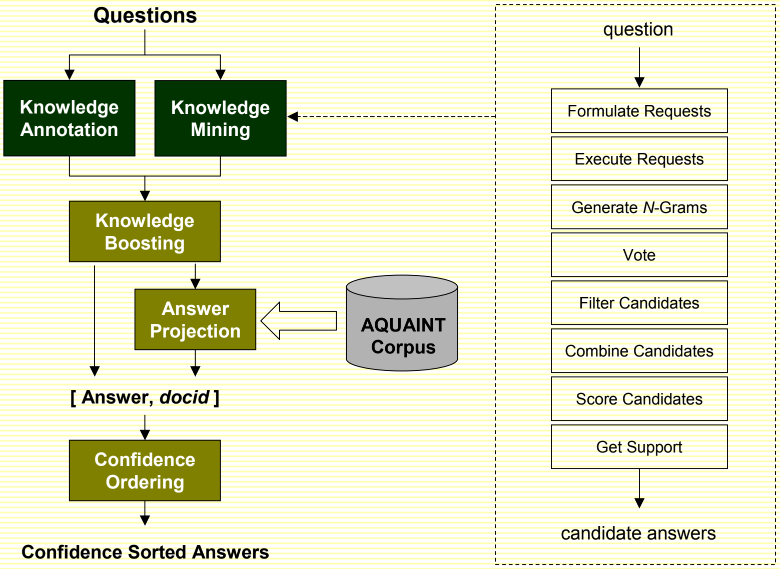
TMP = ?

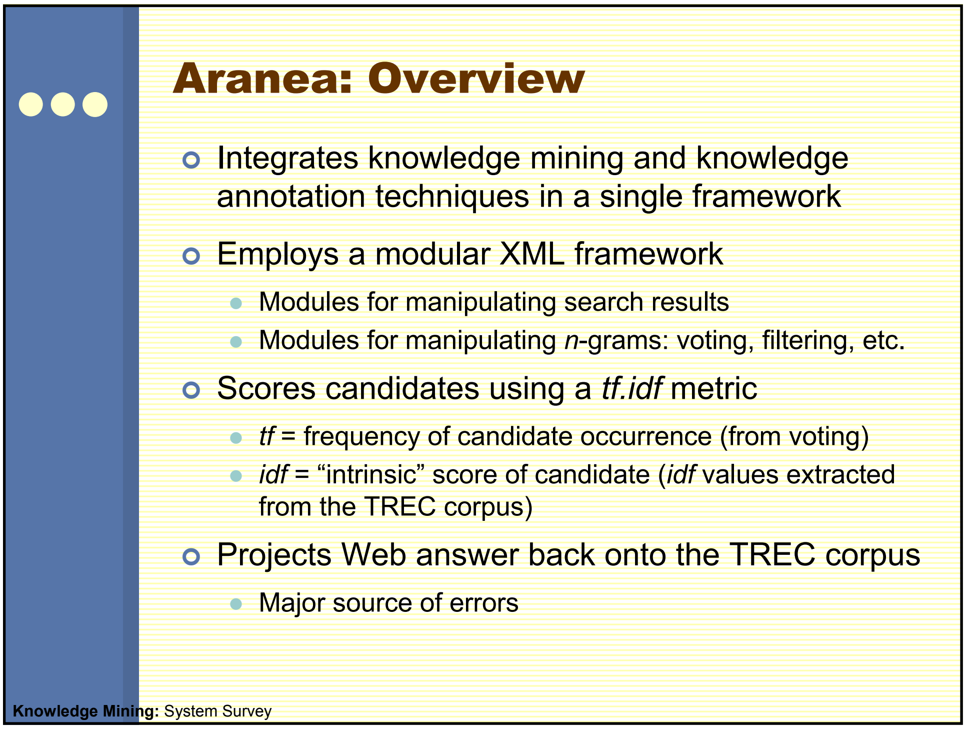
Semantic realization of answer. Parse Google snippets to extract the temporal adjunct

- End-to-end performance: TREC-2001, official
  - MRR: 0.210 (strict), 0.234 (lenient)

Aranea

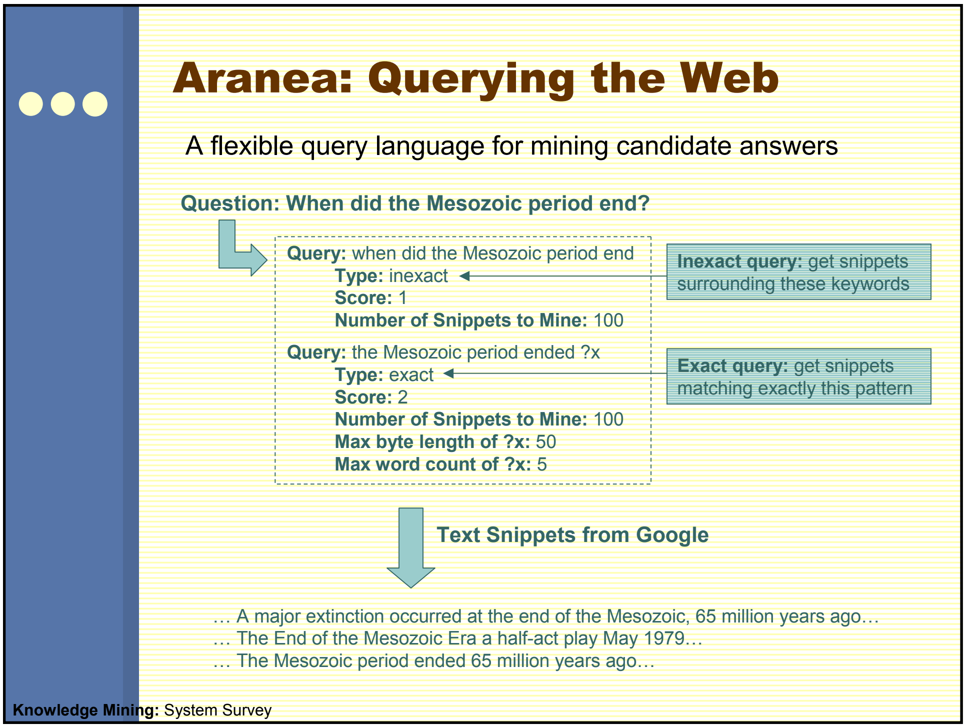
MIT: [Lin, J. *et al.* 2002]





Aranea: Overview

- Integrates knowledge mining and knowledge annotation techniques in a single framework
- Employs a modular XML framework
  - Modules for manipulating search results
  - Modules for manipulating *n*-grams: voting, filtering, etc.
- Scores candidates using a *tf.idf* metric
  - *tf* = frequency of candidate occurrence (from voting)
  - *idf* = “intrinsic” score of candidate (*idf* values extracted from the TREC corpus)
- Projects Web answer back onto the TREC corpus
  - Major source of errors



Aranea: Querying the Web

A flexible query language for mining candidate answers

Question: When did the Mesozoic period end?

Query: when did the Mesozoic period end
Type: inexact
Score: 1
Number of Snippets to Mine: 100

Inexact query: get snippets surrounding these keywords

Query: the Mesozoic period ended ?x
Type: exact
Score: 2
Number of Snippets to Mine: 100
Max byte length of ?x: 50
Max word count of ?x: 5

Exact query: get snippets matching exactly this pattern

Text Snippets from Google

… A major extinction occurred at the end of the Mesozoic, 65 million years ago…
… The End of the Mesozoic Era a half-act play May 1979…
… The Mesozoic period ended 65 million years ago…

Aranea: Evaluation

- End-to-end performance: TREC 2002 (official)
  - Official score: 30.4% correct, CWS 0.433
  - Knowledge mining component contributed 85% of the performance
- Observations:
  - Projection performance: ~75%
  - Without answer projection: 36.6% correct, CWS 0.544
  - Knowledge mining component: refinement of many techniques introduced in AskMSR

Textmap

USC/ISI: [Hermjakob *et al.* 2002]

- Natural language based reformulation resource

cf. S-Rules [Katz and Levin 1988], DIRT [Lin and Pantel 2001ab]

:anchor-pattern “SOMEBODY_1 died of SOMETHING_2.”
:is-equivalent-to “SOMEBODY_1 died from SOMETHING_2.”
:is-equivalent-to “SOMEBODY_1’s death from SOMETHING_2.”
:answers “How did SOMEBODY_1 die?” :answer SOMETHING_2

:anchor-pattern “PERSON_1 invented SOMETHING_2.”
:is-equivalent-to “PERSON_1’s invention of SOMETHING_2”
:answers “Who is PERSON_1?” :answer “the inventor of SOMETHING_2”

- Reformulations are used in two ways:
  - Query expansion: retrieve more relevant documents
  - Answer selection: rank and choose better answers

Question: Who was Johan Vaaler?
Reformulation: Johan Vaaler’s invention of <what>
Text: … Johan Vaaler’s invention of the paper clip …
Answer: the inventor of the paper clip

Textmap

- Applied reformulations to two sources
  - IR on TREC collection: modules developed for Webclopedia [Hovy *et al.* 2001ab,2002]
  - IR on the Web: manually specified query expansion, e.g., morphological expansion, adding synonyms, etc.
- End-to-end performance: TREC 2002 (official)
  - 29.8% correct, CWS 0.498

Reformulations in TextMap are manual generalizations of automatically derived patterns…

Pattern Learning [Ravichandran and Hovy 2002]

Automatically learn surface patterns for answering questions from the World Wide Web

BIRTHYEAR questions: When was <NAME> born?

<NAME> was born on <BIRTHYEAR>
<NAME> (<BIRTHYEAR>-
born in <BIRTHYEAR>, <NAME>
…

cf. [Zhang and Lee 2002]

1. Start with a “seed”, e.g. (Mozart, 1756)
2. Download Web documents using a search engine
3. Retain sentences that contain both question and answer terms
4. Construct a suffix tree for extracting the longest matching substring that spans <QUESTION> and <ANSWER>
   - Suffix Trees: used in computational biology for detecting DNA sequences [Gusfield 1997; Andersson 1999]
5. Calculate precision of patterns
   - Precision for each pattern = # of patterns with correct answer / # of total patterns

Pattern Learning

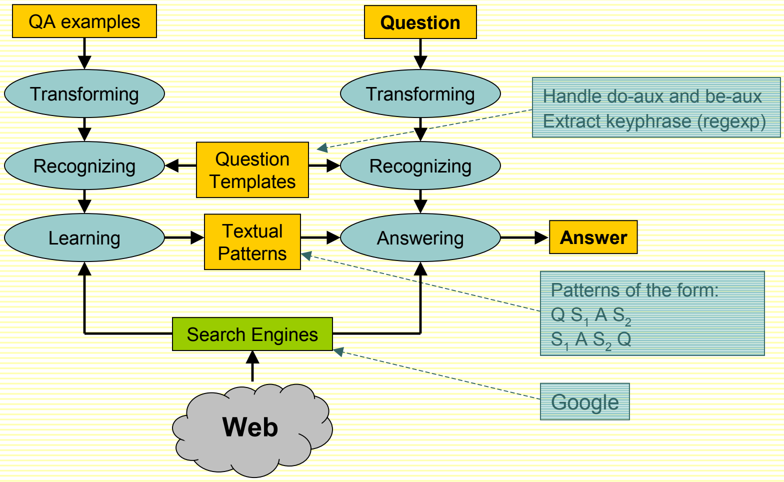
Example: DISCOVERER questions

1.0	when <ANSWER> discovered <NAME>
1.0	<ANSWER>'s discovery of <NAME>
1.0	<ANSWER>, the discoverer of <NAME>
1.0	<ANSWER> discovers <NAME>
1.0	<ANSWER> discover <NAME>
1.0	<ANSWER> discovered <NAME>, the
1.0	discovery of <NAME> by <ANSWER>
0.95	<NAME> was discovered by <ANSWER>
0.91	of <ANSWER>'s <NAME>
0.9	<NAME> was discovered by <ANSWER> in

- Observations
  - Surface patterns perform better on the Web than on the TREC corpus
  - Surface patterns could benefit from notion of constituency, e.g., match not words but NPs, VPs, etc.

LAMP

National University of Singapore: [Zhang and Lee 2002]



QA examples

Question

Transforming

Transforming

Handle do-aux and be-aux
Extract keyphrase (regexp)

Recognizing

Question Templates

Recognizing

Learning

Textual Patterns

Answering

Answer

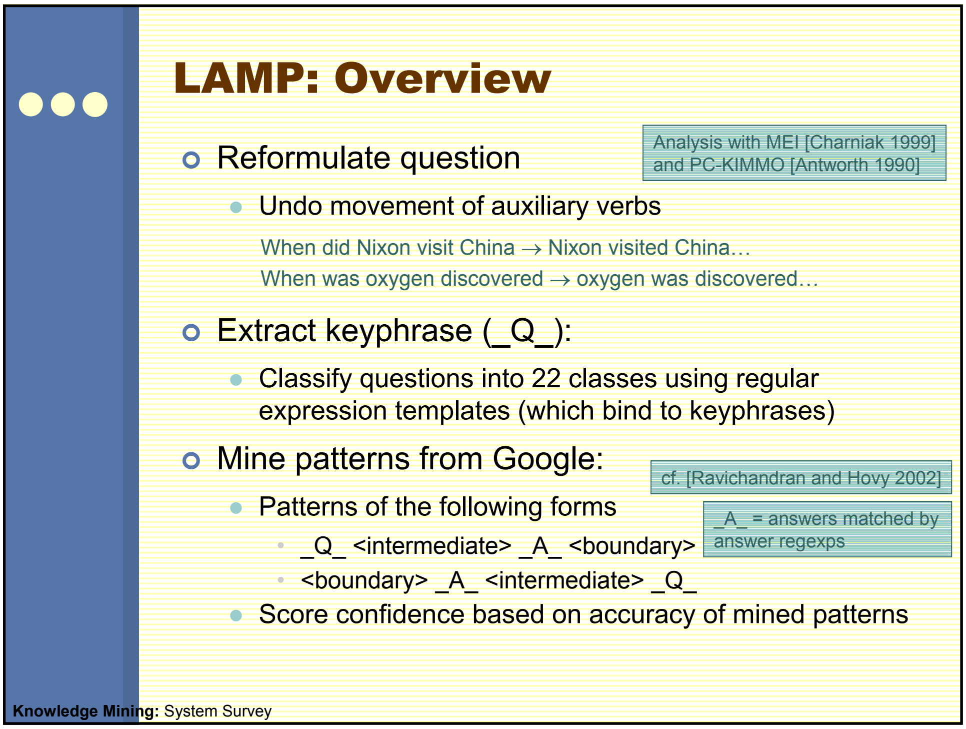
Patterns of the form:
Q S_1 A S_2
S_1 A S_2 Q

Search Engines

Google

Web

http://www.comp.nus.edu.sg/~smadellz/lamp/lamp_index.html



LAMP: Overview

Analysis with MEI [Charniak 1999] and PC-KIMMO [Antworth 1990]

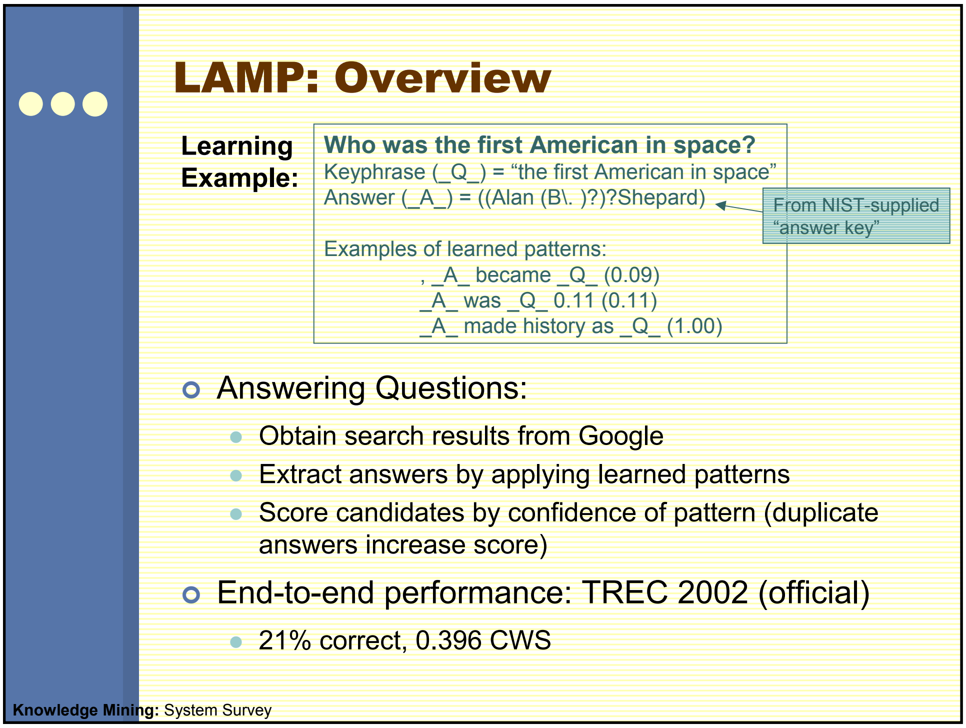
- Reformulate question
  - Undo movement of auxiliary verbs

    When did Nixon visit China → Nixon visited China…

    When was oxygen discovered → oxygen was discovered…
- Extract keyphrase (_Q_):
  - Classify questions into 22 classes using regular expression templates (which bind to keyphrases)
- Mine patterns from Google:
  - Patterns of the following forms
    - _Q_ <intermediate> _A_ <boundary>
    - <boundary> _A_ <intermediate> _Q_
  - Score confidence based on accuracy of mined patterns

cf. [Ravichandran and Hovy 2002]

_A_ = answers matched by answer regexps



LAMP: Overview

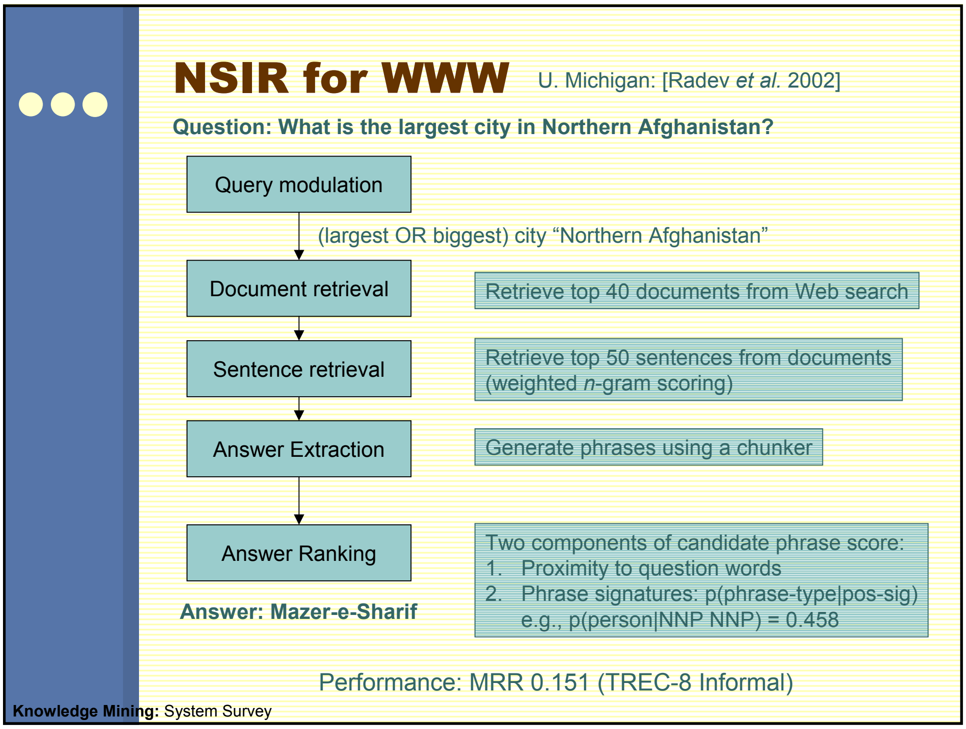
Learning Example:

Who was the first American in space?
Keyphrase (_Q_) = “the first American in space”
Answer (_A_) = ((Alan (B\.)?)?Shepard)

From NIST-supplied “answer key”

Examples of learned patterns:
, _A_ became _Q_ (0.09)
_A_ was _Q_ 0.11 (0.11)
_A_ made history as _Q_ (1.00)

- Answering Questions:
  - Obtain search results from Google
  - Extract answers by applying learned patterns
  - Score candidates by confidence of pattern (duplicate answers increase score)
- End-to-end performance: TREC 2002 (official)
  - 21% correct, 0.396 CWS



NSIR for WWW

U. Michigan: [Radev *et al.* 2002]

Question: What is the largest city in Northern Afghanistan?

Query modulation

(largest OR biggest) city "Northern Afghanistan"

Document retrieval

Retrieve top 40 documents from Web search

Sentence retrieval

Retrieve top 50 sentences from documents (weighted *n*-gram scoring)

Answer Extraction

Generate phrases using a chunker

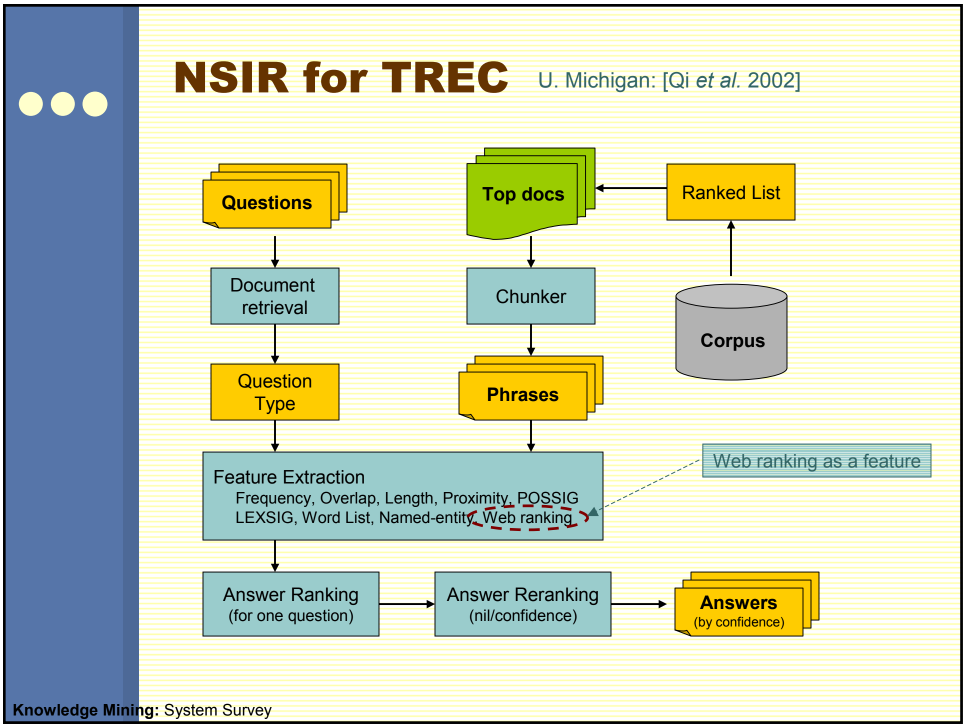
Answer Ranking

Two components of candidate phrase score:

1. Proximity to question words
2. Phrase signatures: p(phrase-type|pos-sig) e.g., p(person|NNP NNP) = 0.458

Answer: Mazer-e-Sharif

Performance: MRR 0.151 (TREC-8 Informal)



NSIR for TREC

U. Michigan: [Qi *et al.* 2002]

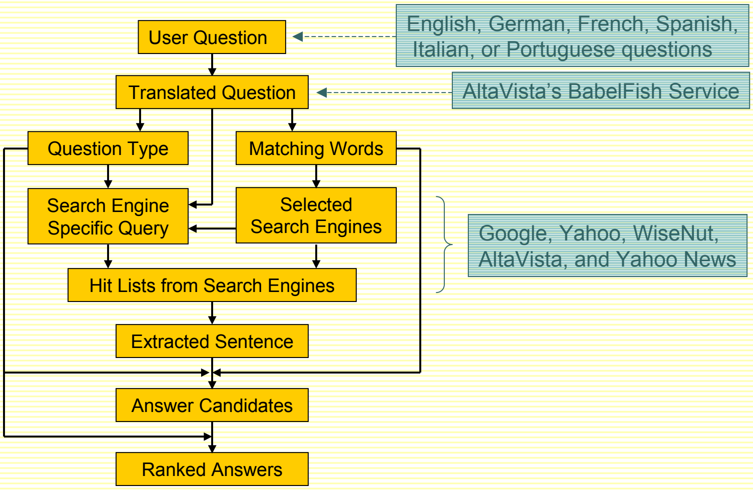
Questions
Top docs
Ranked List
Document retrieval
Chunker
Corpus
Question Type
Phrases
Web ranking as a feature
Feature Extraction
Frequency, Overlap, Length, Proximity, POSSIG
LEXSIG, Word List, Named-entity, Web ranking
Answer Ranking (for one question)
Answer Reranking (nil/confidence)
Answers (by confidence)

NSIR: TREC

- Question classification: allow multiple categories with a probabilistic classifier
- Phrase Extraction: extract phrases from top 20 NIST documents using LT-Chunk
- Feature Extraction: compute nine features of each phrase
  - Web ranking is one such feature
- Answer Ranking: linearly combine individual features to produce final score for each candidate
  - Feature weights specific to each question type
- End-to-end performance: TREC 2002 (official)
  - 17.8% correct, CWS 0.283

AnswerBus

U. Michigan: [Zheng 2002ab]



User Question

English, German, French, Spanish, Italian, or Portuguese questions

Translated Question

AltaVista's BabelFish Service

Question Type

Matching Words

Search Engine Specific Query

Selected Search Engines

Google, Yahoo, WiseNut, AltaVista, and Yahoo News

Hit Lists from Search Engines

Extracted Sentence

Answer Candidates

Ranked Answers

http://misshoover.si.umich.edu/~zzheng/qa-new/

AnswerBus: Overview

- Search query
  - Stopword filtering, low *tf* keyword filtering, some verb conjugation
- Simple sentence scoring:

$$\text{Score} = \begin{cases} q \text{ if } q \geq \lfloor \sqrt{Q-1} \rfloor + 1 \\ 0 \text{ otherwise} \end{cases}$$

Similar to the MITRE Algorithm [Breck *et al.* 2001; Light *et al.* 2001]

q = number of matching words in query
Q = total number of query words

- Other techniques:
  - Question type classification
  - Coreference resolution (in adjacent sentences)

Knowledge Mining:

Selected Techniques

Question Answering Techniques for the World Wide Web

Knowledge Mining Techniques

- Projecting answers onto another corpus
- Using the Web (and WordNet) to rerank answers
- Using the Web to validate answers
  - Verifying the correctness of question answer pairs
  - Estimating the confidence of question answer pairs
- Tweaking search engines: getting the most out of a search
  - Query expansion for search engines
  - Learning search engine specific reformulations

Answer Projection

- Just an artifact of TREC competitions?
  - TREC answers require [answer, docid] pair
  - Document from the TREC corpus must support answer
  - If answers were extracted form an outside source, a supporting TREC document must still be found
- Perhaps not…
  - People prefer paragraph-sized answers [Lin, J. *et al.* 2003]

    find exact answers from the Web (using data redundancy), but present answers from another source
- Sample answer projection algorithms:
  - Use document-retrieval or passage retrieval algorithms
  - query = keywords from question + keywords from answer

Answer Projection Performance

- AskMSR answer projection: [Brill *et al.* 2001]
  - Used the Okapi IR engine (bm25 weighting)
  - Generated query = question + answer
  - Selected top-ranking document as support
  - Performance: ~80% (i.e., 20% of “supporting documents” did not actually support the answer)
- Aranea answer projection: [Lin, J. *et al.* 2002]
  - Projected answer onto NIST-supplied documents
  - Used sliding window technique

    Window score = # keywords from question + # keywords from answer (neither term could be zero)
  - Selected document of highest scoring window as support
  - Performance: ~75%

Answer Projection: Analysis

Question: Who was the first black heavyweight champion?
Answer: Jack Johnson

… Louis was the first African-American **heavyweight** since **Jack Johnson** who was allowed to get close to that symbol of ultimate manhood, the **heavyweight** crown …

Question: Who was the Roman god of the sea?
Answer: Neptune

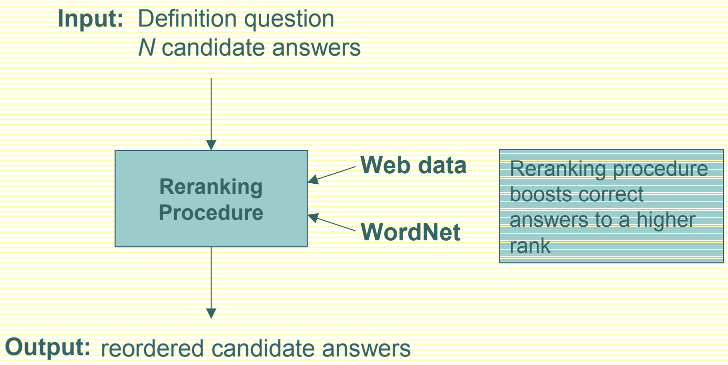
… Romanian Foreign Minister Petre **Roman** Wednesday met at the **Neptune** resort of the Black **Sea** shore with his Slovenian counterpart, Alojz Peterle, …

Question: What is the nickname of Oklahoma?
Answer: Sooner State

… The victory makes the **Sooners** the No. 3 seed in the conference tournament. **Oklahoma State** (23-5, 12-4) will be the fourth seed…

Answer Reranking [Lin, C.Y. 2002]

Use the Web and WordNet to rerank answers to definition questions



Input: Definition question
N candidate answers

Reranking Procedure

Web data

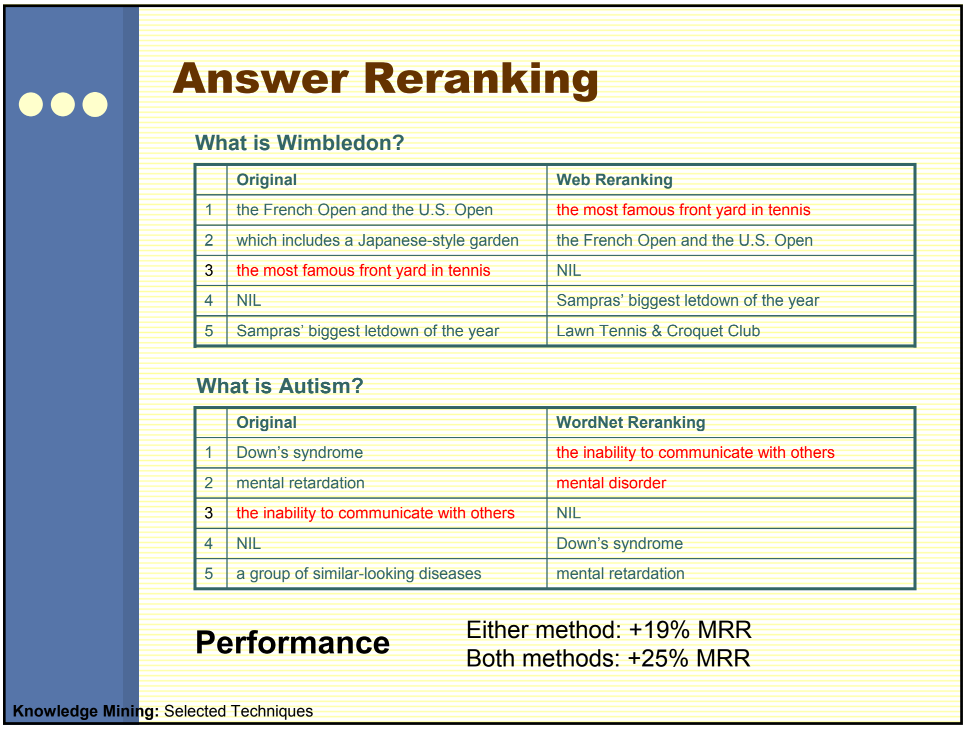
WordNet

Reranking procedure boosts correct answers to a higher rank

Output: reordered candidate answers

Answer Reranking

- Web reranking
  - Obtain pages from Google and calculate *tf.idf* values for keywords
  - matching score = sum of *tf.idf* values of keywords in answer candidates
  - new score = original candidate score × matching score
- WordNet reranking
  - Create a definition database from WordNet glosses; calculate *idf* values for keywords
  - matching score = sum of *idf* values of keywords in answer candidates
  - new score = original candidate score × matching score



Answer Reranking

What is Wimbledon?

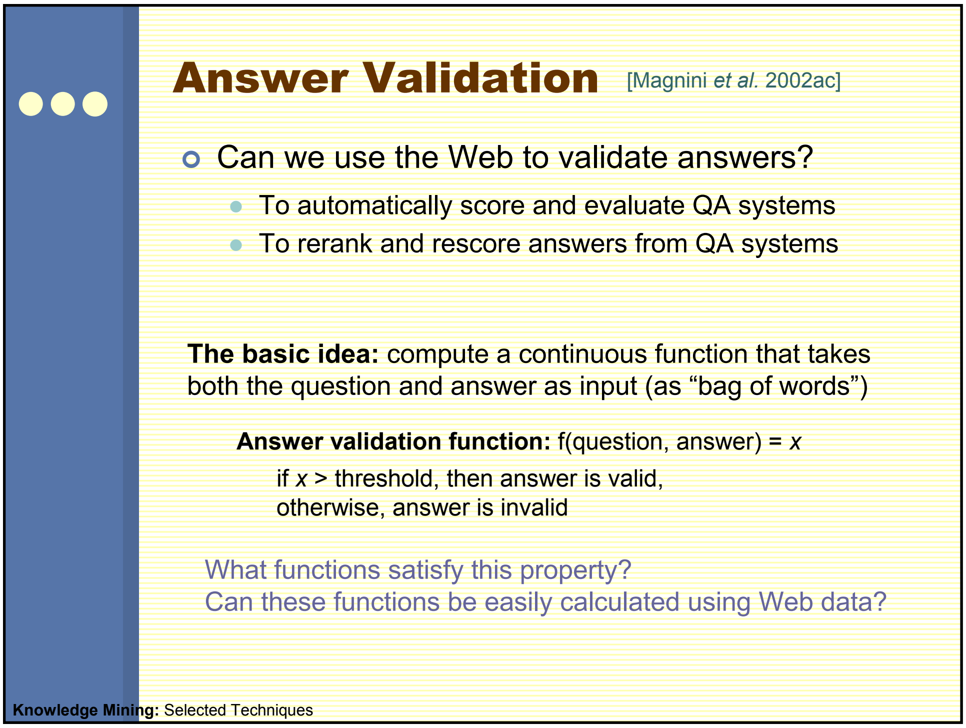
	Original	Web Reranking
1	the French Open and the U.S. Open	the most famous front yard in tennis
2	which includes a Japanese-style garden	the French Open and the U.S. Open
3	the most famous front yard in tennis	NIL
4	NIL	Sampras' biggest letdown of the year
5	Sampras' biggest letdown of the year	Lawn Tennis & Croquet Club

What is Autism?

	Original	WordNet Reranking
1	Down's syndrome	the inability to communicate with others
2	mental retardation	mental disorder
3	the inability to communicate with others	NIL
4	NIL	Down's syndrome
5	a group of similar-looking diseases	mental retardation

Performance

Either method: +19% MRR
Both methods: +25% MRR



Answer Validation

[Magnini *et al.* 2002ac]

- Can we use the Web to validate answers?
  - To automatically score and evaluate QA systems
  - To rerank and rescore answers from QA systems

The basic idea: compute a continuous function that takes both the question and answer as input (as "bag of words")

Answer validation function: f(question, answer) = *x*

if *x* > threshold, then answer is valid,
otherwise, answer is invalid

What functions satisfy this property?
Can these functions be easily calculated using Web data?

Answer Validation

Three different answer validation functions:
(various statistical measures of co-occurrence)

All three can be easily calculated from search engine results

1. **Pointwise Mutual Information (PMI)**
2. **Maximal Likelihood Ratio (MLHR)**
3. **Corrected Conditional Probability (CCP)**

$$CCP(Qsp, Asp) = \frac{p(Asp \mid Qsp)}{p(Asp)^{2/3}} \approx \frac{\text{hits}\,(Qsp\ \ \text{NEAR}\ \ Asp)}{\text{hits}\,(Qsp)\ \ \text{hits}(Asp)} MaxPages^{2/3}$$

Treat questions and answers as “bag of words”

Qsp = question sub-pattern (content words + expansions)
Asp = answer sub-pattern
MaxPages = total number of pages in search engine index

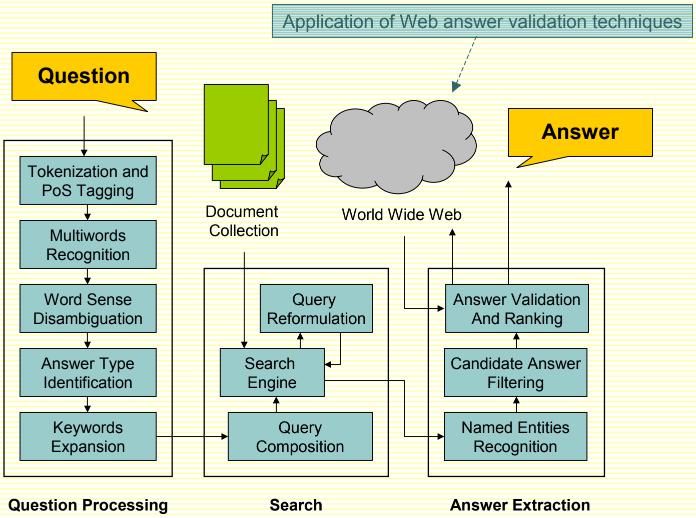
Answer Validation Performance

Evaluation metric: agreement between machine algorithm and human judgment (from TREC)

	Agreement
CCP – relative	81.25%
CCP – absolute	78.42%
PMI – relative	79.56%
PMI – absolute	77.79%
MLHR – relative	79.60%
MLHR – absolute	77.40%

Absolute threshold: fixed threshold
Relative threshold: threshold set to a percentage of the score of the highest scoring answer

DIOGENE [Magnini *et al.* 2001, 2002b]


Application of Web answer validation techniques

Question

Tokenization and PoS Tagging

Multiwords Recognition

Word Sense Disambiguation

Answer Type Identification

Keywords Expansion

Document Collection

World Wide Web

Query Reformulation

Search Engine

Query Composition

Answer

Answer Validation And Ranking

Candidate Answer Filtering

Named Entities Recognition

Question Processing

Search

Answer Extraction

DIOGENE: Answer Validation

- Two measures
  - “Statistical approach”: corrected conditional probability (using Web page hit counts only)
  - “Content-based approach”: co-occurrence between question and answer (from downloaded snippets)
- Performance: TREC 2002 (official)
  - 38.4%, CWS 0.589 (content-based measure)
  - Content-based measure beat statistical measure and combination of both measures
  - Overall contribution of answer validation techniques is unclear

Confidence Estimation

[Xu *et al.* 2002]

Estimating the probability that a question answer pair is correct

- Result useful for confidence estimation
- Similar to Magnini *et al.* except without thresholding

BBN2002B

$p(\text{correct}|Q,A) \approx p(\text{correct}|T, F) \approx p(\text{correct}|T)\times 0.5 + p(\text{correct}|F)\times 0.5$

T = question type
F = frequencies of A in Google summaries

BBN2002C

$p(\text{correct}|Q,A) \approx p(\text{correct}|F, \text{INTREC})$

F = frequencies of A in Google summaries
INTREC = boolean indicator variable, true iff answer also found in TREC

TREC-9 and TREC 2001 questions used for parameter estimation

Confidence Estimation

- Performance: TREC 2002 (official)
  - Baseline (without Web): 18.6% correct, CWS 0.257
  - BBN2002B: 28.8% correct, CWS 0.468
  - BBN2002C: 28.4% correct, CWS 0.499
- Observations
  - Use of Web significantly boosts performance
  - Performance contribution of confidence estimation procedure is unclear

Tweaking Search Engines

"Getting the most out of an existing search engine"

- Large IR literature on query expansion
  - Expand queries based on synonyms and lexical-semantic relations (from WordNet) [Voorhees 1994]

    Even with sense disambiguated queries, synonymy expansion provides little benefit
  - Expand queries based on relevant terms in top-ranking documents [Mitra *et al.* 1998]
  - Expand queries with terms from top-ranking documents that co-occur with query terms [Xu and Croft 2000]

Query Expansion for the Web

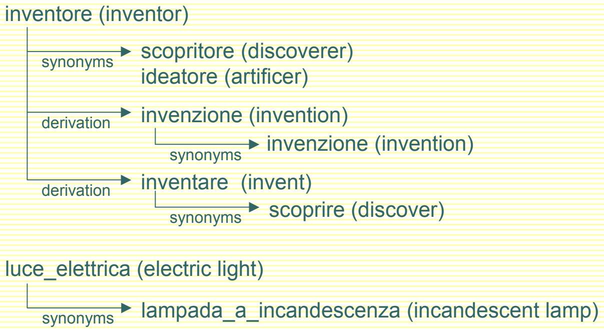
- Query expansion is difficult with Web search engines
  - Search algorithm is hidden: the service must be treated like an opaque black box
  - No principled way for developing query expansion techniques: trial and error required
  - It is beneficial to use more than one service, but how do we assess the relative strengths and weaknesses of each search engine?

Expanding Boolean Queries

[Magnini and Prevete 2000]

Exploiting lexical expansions and boolean compositions

Expand keywords: synonyms and morphological derivations


inventore (inventor)

synonyms → scopritore (discoverer)
ideatore (artificer)

derivation → invenzione (invention)

synonyms → invenzione (invention)

derivation → inventare (invent)

synonyms → scoprire (discover)

luce_elettrica (electric light)

synonyms → lampada_a_incandescenza (incandescent lamp)

How do we combine these keywords into boolean queries?

Query Expansion Strategies

KAS: Keyword "AND" composition Search

Conjoin original keywords

$(\text{inventore} \wedge \text{luce\_elettrica})$

KIS: Keyword Insertion Search

OR of ANDs; each AND clause = original keywords + one derived word

$$\begin{aligned}(\quad & (\text{inventore} \wedge \text{luce\_elettrica} \wedge \text{scopritore}) \\ \vee\ & (\text{inventore} \wedge \text{luce\_elettrica} \wedge \text{ideatore}) \\ \vee\ & (\text{inventore} \wedge \text{luce\_elettrica} \wedge \text{invenzione}) \\ \ldots)\end{aligned}$$

KCS: Keyword Cartesian Search

OR of ANDs; AND clauses = Cartesian product of all derivations

$$\begin{aligned}(\quad & (\text{inventore} \wedge \text{luce\_elettrica}) \\ \vee\ & (\text{inventore} \wedge \text{lampada\_a\_incandescenza}) \\ \vee\ & (\text{scopritore} \wedge \text{luce\_elettrica}) \\ \vee\ & (\text{scopritore} \wedge \text{lampada\_a\_incandescenza}) \\ \ldots)\end{aligned}$$

KAS vs. KIS vs. KCS

- Evaluation: 20 questions, documents from Excite
- Relevance determined by three human judges
- Measures: compared to KAS baseline
  - With *f*-, document ordering is not taken into account
  - With *f*+, document ordering is taken into account

	KIS		KCS	
	f-	***f+***	***f-***	***f+***
QS1	+7%	-15%	+7%	-15%
QS2	-3%	+19%	+59%	+77%
QS3	+18%	+17%	+23%	+17%
All	+19%	+13%	+33%	+22%

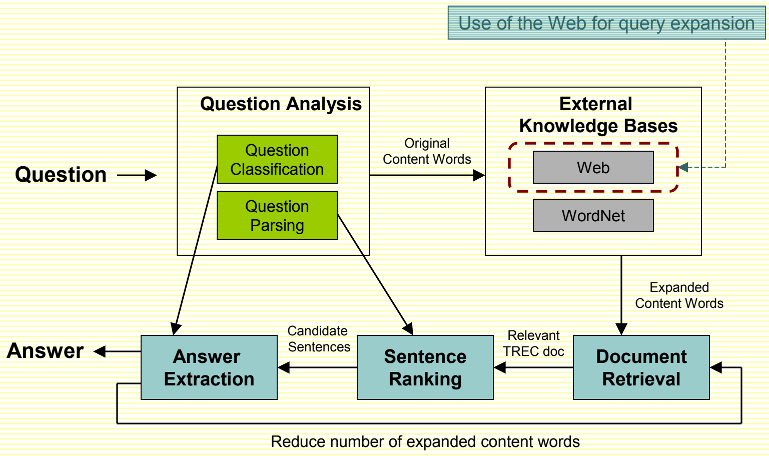
QS1: Subset of questions where number of morphological derivations and synonyms is greater than 3

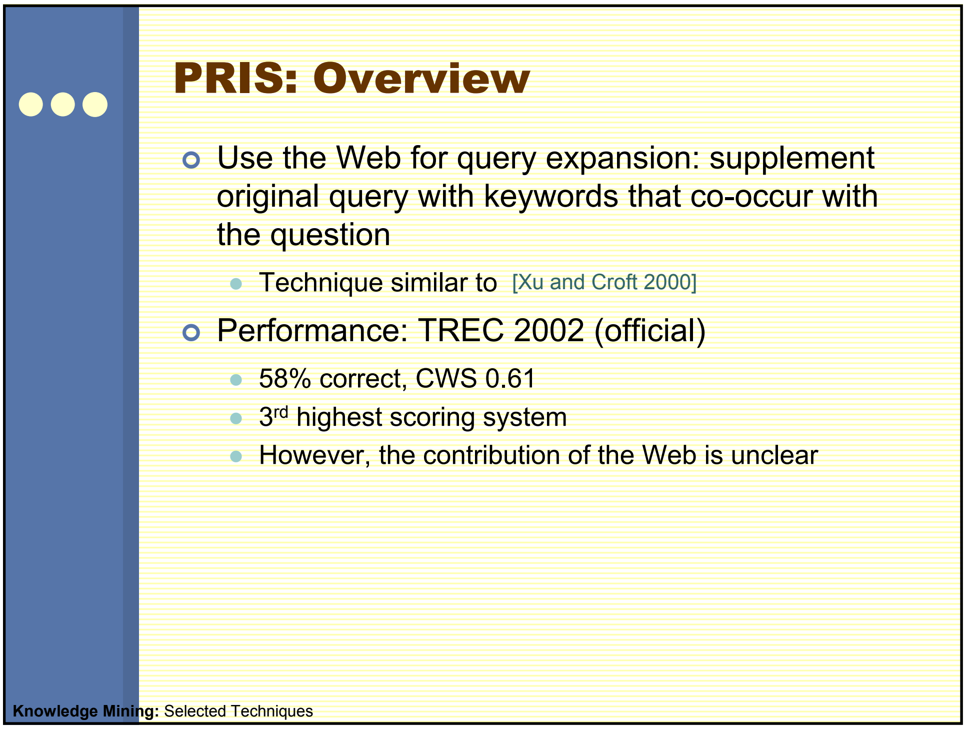
QS2: equal to 2 or 3

QS3: less than 2

Web Query Expansion: PRIS

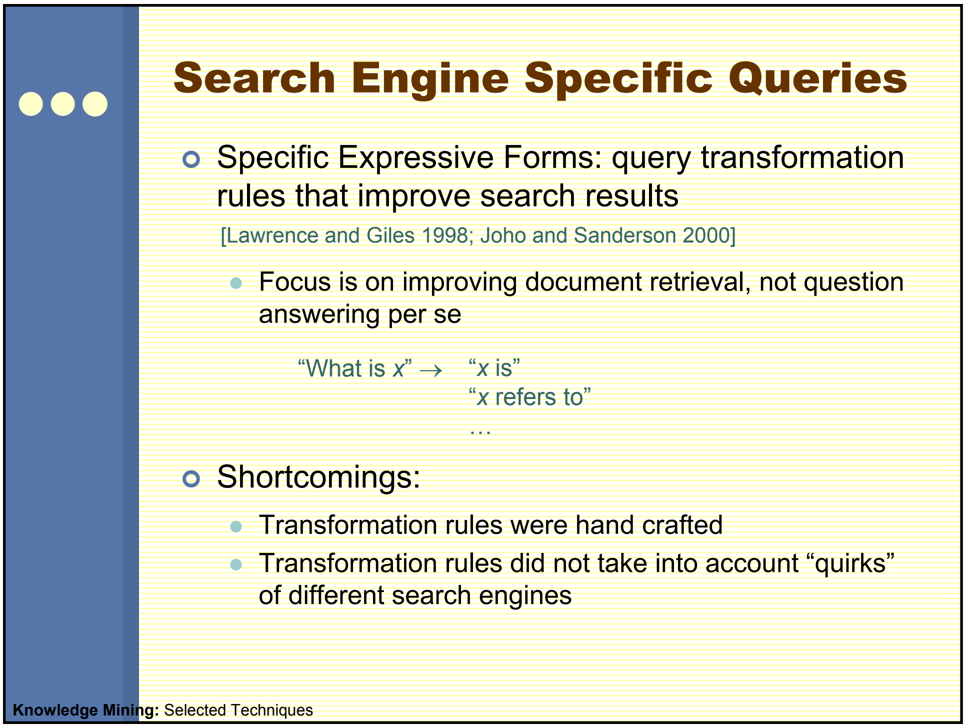
[Yang and Chua 2002]





PRIS: Overview

- Use the Web for query expansion: supplement original query with keywords that co-occur with the question
  - Technique similar to [Xu and Croft 2000]
- Performance: TREC 2002 (official)
  - 58% correct, CWS 0.61
  - 3rd highest scoring system
  - However, the contribution of the Web is unclear



Search Engine Specific Queries

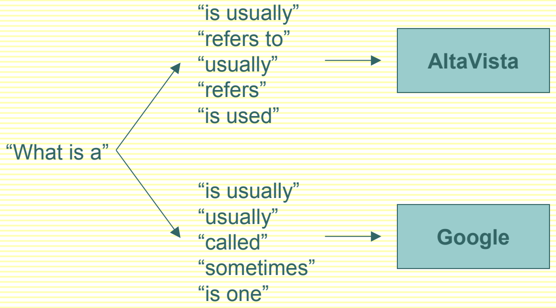
- Specific Expressive Forms: query transformation rules that improve search results

  [Lawrence and Giles 1998; Joho and Sanderson 2000]
  - Focus is on improving document retrieval, not question answering per se

    "What is *x*" → "*x* is"
    "*x* refers to"
    …
- Shortcomings:
  - Transformation rules were hand crafted
  - Transformation rules did not take into account "quirks" of different search engines

Tritus [Agichtein *et al.* 2001]

Learn query transformations optimized for each search engine



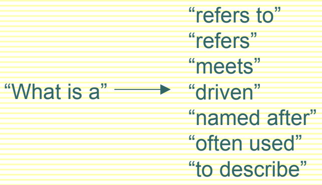
Transformations capture the "quirks" of different search engines

Tritus: Transformation Learning

Select Question Phrase (QP): Group questions by their initial tokens

Who was Albert Einstein?
How do I fix a broken television?
Where can I find a Lisp Machine?
What is a pulsar?

Generate Candidate Transformations (TR): From <Q, A> pairs, generate all *n*-grams of answers that do not contain content words



Two components to TR score:

- Frequency of co-occurrence between TR and QP
- Okapi bm25 weighting on TR
  [Robertson and Walker 1997; Robertson et al. 1998]

Tritus: Transformation Learning

Train Candidate Transformations (TR) against search engines

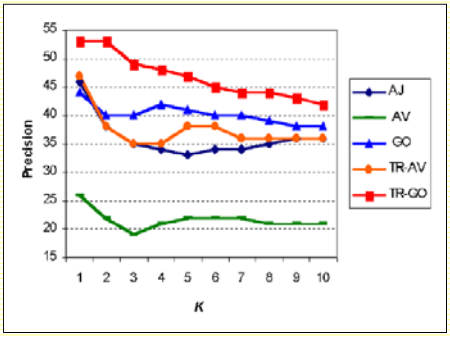
C = question – question phrase

1. Break questions into {QP C}
2. Submit the query {TR C} to various search engines
3. Score TR with respect to known answer (Okapi bm25 weighting)
4. Keep highest scoring TR for each particular search engine

Experimental Setting:

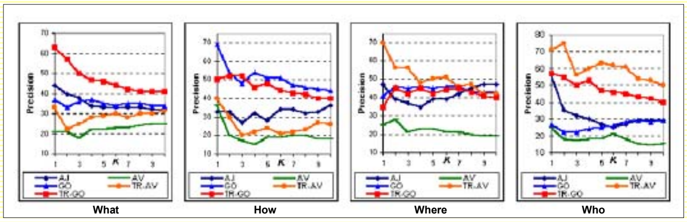
- Training Set
  - ~10k <Question, Answer> pairs from Internet FAQs
  - Seven question types
  - Three search Engines (Google, AltaVista, AskJeeves)
- Test Set
  - 313 questions in total (~50 per question type)
  - Relevance of documents manually evaluated by human judges

Tritus: Results



Indeed, transformations learned for each search engine were slightly different

Tritus + search engine performs better than search engine alone



What

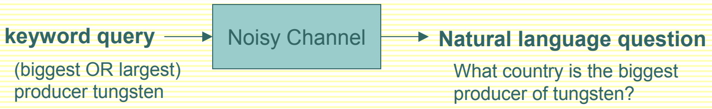
How

Where

Who

QASM [Radev *et al.* 2001]

- QASM = Question Answering using Statistical Models cf. [Mann 2001, 2002]
- Query reformulation using a noisy channel translation model



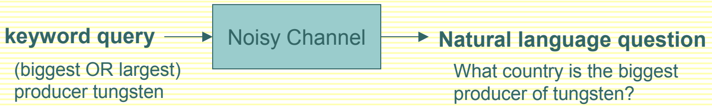
keyword query
(biggest OR largest) producer tungsten
Noisy Channel
Natural language question
What country is the biggest producer of tungsten?

Setup: the keyword query is somehow “scrambled” in the noisy channel and converted into a natural language question

Task: given the natural language question and known properties about the noisy channel, recover the keyword query

Applications of similar techniques in other domains: machine translation [Brown *et al.* 1990],
speech processing [Jelinek 1997],
information retrieval [Berger and Lafferty 1999]

QASM: Noisy Channels



keyword query
(biggest OR largest) producer tungsten
Noisy Channel
Natural language question
What country is the biggest producer of tungsten?

What is the noisy channel “allowed to do”?

Channel Operators = possible methods by which the message can be corrupted

DELETE: e.g., delete prepositions, stopwords, etc.

REPLACE: e.g., replace the *n*-th noun phrase with WordNet expansions

DISJUNCT: e.g., replace the *n*-th noun phrase with OR disjunction

Once the properties of the noisy channel are learned, we can “decode” natural language questions into keyword queries

QASM: Training

- Training using EM Algorithm
  - Use {Question, Answer} pairs from TREC (and from custom collection)
  - Measure the “fitness” of a keyword query by scoring the documents it returns
  - Maximize total reciprocal document rank
- Evaluation: test set of 18 questions
  - Increase of 42% over the baseline
  - For 14 of the questions, sequence of same two operators were deemed the best: delete stopwords and delete auxiliary verbs

Couldn’t we have hand-coded these two operators from the beginning?

Knowledge Mining:

Challenges and Potential Solutions

Question Answering Techniques for the World Wide Web

Knowledge Mining: Challenges

- Search engine behavior changes over time
- Sheer amount of useless data floods out answers
- Anaphora poses problems

> Andorra is a tiny land-locked country in southwestern Europe, between France and Spain.
> …
> Tourism, the largest sector of **its** tiny, well-to-do economy, accounts for roughly 80% of GDP…

What is the biggest sector in Andorra's economy? I don't know

More Challenges

- Answers change over time

Who is the governor of Alaska?
What is the population of Gambia?

- Relative time and temporal expressions complicate analysis
  - Documents refer to events in the past or future (relative to the date the article was written)

> Date: January 2003 … Five years ago, when **Bill Clinton** was still the president of the United States…

Who is the president of the United States? Bill Clinton

Even More Challenges

- Surface patterns are often wrong
  - No notion of constituency

In **May Jane Goodall** spoke at Orchestra Hall in Minneapolis/St. Paul…

Who spoke at Orchestra Hall? May Jane Goodall

  - Patterns can be misleading

The **55 people in Massachusetts** that have suffered from the recent outbreak of…

What is the population of Massachusetts? 55 people

- Most popular ≠ correct

What is the tallest mountain in Europe?

Most common incorrect answer = Mont Blanc (4807m)
Correct answer = Mount Elbrus (5642m)

Still More Challenges

- “Bag-of-words” approaches fail to capture syntactic relations
  - Named-entity detection alone isn’t sufficient to determine the answer!

Lee Harvey Oswald, the gunman who assassinated President **John F. Kennedy**, was later shot and killed by **Jack Ruby**.

Who killed Lee Harvey Oswald? John F. Kennedy

- Knowledge coverage is not consistent

When was Albert Einstein born? March 14, 1879
When was Alfred Einstein born? [Who’s Alfred Einstein?]

Albert Einstein is more famous than Alfred Einstein, so questions about Alfred are “overloaded” by information about Albert.

Really Hard Challenges

- Myths and Jokes

In March, 1999, **Trent Lott** claimed to have invented the paper clip in response to Al Gore’s claim that he invented the Internet

Who invented the paper clip? Trent Lott

George Bush Jokes…George Bush thinks that **Steven Spielberg** is the Prime Minister of Israel…

Who is the Prime Minister of Israel? Steven Spielberg

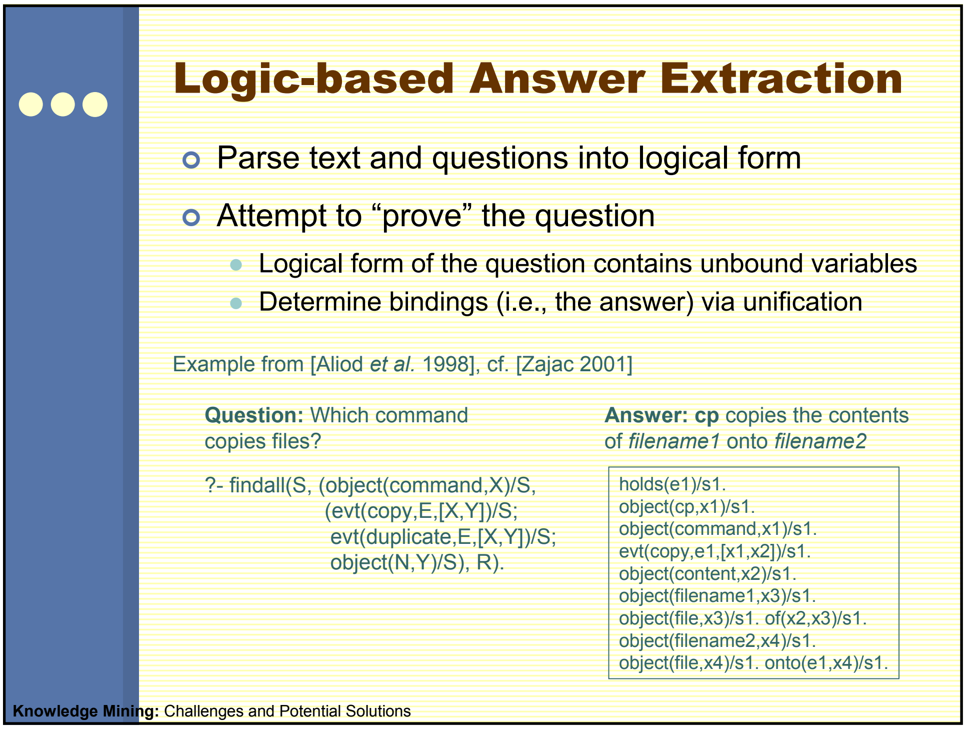
Because: Who is the Prime Minister of Israel?
→ **X** is the Prime Minister of Israel

Where does Santa Claus live?
What does the Tooth Fairy leave under pillows?
How many horns does a unicorn have?

We really need semantics to solve these problems!

NLP Provides Some Solutions

- Linguistically-sophisticated techniques:
  - Parse embedded constituents (Bush thinks that…)
  - Determine the correct semantic role of the answer (Who visited whom?)
  - Resolve temporal referring expressions (Last year…)
  - Resolve pronominal anaphora (It is the tallest…)
- Genre classification [Biber 1986; Kessler *et al.* 1997]
  - Determine the type of article
  - Determine the “authority” of the article (based on sentence structure, etc.)



Logic-based Answer Extraction

- Parse text and questions into logical form
- Attempt to “prove” the question
  - Logical form of the question contains unbound variables
  - Determine bindings (i.e., the answer) via unification

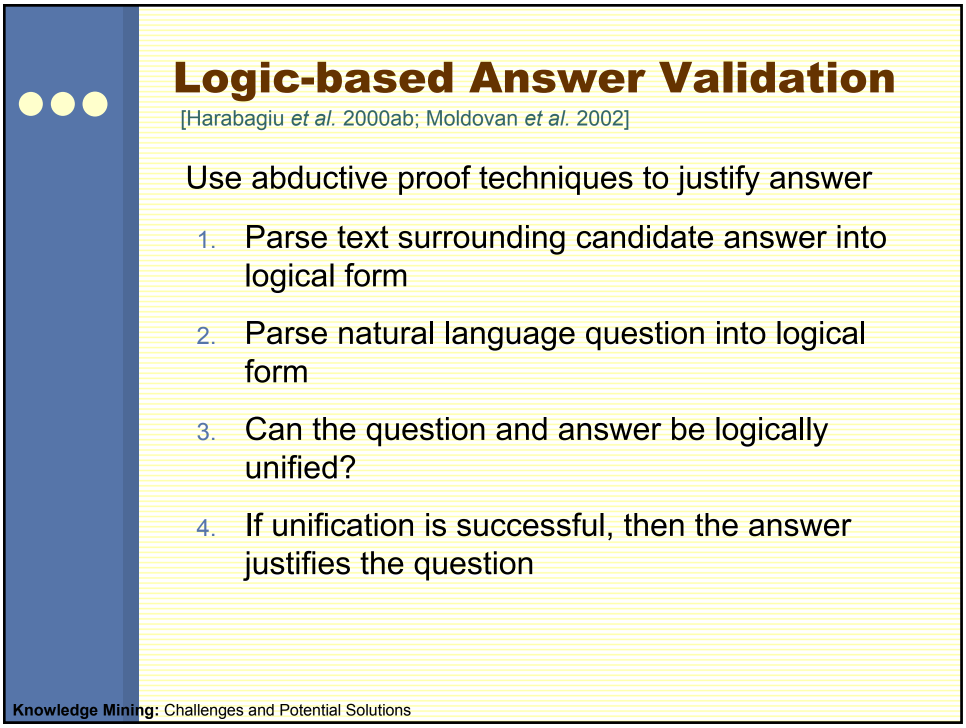
Example from [Aliod *et al.* 1998], cf. [Zajac 2001]

Question: Which command copies files?

```
?- findall(S, (object(command,X)/S,
              (evt(copy,E,[X,Y])/S;
               evt(duplicate,E,[X,Y])/S;
               object(N,Y)/S), R).
```

Answer: cp copies the contents of *filename1* onto *filename2*

```
holds(e1)/s1.
object(cp,x1)/s1.
object(command,x1)/s1.
evt(copy,e1,[x1,x2])/s1.
object(content,x2)/s1.
object(filename1,x3)/s1.
object(file,x3)/s1. of(x2,x3)/s1.
object(filename2,x4)/s1.
object(file,x4)/s1. onto(e1,x4)/s1.
```



Logic-based Answer Validation

[Harabagiu *et al.* 2000ab; Moldovan *et al.* 2002]

Use abductive proof techniques to justify answer

1. Parse text surrounding candidate answer into logical form
2. Parse natural language question into logical form
3. Can the question and answer be logically unified?
4. If unification is successful, then the answer justifies the question

How Can Relations Help?

- Lexical content alone cannot capture meaning

The bird ate the snake.
The snake ate the bird.

the largest planet's volcanoes
the planet's largest volcanoes

the meaning of life
a meaningful life

the house by the river
the river by the house

- Two phenomena where syntactic relations can overcome failures of "bag-of-words" approaches

  [Katz and Lin 2003]

  - **Semantic Symmetry** – selectional restrictions of different arguments of the same head overlap
  - **Ambiguous Modification** – certain modifiers can potentially modify a large number of heads

Semantic Symmetry

The selectional restrictions of different arguments of the same head overlap, e.g., when *verb(x,y)* and *verb(y,x)* can both be found in the corpus

Question: What do frogs eat?

Correct lexical content, correct syntactic relations

(1) Adult **frogs eat** mainly insects and other small animals, including earthworms, minnows, and spiders.

Correct lexical content, incorrect syntactic relations

(2) Alligators **eat** many kinds of small animals that live in or near the water, including fish, snakes, **frogs**, turtles, small mammals, and birds.

(3) Some bats catch fish with their claws, and a few species **eat** lizards, rodents, small birds, tree **frogs**, and other bats.

Ambiguous Modification

Some modifiers can potentially modify a large number of co-occurring heads

Question: What is the largest volcano in the Solar System?

Correct lexical content, correct syntactic relations

(1) Mars boasts many extreme geographic features; for example, Olympus Mons, is the **largest volcano in the solar system**.

(2) Olympus Mons, which spans an area the size of Arizona, is the **largest volcano in the Solar System**.

Correct lexical content, incorrect syntactic relations

(3) The Galileo probe's mission to Jupiter, the **largest** planet **in the Solar system**, included amazing photographs of the **volcanoes** on Io, one of its four most famous moons.

(4) Even the **largest volcanoes** found on Earth are puny in comparison to others found around our own cosmic backyard, **the Solar System**.

Sapere: Using NLP Selectively

[Lin, J. 2001; Katz and Lin 2003]

- Sophisticated linguistic techniques are too brittle to apply indiscriminately

  Natural language techniques often achieve high precision, but poor recall

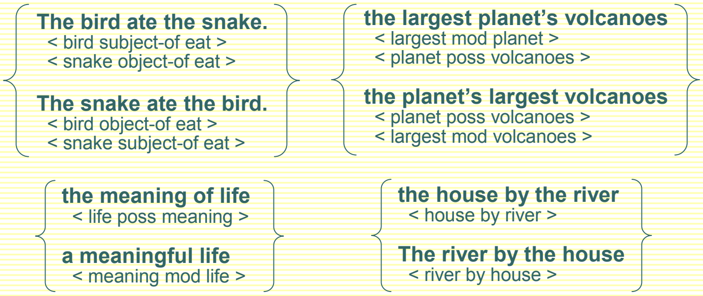
- Simple and robust statistical techniques should not be abandoned
- Sophisticated linguistic techniques should be applied only when necessary, e.g., to handle
  - Semantic symmetry
  - Ambiguous modification
- Our prototype Sapere system is specially designed to handle these phenomena

Using Syntactic Relations

- Automatically extract syntactic relations from questions and corpus, e.g.,
  - Subject-verb-object relations
  - Adjective-noun modification relations
  - Possessive relations
  - NP-PP attachment relations
- Match questions and answers at the level of syntactic relations

Why Syntactic Relations?

Syntactic relations can approximate “meaning”



The bird ate the snake.
< bird subject-of eat >
< snake object-of eat >

The snake ate the bird.
< bird object-of eat >
< snake subject-of eat >

the largest planet’s volcanoes
< largest mod planet >
< planet poss volcanoes >

the planet’s largest volcanoes
< planet poss volcanoes >
< largest mod volcanoes >

the meaning of life
< life poss meaning >

a meaningful life
< meaning mod life >

the house by the river
< house by river >

The river by the house
< river by house >

Benefit of Relations

Preliminary experiments with the WorldBook Encyclopedia show significant increase in precision

	Sapere	Baseline
Avg. # of sentence returned	4	43.88
Avg. # of correct sentences	3.13	5.88
Avg. precision	0.84	0.29

Sapere: entire corpus is parsed into syntactic relations, relations are matched at the sentential level

Baseline: standard boolean keyword retriever (indexed at sentential level)

Test set = 16 question hand-selected questions designed to illustrate semantic symmetry and ambiguous modification

TREC Examples

Ambiguous modification is prevalent in the TREC corpus

(Q1003) What is the highest dam in the U.S.?

Typical wrong answers from the TREC corpus:

Extensive flooding was reported Sunday on the Chattahoochee River in Georgia as it neared its crest at Tailwater and George **Dam**, its **highest** level since 1929.

A swollen tributary the Ganges River in the capital today reached its **highest** level in 34 years, officials said, as soldiers and volunteers worked to build **dams** against the rising waters.

Two years ago, the numbers of steelhead returning to the river was the **highest** since the **dam** was built in 1959.

Knowledge Mining:

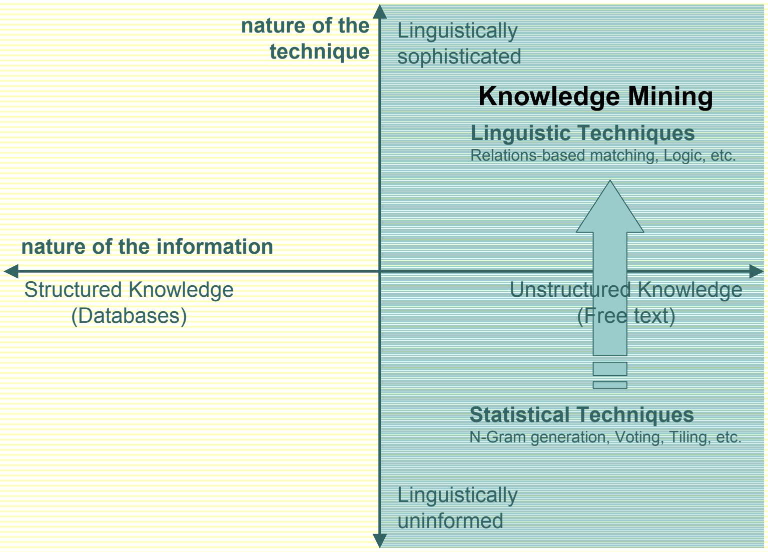
Conclusion

Question Answering Techniques for the World Wide Web

Summary

- The enormous amount of text available on the Web can be successfully utilized for QA
- Knowledge mining is a relatively new, but active field of research
- Significant progress has been made in the past few years
- Significant challenges have yet to be addressed
- Linguistically-sophisticated techniques promise to further boost knowledge mining performance

The Future



nature of the technique

Linguistically sophisticated

Knowledge Mining

Linguistic Techniques
Relations-based matching, Logic, etc.

nature of the information

Structured Knowledge (Databases)

Unstructured Knowledge (Free text)

Statistical Techniques
N-Gram generation, Voting, Tiling, etc.

Linguistically uninformed

Knowledge Annotation

Question Answering Techniques for the World Wide Web

Knowledge Annotation:
General Overview

Question Answering Techniques for the World Wide Web

Knowledge Annotation

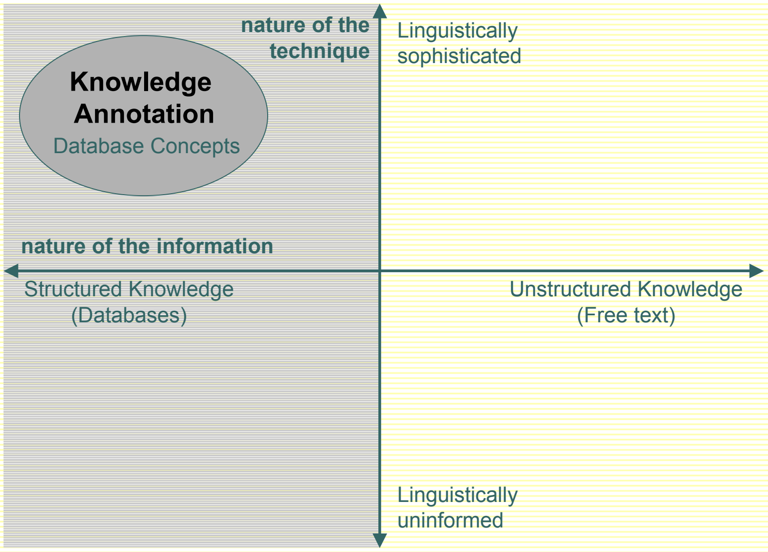
- **Definition:** techniques that effectively employ structured and semistructured sources on the Web for question answering
- **Key Ideas:**
  - “Wrap” Web resources for easy access
  - Employ annotations to connect Web resources to natural language
  - Leverage “Zipf’s Law of question answering”

Key Questions

- How can we organize diverse, heterogeneous, and semistructured sources on the Web?
- Is it possible to “consolidate” these diverse resources under a unified framework?
- Can we effectively integrate this knowledge into a question answering system?
- How can we ensure adequate knowledge coverage?

How can we effectively employ structured and semistructured sources on the Web for question answering?

Knowledge Annotation



nature of the technique

Linguistically sophisticated

Knowledge Annotation

Database Concepts

nature of the information

Structured Knowledge (Databases)

Unstructured Knowledge (Free text)

Linguistically uninformed

The Big Picture

- Start with structured or semistructured resources on the Web
- Organize them to provide convenient methods for access
- “Annotate” these resources with metadata that describes their information content
- Connect these annotated resources with natural language to provide question answering capabilities

Why Knowledge Annotation?

- The Web contains many databases that offer a wealth of information
- They are part of the “hidden” or “deep” Web
  - Information is accessible only through specific search interfaces
  - Pages are dynamically generated upon request
  - Content cannot be indexed by search engines
  - Knowledge mining techniques are not applicable
- With knowledge annotation, we can achieve high-precision question answering

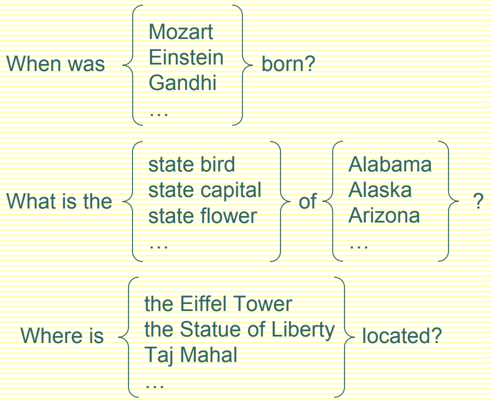
Sample Resources

- Internet Movie Database
  - Content: cast, crew, and other movie-related information
  - Size: hundreds of thousands of movies; tens of thousands of actors/actresses
- CIA World Factbook
  - Content: geographic, political, demographic, and economic information
  - Size: approximately two hundred countries/territories in the world
- Biography.com
  - Content: short biographies of famous people
  - Size: tens of thousands of entries

“Zipf’s Law of QA”

Observation: a few “question types” account for a large portion of all question instances

Similar questions can be parameterized and grouped into question classes, e.g.,



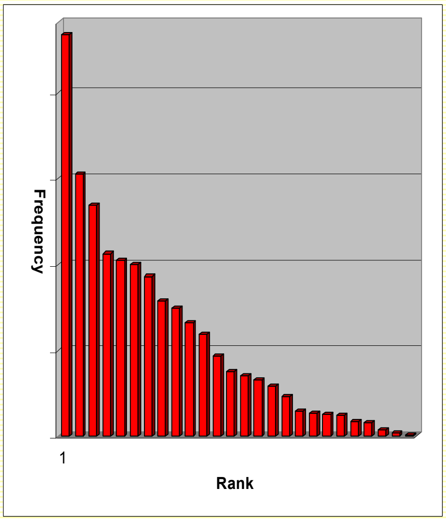
When was { Mozart / Einstein / Gandhi / … } born?

What is the { state bird / state capital / state flower / … } of { Alabama / Alaska / Arizona / … } ?

Where is { the Eiffel Tower / the Statue of Liberty / Taj Mahal / … } located?

Zipf's Law in Web Search [Lowe 2000]

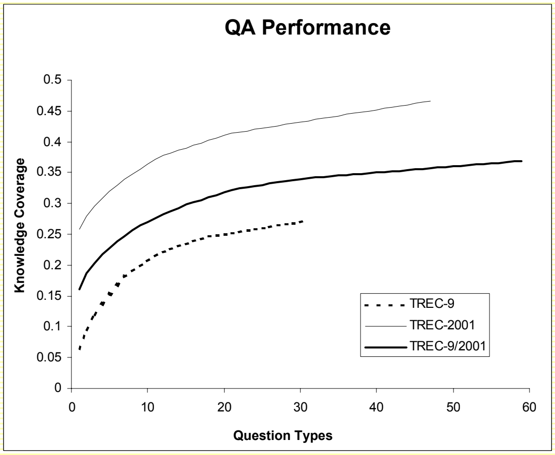
Frequency distribution of user queries from AskJeeves' search logs



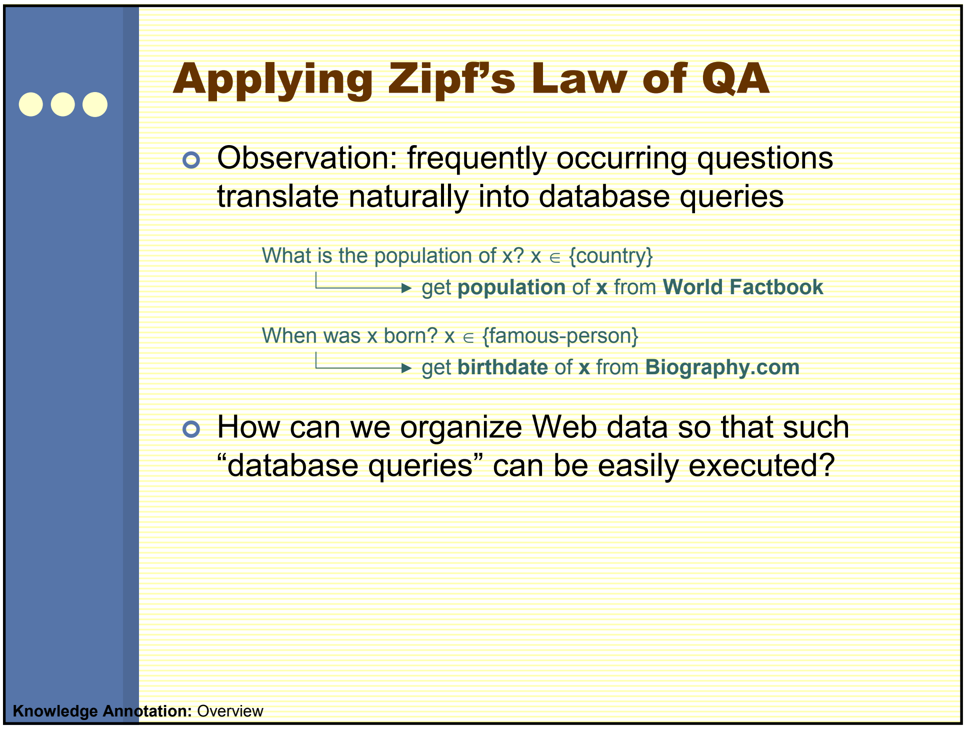
Frequently occurring questions dominate all questions

Zipf's Law in TREC [Lin, J. 2002]

Cumulative distribution of question types in the TREC test collections



Ten question types alone account for ~20% of questions from TREC-9 and ~35% of questions from TREC-2001



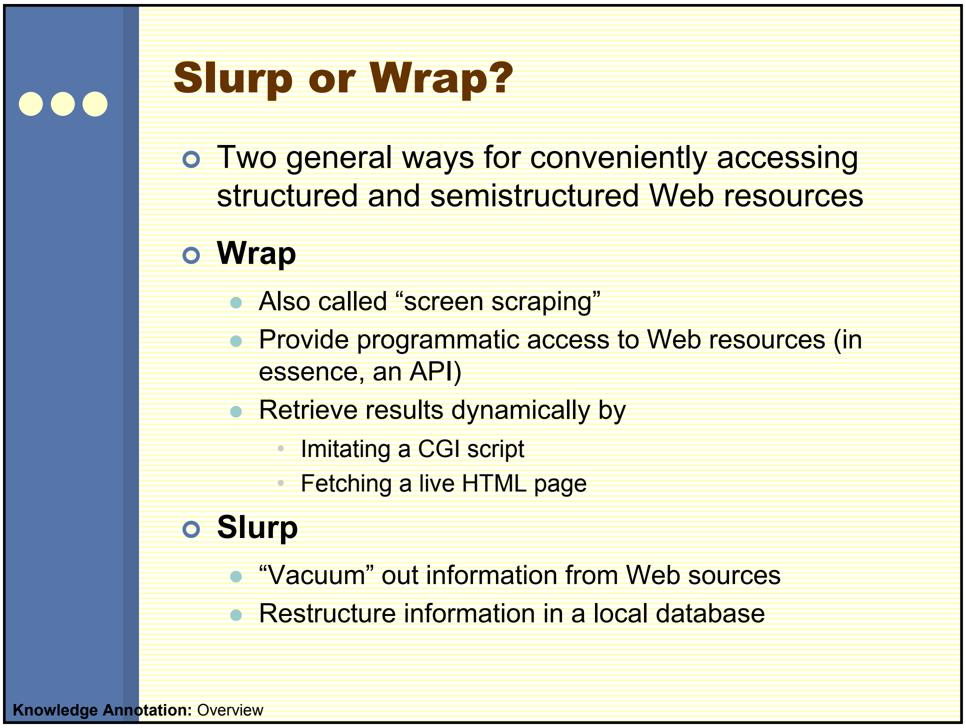
Applying Zipf's Law of QA

- Observation: frequently occurring questions translate naturally into database queries

  What is the population of x? x ∈ {country}
  → get **population** of **x** from **World Factbook**

  When was x born? x ∈ {famous-person}
  → get **birthdate** of **x** from **Biography.com**

- How can we organize Web data so that such “database queries” can be easily executed?



Slurp or Wrap?

- Two general ways for conveniently accessing structured and semistructured Web resources
- **Wrap**
  - Also called “screen scraping”
  - Provide programmatic access to Web resources (in essence, an API)
  - Retrieve results dynamically by
    - Imitating a CGI script
    - Fetching a live HTML page
- **Slurp**
  - “Vacuum” out information from Web sources
  - Restructure information in a local database

Tradeoffs: Wrapping

- **Advantages:**
  - Information is always up-to-date (even when the content of the original source changes)
  - Dynamic information (e.g., stock quotes and weather reports) is easy to access
- **Disadvantages:**
  - Queries are limited in expressiveness

    Queries limited by the CGI facilities offered by the website
    Aggregate operations (e.g., max) are often impractical
  - Reliability issues: what if source goes down?
  - Wrapper maintenance: what if source changes layout/format?

Tradeoffs: Slurping

- **Advantages:**
  - Queries can be arbitrarily expressive

    Allows retrieval of records based on different keys
    Aggregate operations (e.g., max) are easy
  - Information is always available (high reliability)
- **Disadvantages:**
  - Stale data problem: what if the original source changes or is updated?
  - Dynamic data problem: what if the information changes frequently? (e.g., stock quotes and weather reports)
  - Resource limitations: what if there is simply too much data to store locally?

Data Modeling Issues

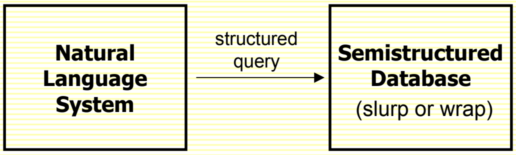
- How can we impose a data model on the Web?

  Two constraints:
  1. The data model must accurately capture both structure and content
  2. The data model must naturally mirror natural language questions

- Difficulties
  - Data is often inconsistent or incomplete
  - Data complexity varies from resource to resource

Putting it together

Connecting natural language questions to structured and semistructured data



What is the population of x? x ∈ {country}

→ get **population** of **x** from **CIA Factbook**

When was x born? x ∈ {famous-person}

→ get **birthdate** of **x** from **Biography.com**

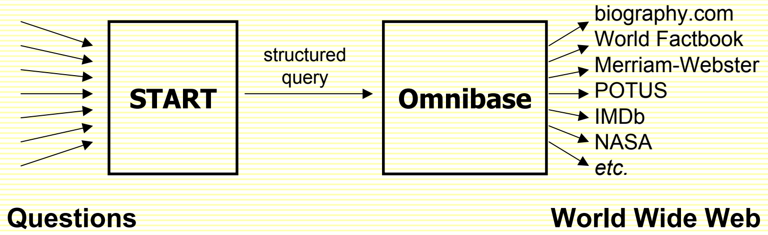
Knowledge Annotation:
START and Omnibase

Question Answering Techniques for the World Wide Web

START and Omnibase

[Katz 1988,1997; Katz *et al.* 2002a]

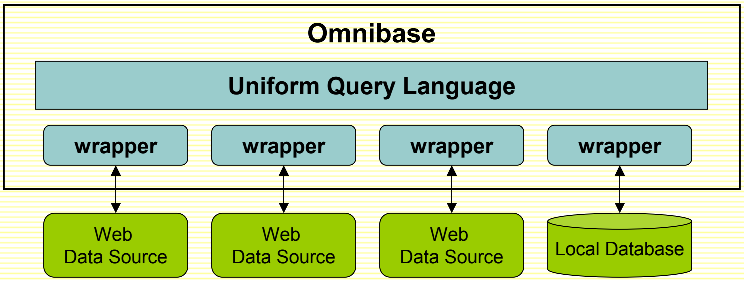
The first question answering system for the World Wide Web – employs knowledge annotation techniques



How does Omnibase work?
How does START work?
How is Omnibase connected to START?

Omnibase: Overview

- A “virtual” database that integrates structured and semistructured data sources
- An abstraction layer over heterogeneous sources



Omnibase

Uniform Query Language

wrapper | wrapper | wrapper | wrapper

Web Data Source | Web Data Source | Web Data Source | Local Database

Omnibase: OPV Model

- The Object-Property-Value (OPV) data model
  - Relational data model adopted for natural language
  - Simple, yet pervasive

Sources contain **objects**
Objects have **properties**
Properties have **values**

Many natural language questions can be analyzed as requests for the value of a property of an object

- The “get” command:

(get **source object property**) → **value**

Omnibase: OPV Examples

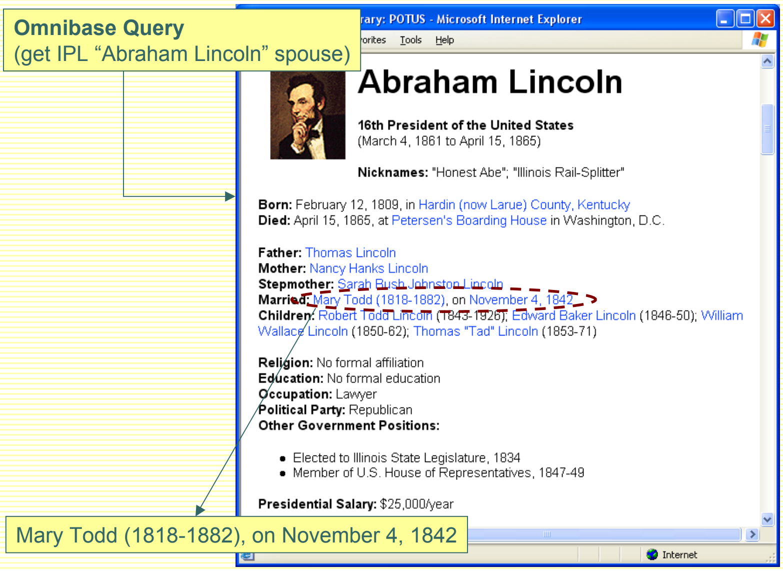
- “What is the population of Taiwan?”
  - **Source:** CIA World Factbook
  - **Object:** Taiwan
  - **Property:** Population
  - **Value:** 22 million
- “When was Andrew Johnson president?”
  - **Source:** Internet Public Library
  - **Object:** Andrew Johnson
  - **Property:** Presidential term
  - **Value:** April 15, 1865 to March 3, 1869

Omnibase: OPV Coverage

10 Web sources mapped into the Object-Property-Value data model cover 27% of the TREC-9 and 47% of the TREC-2001 QA Track questions

Question	Object	Property	Value
Who wrote the music for the Titanic?	Titanic	composer	John Williams
Who invented dynamite?	dynamite	inventor	Alfred Nobel
What languages are spoken in Guernsey?	Guernsey	languages	English, French
Show me paintings by Monet.	Monet	works	

Omnibase: Wrappers



Omnibase Query
(get IPL "Abraham Lincoln" spouse)

Abraham Lincoln

16th President of the United States
(March 4, 1861 to April 15, 1865)

Nicknames: "Honest Abe"; "Illinois Rail-Splitter"

Born: February 12, 1809, in Hardin (now Larue) County, Kentucky
Died: April 15, 1865, at Petersen's Boarding House in Washington, D.C.

Father: Thomas Lincoln
Mother: Nancy Hanks Lincoln
Stepmother: Sarah Bush Johnston Lincoln
Married: Mary Todd (1818-1882), on November 4, 1842
Children: Robert Todd Lincoln (1843-1926); Edward Baker Lincoln (1846-50); William Wallace Lincoln (1850-62); Thomas "Tad" Lincoln (1853-71)

Religion: No formal affiliation
Education: No formal education
Occupation: Lawyer
Political Party: Republican
Other Government Positions:

- Elected to Illinois State Legislature, 1834
- Member of U.S. House of Representatives, 1847-49

Presidential Salary: $25,000/year

Mary Todd (1818-1882), on November 4, 1842

Omnibase: Wrapper Operation

1. Generate URL
   - Map symbols onto URL

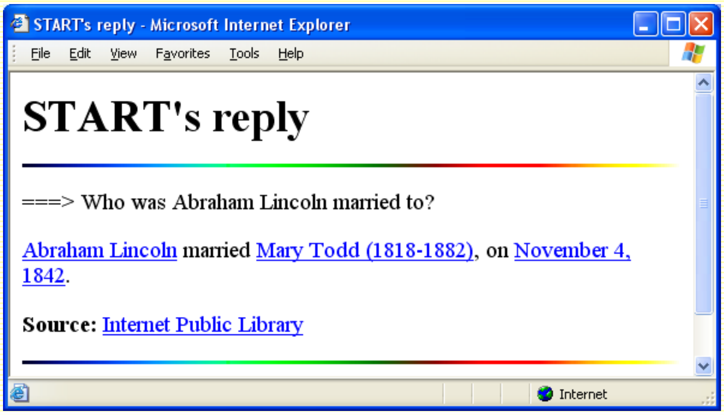
     Sometimes URLs can be computed directly from symbol
     Sometimes the mapping must be stored locally

     "Abraham Lincoln"
     "Abe Lincoln"　}　http://www.ipl.org/div/potus/alincoln.html
     "Lincoln"
2. Fetch Web page
3. Extract relevant information
   - Search for textual landmarks that delimit desired information (usually with regular expressions)

     <strong>Married: </strong>(.*)<br>

     Relevant information

Connecting the Pieces


START's reply - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Internet

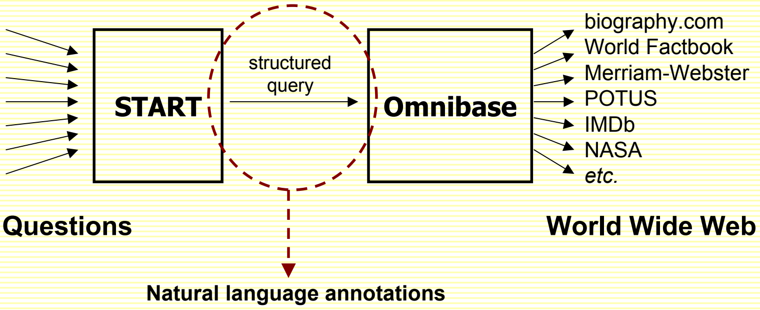
START's reply

===> Who was Abraham Lincoln married to?

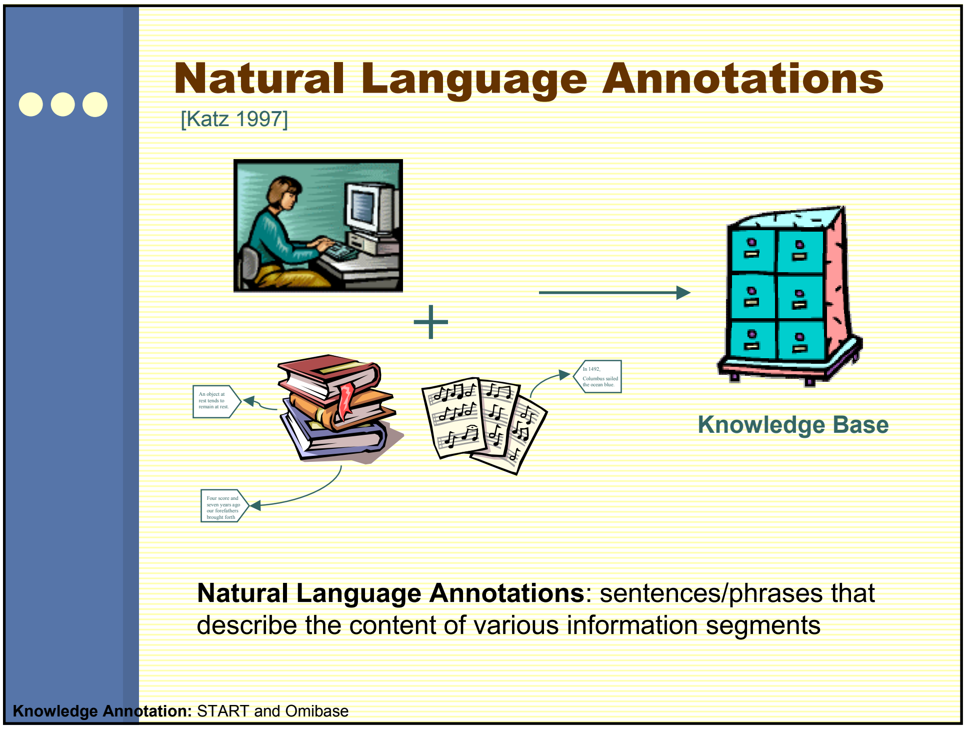
Abraham Lincoln married Mary Todd (1818-1882), on November 4, 1842.

Source: Internet Public Library

START and Omibase



- Natural language annotation technology connects START and Omnibase
- Detour into annotation-based question answering…

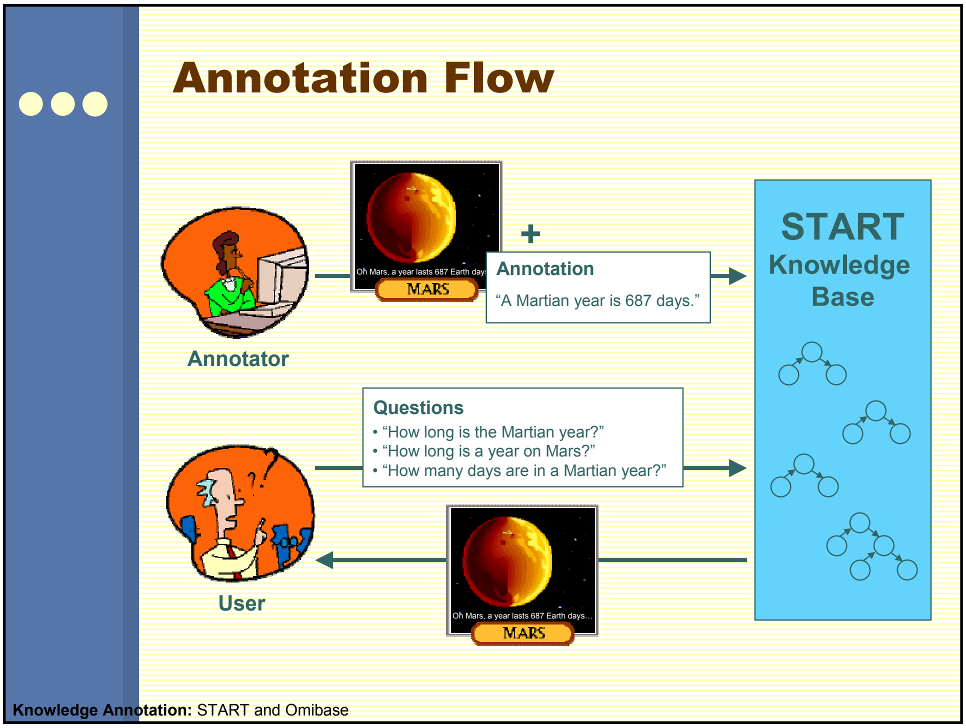


Natural Language Annotations

[Katz 1997]

An object at rest tends to remain at rest.
Four score and seven years ago our forefathers brought forth
In 1492, Columbus sailed the ocean blue.
Knowledge Base

Natural Language Annotations: sentences/phrases that describe the content of various information segments



Annotation Flow

Oh Mars, a year lasts 687 Earth day
MARS
+
Annotator
START Knowledge Base
User
Oh Mars, a year lasts 687 Earth days…
MARS

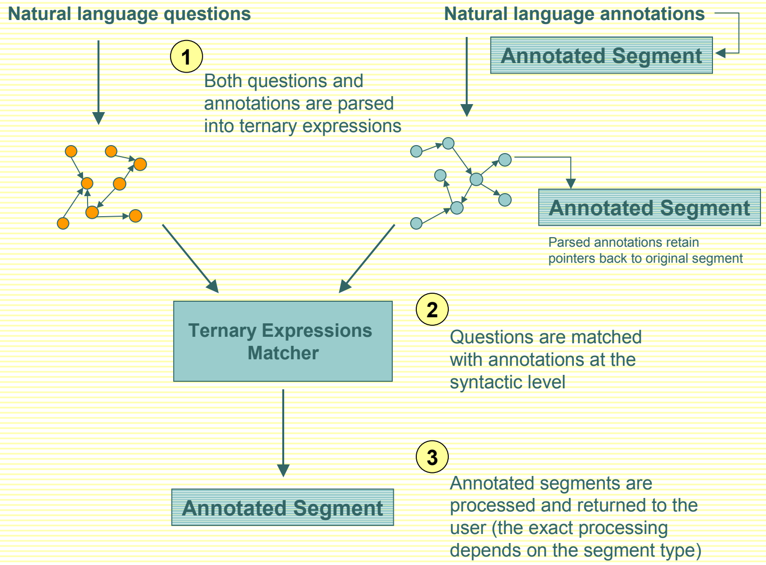
Annotation

"A Martian year is 687 days."

Questions

- "How long is the Martian year?"
- "How long is a year on Mars?"
- "How many days are in a Martian year?"

Matching Annotations



Natural language questions

Natural language annotations

Annotated Segment

1

Both questions and annotations are parsed into ternary expressions

Annotated Segment

Parsed annotations retain pointers back to original segment

Ternary Expressions Matcher

2

Questions are matched with annotations at the syntactic level

Annotated Segment

3

Annotated segments are processed and returned to the user (the exact processing depends on the segment type)

Syntactic Matching

- Allows utilization of linguistic techniques to aid in the matching process:
  - Synonyms
  - Hypernyms and hyponyms
  - Transformation rules to handle syntactic alternations

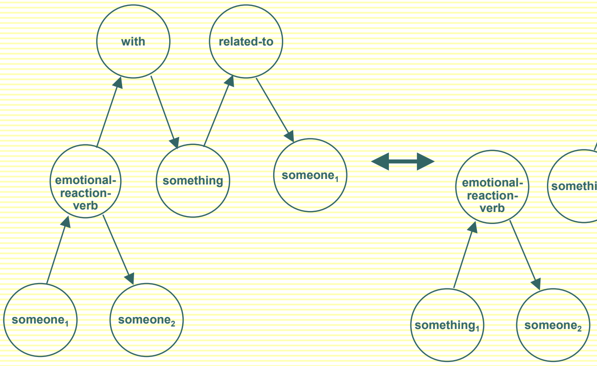
Transformation Rules

[Katz and Levin 1988]

The president impressed the country with his determination. ↔ The president's determination impressed the country.

S-rule for the Property Factoring alternation:

someone$_1$ emotional-reaction-verb someone$_2$ with something ↔ someone$_1$'s something emotional-reaction-verb someone$_2$



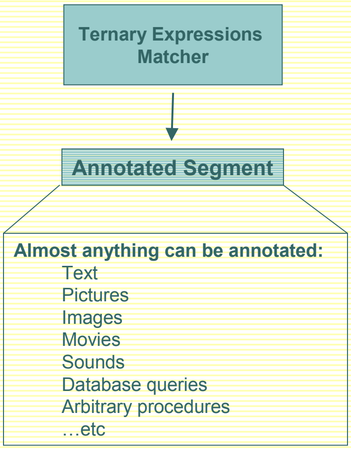
Emotional reaction verbs:

surprise stun
amaze startle
impress please
etc.

Matching and Retrieval

1. Both questions and annotations are parsed into ternary expressions
2. Questions are matched with annotations at the syntactic level
3. Annotated segments are processed and returned to the user

The action taken when an annotation matches a question depends on the type of annotated segment



Almost anything can be annotated:
Text
Pictures
Images
Movies
Sounds
Database queries
Arbitrary procedures
…etc

What Can We Annotate?

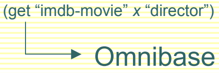
Direct Parseables

The annotated segment is the annotation itself. This allows us to assert facts and answer questions about them

Multimedia Content

Annotating pictures, sounds, images, etc. provides access to content we otherwise could not analyze directly

Structured Queries



(get "imdb-movie" *x* "director")

Omnibase

Annotating Omnibase queries provides START access to semistructured data

Arbitrary Procedures

get-time

λ

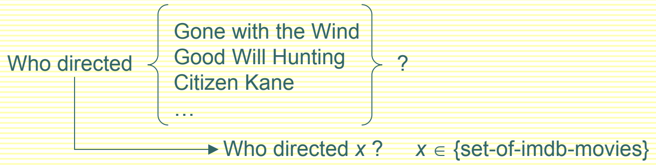
Annotating procedures (e.g., a system call to a clock) allows START to perform a computation in response to a question

Retrieving Knowledge

- Matching of natural language annotations triggers the retrieval process
- Retrieval process depends on the annotated segment:
  - Direct parseables – generate the sentence
  - Multimedia content – return the segment directly
  - Arbitrary procedures – execute the procedure
  - Database queries – execute the database query
- Annotations provide access to content that our systems otherwise could not analyze

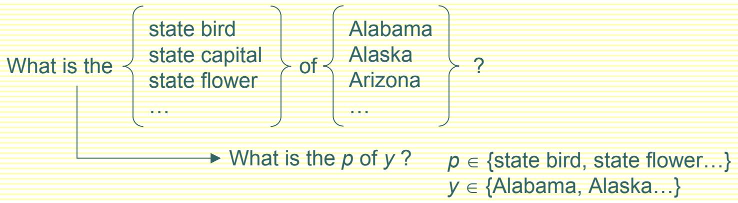
Parameterized Annotations

Natural language annotations can contain parameters that stand in for large classes of lexical entries



Who directed {Gone with the Wind, Good Will Hunting, Citizen Kane, …} ?

→ Who directed *x* ? $x \in$ {set-of-imdb-movies}



What is the {state bird, state capital, state flower, …} of {Alabama, Alaska, Arizona, …} ?

→ What is the *p* of *y* ? $p \in$ {state bird, state flower…}
$y \in$ {Alabama, Alaska…}

Natural language annotations can be sentences, phrases, or questions

Recognizing Objects

In order for parameterized annotations to match, objects have to be recognized

Extraction of objects makes parsing possible:

compare {Who directed smultronstallet? Who directed mfbflxt?}

Which one is gibberish?
Which one is a real question?

compare {Who directed gone with the wind? Who hopped flown past the street?}

Omnibase serves as a gazetteer for START (to recognize objects)

Who directed smultronstallet?
→ Who directed *x* ?
x = “Smultronstället (1957)” (“Wild Strawberries”) from imdb-movie

Who directed gone with the wind?
→ Who directed *x* ?
x = “Gone with the Wind (1939)” from imdb-movie

The Complete QA Process

- START, with the help of Omnibase, figures out which sources can answer the question
- START translates the question into a structured Omnibase query
- Omnibase executes the query by
  - Fetching the relevant pages
  - Extracting the relevant fragments
- START performs additional generation and returns the answer to the user

START: Performance

From January 2000 to December 2002, about a million questions were posed to START and Omnibase

	2000	2001	2002
Answer: Omnibase	**85k (27.1%)**	**100k (37.6%)**	**129k (37.9%)**
Answer: START native	**123k (39.3%)**	**74k (27.9%)**	**107k (31.5%)**
Don't know	72k (22.9%)	65k (24.3%)	78k (22.8%)
Don't understand	19k (6.0%)	15k (5.5%)	14k (4.2%)
Unknown word	15k (4.8%)	12k (4.7%)	12k (3.6%)
Total	313k (100%)	266k (100%)	342k (100%)

Don't know = question successfully parsed, but no knowledge available
Don't know = question couldn't be parsed

Of those, 619k questions were successfully answered

	2000	2001	2002
Total Answered Correctly	208k (66.4%)	174k (65.5%)	237k (69.4)
Answered using Omnibase	40.9%	57.4%	54.6%
Answer with native KB	59.1%	42.6%	45.4%

Knowledge Annotation:

Other Annotation-based Systems

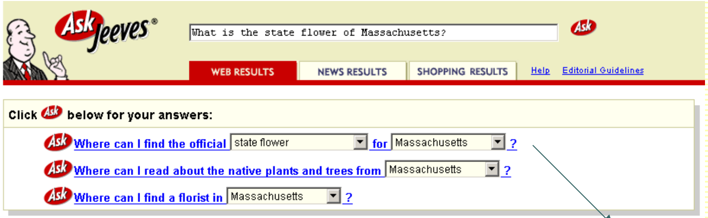
Question Answering Techniques for the World Wide Web

Annotation-Based Systems

- AskJeeves
- FAQ Finder (U. Chicago)
- Aranea (MIT)
- KSP (IBM)
- “Early Answering” (U. Waterloo)
- Annotation-based Image Retrieval

AskJeeves www.ask.com

- Lots of manually annotated URLs
- Includes keyword-based matching
- Licenses certain technologies pioneered by START



What is the state flower of Massachusetts?

Click below for your answers:

Where can I find the official [state flower] for [Massachusetts] ?

Where can I read about the native plants and trees from [Massachusetts] ?

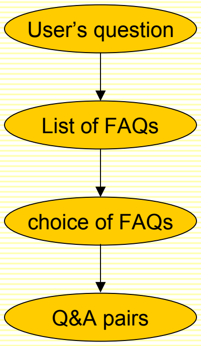
Where can I find a florist in [Massachusetts] ?

compare

What is the {state bird, state capital, state flower, …} of {Alabama, Alaska, Arizona, …} ?

FAQ Finder U. Chicago: [Burke *et al.* 1997]

Question answering using lists of frequently asked questions (FAQ) mined from the Web: the questions from FAQ lists can be viewed as annotations for the answers



Uses SMART [Salton 1971] to find potentially relevant lists of FAQ

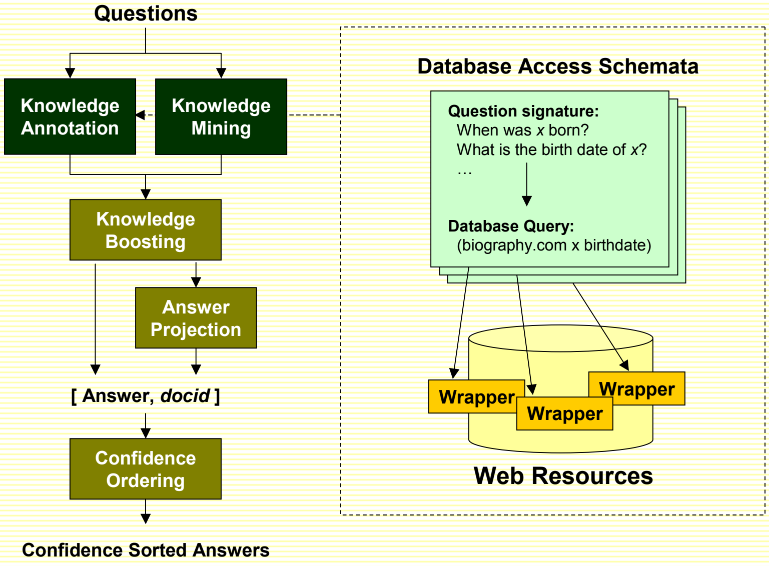
User manually chooses which FAQs to search

System matches user question with FAQ questions and returns Q&A pairs

Metrics of similarity
- **Statistical**: *tf.idf* scoring
- **Semantic:** takes into account the length of path between words in WordNet

Aranea

MIT: [Lin, J. *et al.* 2002]



Questions

Knowledge Annotation

Knowledge Mining

Knowledge Boosting

Answer Projection

[Answer, *docid*]

Confidence Ordering

Confidence Sorted Answers

Database Access Schemata

Question signature:
When was *x* born?
What is the birth date of *x*?
…

Database Query:
(biography.com x birthdate)

Wrapper

Wrapper

Wrapper

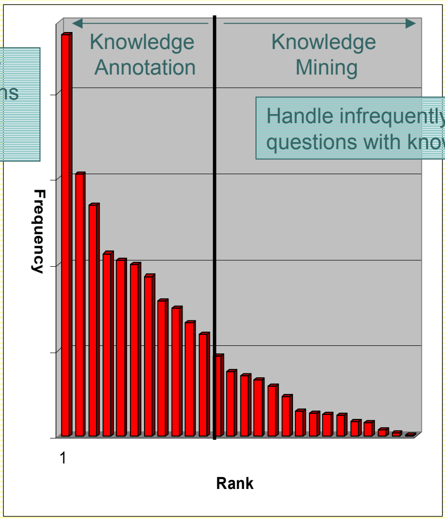
Web Resources

Aranea: Overview

- Database access schemata
  - Regular expressions connect question signatures to wrappers
  - If user question matches question signature, database query is executed (via wrappers)
- Overall performance: TREC 2002 (official)
  - Official score: 30.4% correct, CWS 0.433
  - Knowledge annotation component contributed 15% of the performance (with only six sources)
- Observations:
  - High precision, lower recall
  - Failure modes: question signature mismatch, wrapper malfunction

Aranea: Integration

Capitalize on the Zipf's Law of question distribution:



Handle frequently occurring questions with knowledge annotation

Handle infrequently occurring questions with knowledge mining

KSP

IBM: [Chu-Carroll *et al.* 2002]

- KSP = Knowledge Server Portal
  - A "structured knowledge agent" in a multi-agent QA architecture: IBM's entry to TREC 2002
  - Composed of a set of knowledge-source adaptors
  - Performance contribution is unclear
- Supports queries that the question analysis component is capable of recognizing, e.g.,
  - "What is the capital of Syria?"
  - "What is the state bird of Alaska?"
- Sample sources
  - US Geological Survey
  - www.uselessknowledge.com
  - WordNet

"Early Answering"

U. Waterloo: [Clarke *et al.* 2002]

Answer specific types of questions using a structured database gathered from Web sources

Sample Resources:

Table	# elements
Airports (code, name, location)	1,500
Rulers (location, period, title)	25,000
Acronyms	112,000
Colleges and Universities (name, location)	5,000
Holidays	171
Animal Names (baby, male, female, group)	500

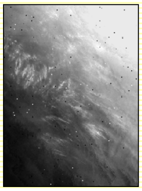
Performance: +10-14% in correct answers +16-24% CWS

Image Retrieval

- Annotation-based techniques are commonly used for image retrieval

  e.g., [Flank *et al.* 1995; Smeaton and Quigley 1996]

  - Image captions are natural sources of annotations



This *Viking 1* Orbiter image shows clouds to the north of Valles Marineris that look similar to cirrus clouds on Earth

Knowledge Annotation:

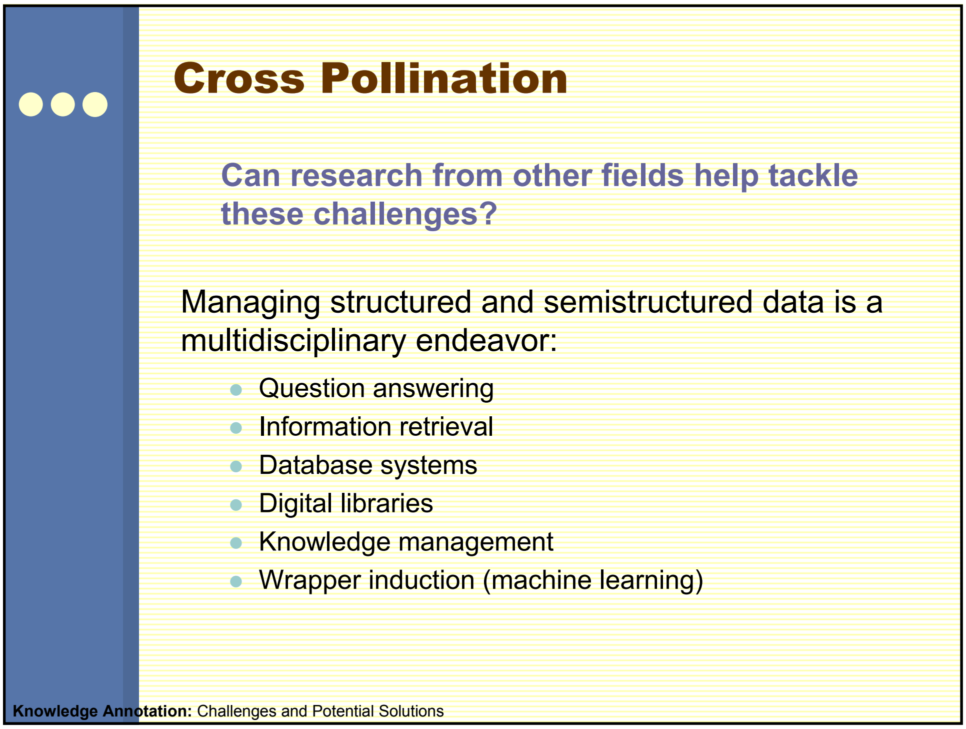
Challenges and Potential Solutions

Question Answering Techniques for the World Wide Web

Four Challenges

[Katz and Lin 2002b; Katz *et al.* 2002b]

- The Knowledge Integration Problem:
  - How can we integrate information from multiple sources?
- The Scaling Problem:
  - Annotations are simple and intuitive, but…
  - There is simply too much data to annotate
- The Knowledge Engineering Bottleneck:
  - Only trained individuals can write wrappers
  - “Knowledge engineers” are required to integrate new data sources
- The Fickle Web Problem:
  - Layout changes, content changes, and…
  - Our wrappers break

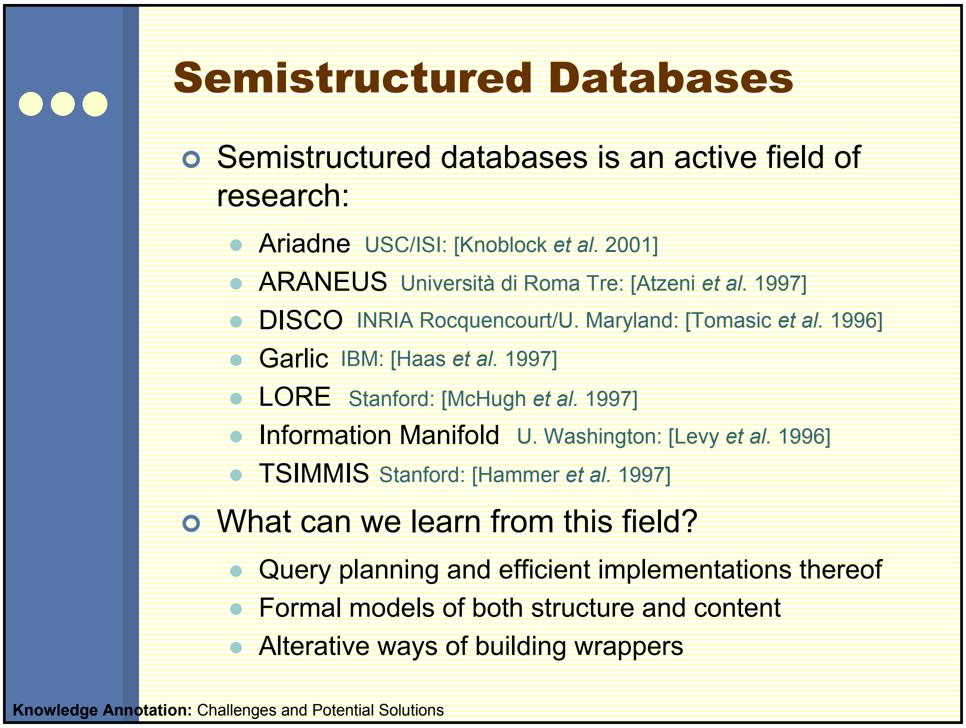


Cross Pollination

Can research from other fields help tackle these challenges?

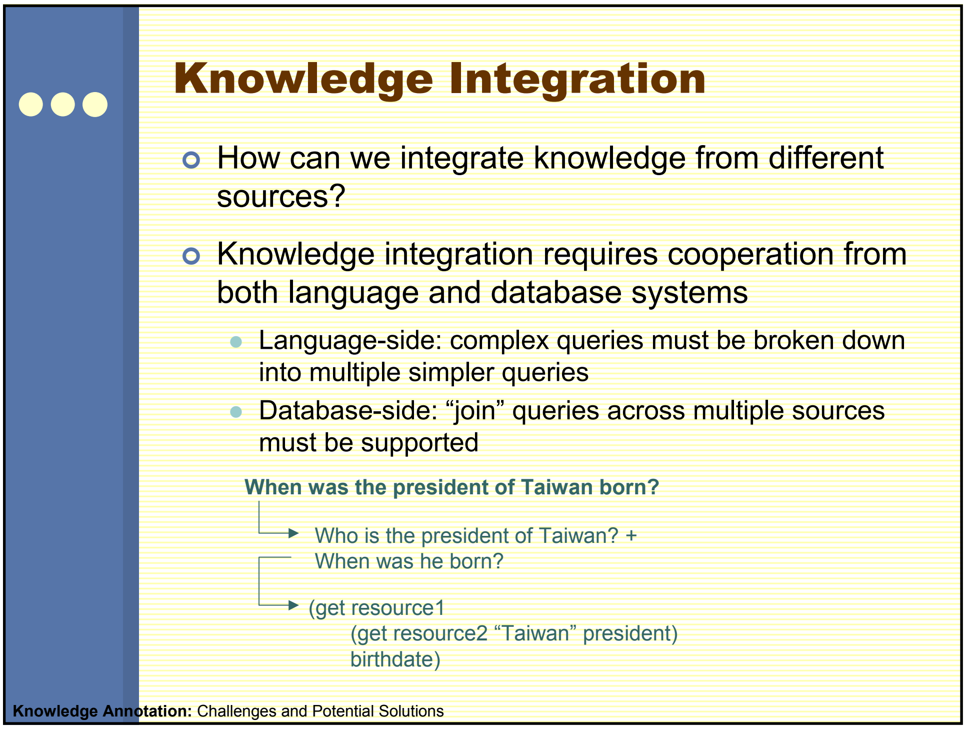
Managing structured and semistructured data is a multidisciplinary endeavor:

- Question answering
- Information retrieval
- Database systems
- Digital libraries
- Knowledge management
- Wrapper induction (machine learning)



Semistructured Databases

- Semistructured databases is an active field of research:
  - Ariadne USC/ISI: [Knoblock *et al.* 2001]
  - ARANEUS Università di Roma Tre: [Atzeni *et al.* 1997]
  - DISCO INRIA Rocquencourt/U. Maryland: [Tomasic *et al.* 1996]
  - Garlic IBM: [Haas *et al.* 1997]
  - LORE Stanford: [McHugh *et al.* 1997]
  - Information Manifold U. Washington: [Levy *et al.* 1996]
  - TSIMMIS Stanford: [Hammer *et al.* 1997]
- What can we learn from this field?
  - Query planning and efficient implementations thereof
  - Formal models of both structure and content
  - Alterative ways of building wrappers



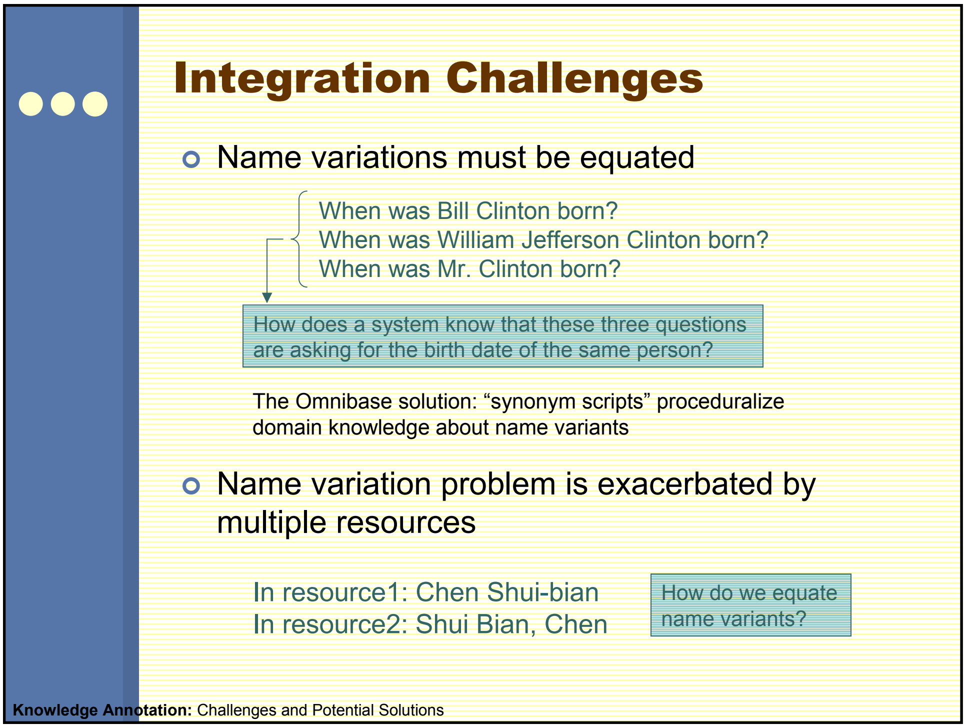
Knowledge Integration

- How can we integrate knowledge from different sources?
- Knowledge integration requires cooperation from both language and database systems
  - Language-side: complex queries must be broken down into multiple simpler queries
  - Database-side: “join” queries across multiple sources must be supported

When was the president of Taiwan born?

→ Who is the president of Taiwan? +
When was he born?

→ (get resource1
    (get resource2 “Taiwan” president)
    birthdate)



Integration Challenges

- Name variations must be equated

When was Bill Clinton born?
When was William Jefferson Clinton born?
When was Mr. Clinton born?

How does a system know that these three questions are asking for the birth date of the same person?

The Omnibase solution: “synonym scripts” proceduralize domain knowledge about name variants

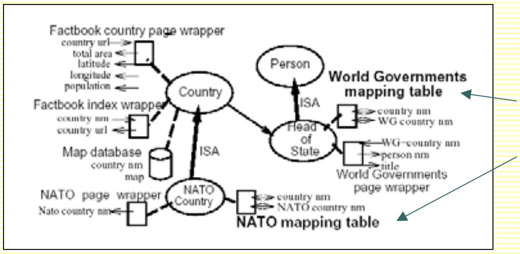
- Name variation problem is exacerbated by multiple resources

In resource1: Chen Shui-bian
In resource2: Shui Bian, Chen

How do we equate name variants?

Two Working Solutions

- Ariadne: manual "mapping tables" [Knoblock *et al.* 2001]



Manually specify mappings between object names from different sources

- WHIRL: "soft joins" [Cohen 2000]
  - Treat names as term vectors (with *tf.idf* weighting)
  - Calculate similarity score from the vectors:

$$Sim(\vec{u},\vec{v}) = \frac{\vec{u}\cdot\vec{v}}{\|\vec{u}\|\cdot\|\vec{v}\|}$$

Complex and Brittle Wrappers

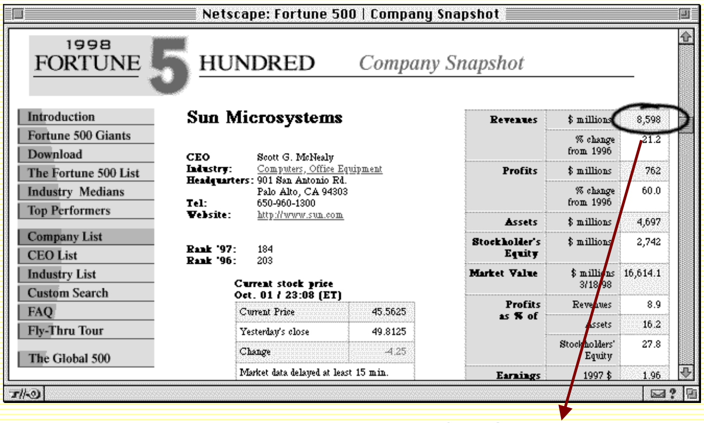
- Most wrappers are written in terms of textual "landmarks" found in a document, e.g.,
  - Category headings (such as "population:")
  - HTML tags (such as "<B>…</B>")
- Disadvantages of this approach:
  - Requires knowledge of the underlying encoding language (i.e., HTML), which is often very complex
  - Wrappers are brittle and may break with minor changes in page layout (tags change, different spacing, etc.)

LaMeTH

MIT: [Katz *et al.* 1999]

- “Semantic wrapper” approach: describe relevant information in terms of content elements, e.g.
  - Tables (e.g., 4th row, 3rd column)
  - Lists (e.g., 5th bulleted item)
  - Paragraphs (e.g., 2nd paragraph on the page)
- Advantages of this approach:
  - Wrappers become more intuitive and easier to write
  - Wrappers become more resistant to minor changes in page layout

LaMeTH: Example



Sun Microsystems

CEO Scott G. McNealy
Industry: Computers, Office Equipment
Headquarters: 901 San Antonio Rd.
Palo Alto, CA 94303
Tel: 650-960-1300
Website: http://www.sun.com

Rank '97: 184
Rank '96: 203

Current stock price
Oct. 01 / 23:08 (ET)

Current Price	45.5625
Yesterday's close	49.8125
Change	-4.25
Market data delayed at least 15 min.	

Revenues	$ millions	8,598
	% change from 1996	21.2
Profits	$ millions	762
	% change from 1996	60.0
Assets	$ millions	4,697
Stockholder's Equity	$ millions	2,742
Market Value	$ millions 3/18/98	16,614.1
Profits as % of	Revenues	8.9
	Assets	16.2
	Stockholders' Equity	27.8
Earnings	1997 $	1.96

(get-column 3 (get-row 1 (get-table 5 (get-profile “Sun Microsystems”))))

“Get the **3rd** column from the **1st** row of the **5th** table in Sun’s profile”

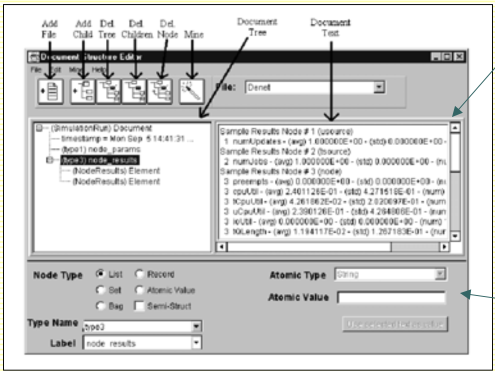
Write wrappers in terms of content blocks, not in terms of the underlying encoding

Simplifying Wrapper Creation

- Manual wrapper creation is time-consuming and laborious
- How can we simplify and speed up this process?
- Potential solutions:
  - GUI interfaces
  - Wrapper toolkits
  - Machine learning approaches

NoDoSE [Adelberg 1998; Adelberg and Denny 1999]

- NoDoSE = Northwestern Document Structure Extractor
- A GUI for hierarchically composing wrappers



File: Denell

```
(SimulationRun) Document
  timestamp = Mon Sep 5 14:41:31 ...
  (type1) node_params
  (type3) node_results
    (NodeResults) Element
    (NodeResults) Element
```

```
Sample Results Node # 1 (usource)
 1  numUpdates - (avg) 1.000000E+00 - (std) 0.000000E+00 -
Sample Results Node # 2 (tsource)
 2  numJobs - (avg) 1.000000E+00 - (std) 0.000000E+00 - (nu
Sample Results Node # 3 (node)
 3  preempts - (avg) 0.000000E+00 - (std) 0.000000E+00 - (nu
 3  cpuUtil - (avg) 2.401126E-01 - (std) 4.271518E-01 - (num)
 3  tCpuUtil - (avg) 4.261862E-02 - (std) 2.020097E-01 - (num
 3  uCpuUtil - (avg) 2.390126E-01 - (std) 4.264806E-01 - (num
 3  ioUtil - (avg) 0.000000E+00 - (std) 0.000000E+00 - (num)
 3  IOLength - (avg) 1.194117E-02 - (std) 1.267183E-01 - (nur
```

Node Type: ● List ○ Record ○ Set ○ Atomic Value ○ Bag □ Semi-Struct

Type Name: type3

Label: node_results

Atomic Type: String

Atomic Value:

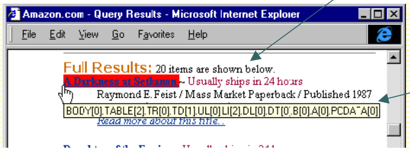
Wrappers are specified in terms of textual markers and offsets

Includes analyzer to detect non-functional scripts

W4F [Sahuguet and Azavant 1999]

- W4F = WysiWyg Web Wrapper Factory
- A wrapper construction GUI with point-and-click functionality

HTML document is analyzed as a tree



Full Results: 20 items are shown below.
A Darkness at Sethanon ~ Usually ships in 24 hours
Raymond E. Feist / Mass Market Paperback / Published 1987
BODY[0].TABLE[2].TR[0].TD[1].UL[0].LI[2].DL[0].DT[0].B[0].A[0].PCDATA[0]
Read more about this title...

Pointing at an element automatically calculates its “extraction path” – an Xpath-like expression

```
EXTRACTION_RULES ::
books = html.body.table[2].tr[0].td[1].ul[0].li[2].dl[0].dt[*]
 ( .b[0].a[0].pcdata[0].txt                                  // title
#  .b[0].a[0].getAttr(href)                                  // url
# ->dd[0].pcdata[0].txt, match /Published (19[0-9]{2})/      // year
# ->dd[0].pcdata[0].txt, match /(.*?)\//, split /, /         // authors
# ->dd[0].pcdata[1].txt, match /(\$[^ ]+)/                   // price
  );
```

Complex elements in a schema (e.g., regular expressions) must be specified manually

Wrapper Toolkits

- ISI’s Wrapper Toolkit [Ashish and Knoblock 1997]
  - System guesses Web page structure; user manually corrects computer mistakes
  - Extraction parser is generated using LEX and YACC
- UMD’s Wrapper Toolkit [Gruser *et al.* 1998]
  - User must manually specify output schema, input attributes, and input-output relations
  - Simple extractors analyze HTML as a tree and extract specific nodes
- AutoWrapper [Gao and Sterling 1999]
  - Wrappers are generated automatically using similarity heuristics
  - Approach works only on pages with repeated structure, e.g., tables
  - System does not allow human intervention

Wrapper Induction

- Apply machine learning algorithms to generate wrappers automatically
- From a set of labeled training examples, induce a wrapper that
  - Parses new sample documents
  - Extracts the relevant information

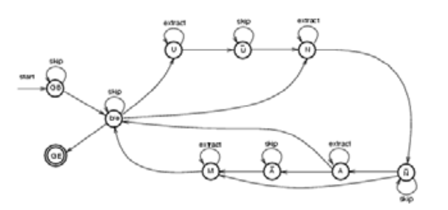
For Example:

Restaurants Review Site →
{ (name$_1$, location$_1$, cuisine-type$_1$, rating$_1$, …),
(name$_2$, location$_2$, cuisine-type$_2$, rating$_2$, …),
…
}

- Output of a wrapper is generally a set of tuples

Finite State Wrapper Induction

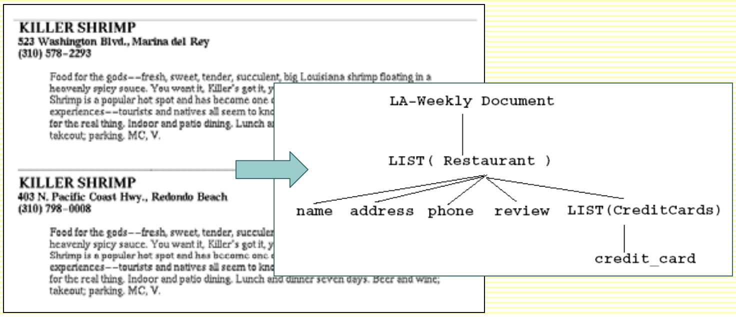
- HLRT Approach [Kushmerick *et al.* 1997; Kushmerick 1997]
  - Finds Head-Left-Right-Tail delimiters from examples and induces a restricted class of finite-state automata
  - Works only on tabular content layout
- SoftMealy [Hsu 1998; Hsu and Chang 1999]
  - Induces finite-state transducers from examples; single-pass or multi-pass (hierarchical) variants
  - Works on tabular documents and tagged-list documents
  - Requires very few training examples



Hierarchical Wrapper Induction

STALKER [Muslea *et al.* 1999]

EC (Embedded catalog) formalism: Web documents are analyzed as trees where non-terminal nodes are lists of tuples



KILLER SHRIMP
523 Washington Blvd., Marina del Rey
(310) 578-2293

Food for the gods--fresh, sweet, tender, succulent, big Louisiana shrimp floating in a heavenly spicy souce. You want it, Killer's got it, y
Shrimp is a popular hot spot and has become one c
experiences--tourists and natives all seem to kno
for the real thing. Indoor and patio dining. Lunch a
takeout; parking. MC, V.

KILLER SHRIMP
403 N. Pacific Coast Hwy., Redondo Beach
(310) 798-0008

Food for the gods--fresh, sweet, tender, succulen
heavenly spicy sauce. You want it, Killer's got it, y
Shrimp is a popular hot spot and has become one c
experiences--tourists and natives all seem to kno
for the real thing. Indoor and patio dining. Lunch and dinner seven days. Beer and wine;
takeout; parking. MC, V.

Extraction rules are attached to edges
List iteration rules are attached to list nodes
Rules implemented as finite state automata

Example:
R1 = SkipTo(</b>)
"ignore everything until a </b> marker"

Wrapper Induction: Issues

- Machine learning approaches require labeled training examples
  - Labeled examples are not reusable in other domains and for other applications
  - What is the time/effort tradeoff between labeling training examples and writing wrappers manually?
- Automatically induced wrappers are more suited for "slurping"
  - Wrapper induction is similar in spirit to information extraction: both are forms of template filling
  - All relations are extracted from a page at the same time
  - Less concerned with support services, e.g., dynamically generating URLs and fetching documents

Discovering Structure

- The Web contains mostly unstructured documents
- Can we organize unstructured sources for use by knowledge annotation techniques?
- Working solutions: automatically discover structured data from free text
  - DIPRE
  - Snowball
  - WebKB

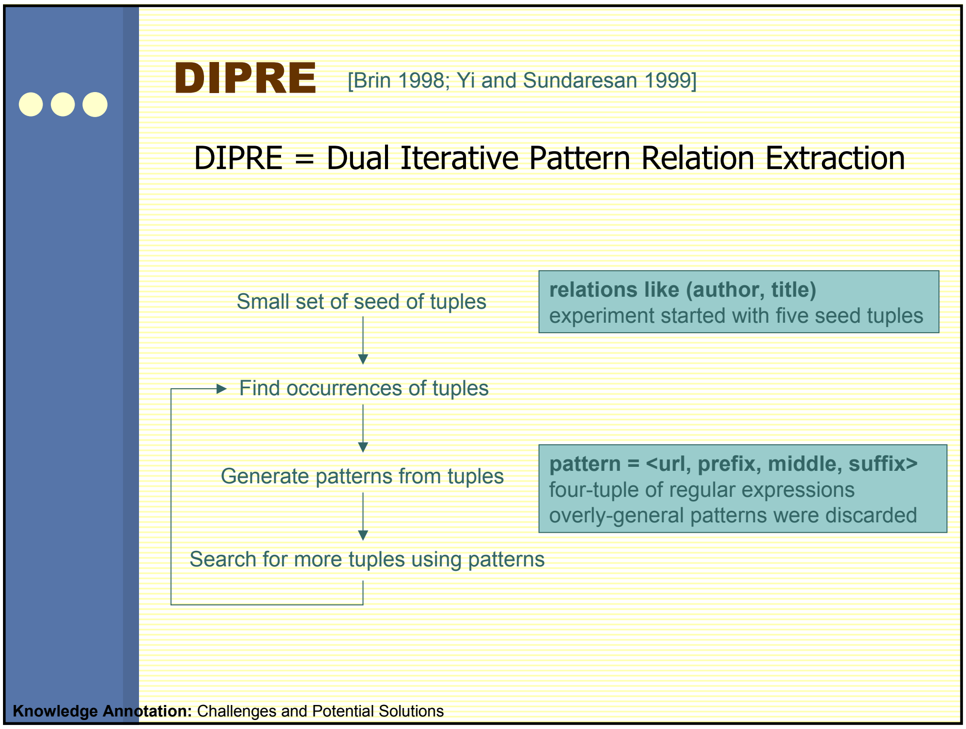
Extract Relations from Patterns

- Duality of patterns and relations
  - Relations can be gathered by applying surface patterns over large amounts of text

    For example, the relation between NAME and BIRTHDATE can be used for question answering
  - Surface patterns can be induced from sample relations by searching through large amounts of text

    For example, starting with the relation “Albert Einstein” and “1879”, a system can induce the pattern “was born in”
- What if…

  relations → patterns → more relations →
  more patterns → more relations …



DIPRE [Brin 1998; Yi and Sundaresan 1999]

DIPRE = Dual Iterative Pattern Relation Extraction

Small set of seed of tuples

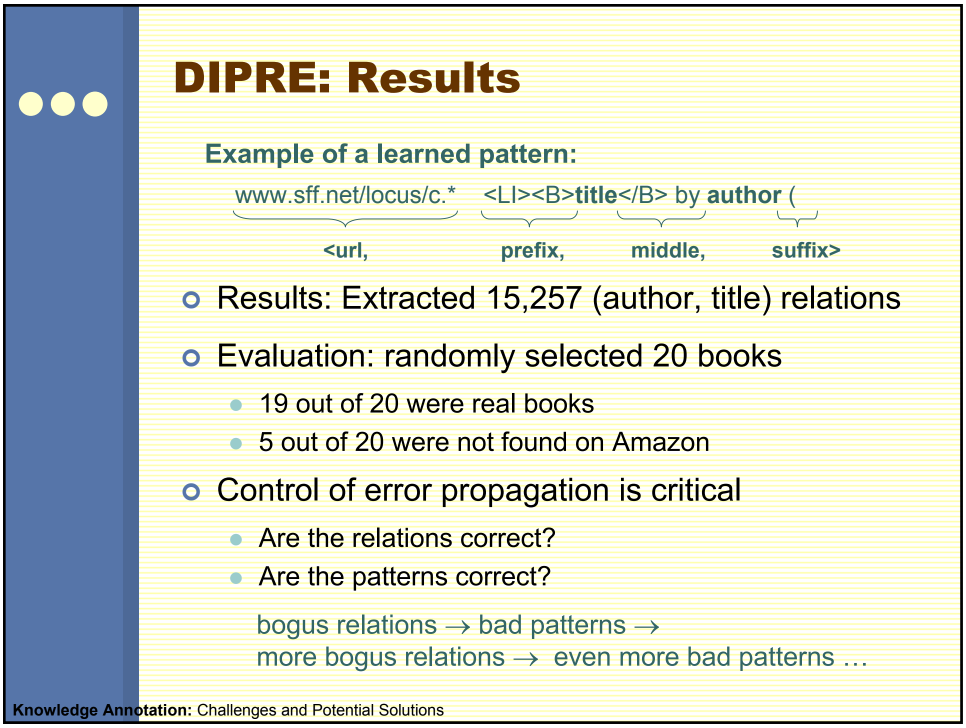
Find occurrences of tuples

Generate patterns from tuples

Search for more tuples using patterns

relations like (author, title)
experiment started with five seed tuples

pattern = <url, prefix, middle, suffix>
four-tuple of regular expressions
overly-general patterns were discarded



DIPRE: Results

Example of a learned pattern:

www.sff.net/locus/c.* <LI><B>**title**</B> by **author** (

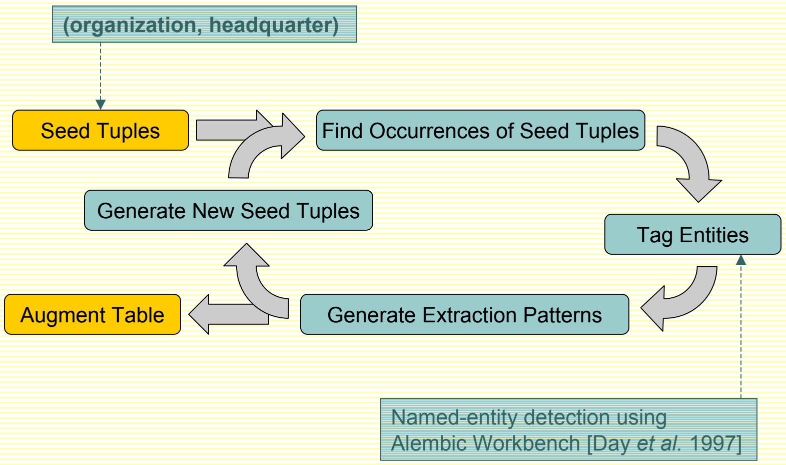
<url, prefix, middle, suffix>

- Results: Extracted 15,257 (author, title) relations
- Evaluation: randomly selected 20 books
  - 19 out of 20 were real books
  - 5 out of 20 were not found on Amazon
- Control of error propagation is critical
  - Are the relations correct?
  - Are the patterns correct?

bogus relations → bad patterns →
more bogus relations → even more bad patterns …

Snowball [Agichtein *et al.* 2000]

Snowball: several enhancements over DIPRE


(organization, headquarter)
Seed Tuples
Find Occurrences of Seed Tuples
Tag Entities
Generate Extraction Patterns
Generate New Seed Tuples
Augment Table
Named-entity detection using
Alembic Workbench [Day et al. 1997]

Snowball: Features

- Pattern: <left, tag1, mid, tag2, right>
  - **left**, **mid**, and **right** are vectors of term weights

Example Pattern:

<{<'the', 0.2>}, LOCATION, {<'-', 0.5>, <'based', 0.5>}, ORGANIZATION, {}>
left **tag1** **mid** **tag2** **right**

Example Text:

the Irving-based Exxon Corporation → (Exxon, Irving)

Matching Patterns with Text: take sum of dot products between term vectors

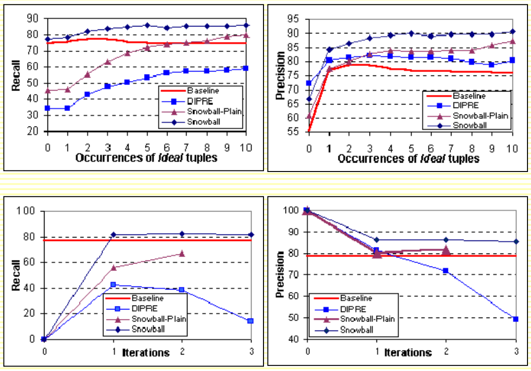
$$\text{Match}(t_p, t_s) = \begin{cases} l_p \cdot l_s + m_p \cdot m_s + r_p \cdot r_s & \text{if tags match} \\ 0 & \text{otherwise} \end{cases}$$

- Pattern learning: using tuples, find all pattern occurrences; cluster left, mid, and right vectors

Snowball: Features

- Confidence of a pattern is affected by
  - Accuracy of a pattern
  - Number of relations it generates
- Confidence of a tuple is affected by
  - Confidence of the patterns that generated it
  - Degree of match between relations and patterns
- “Learning rate” is used to control increase in pattern confidence
  - Dampening effect: system trusts new patterns less on each iteration

Snowball: Results



The more often a tuple occurs, the more likely it will be extracted

DIPRE has a tendency to “blow up” as irrelevant results are accumulated during each iteration. Snowball achieves both higher precision and recall

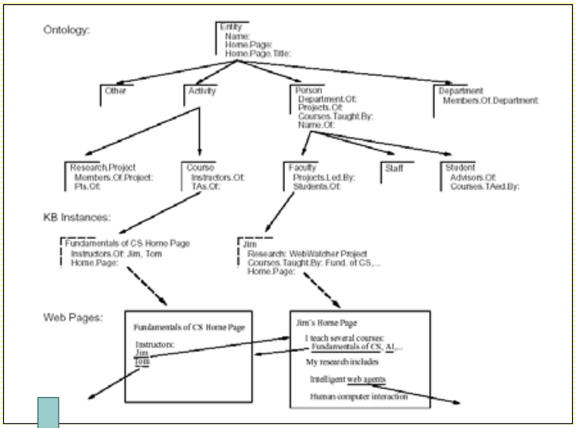
Snowball: punctuation used
Snowball-plain: punctuation ignored
DIPRE: from [Brin 1998]
Baseline: frequency of co-occurrence

Ground Truth = 13k organizations from Hoover’s Online crossed with extracted relations from Snowball

WebKB [Craven *et al.* 1998ab]

- Input:
  - Ontology that specifies classes and relations
  - Training examples that represent instances of relevant classes and relations
- Output:
  - A set of general procedures for extracting new instances of classes and relations

WebKB: Overview



Automatically learns extraction rules such as:

members-of-project(A,B) :- research_project(A), person(B), link_to(C,A,D), link_to(E,D,B), neighborhood_word_people(C).

Translation: Person B is a member of project A if there is a link from B to A near the keyword "people"

WebKB: Machine Learning

- Learns extraction rules using FOIL

  FOIL = a greedy covering algorithm for learning function free Horn clauses [Quinlan and Cameron-Jones 1993]

- Background relations used as “features”, e.g.,
  - has_*word*: boolean predicate that indicates the presence of a word on a page
  - link_to: represents a hyperlink between two pages
  - length: the length of a particular field
  - position: the position of a particular field
- Experimental results
  - Extracting relations from a CS department Web site (e.g., student, faculty, project, course)
  - Typical performance: 70-80% accuracy

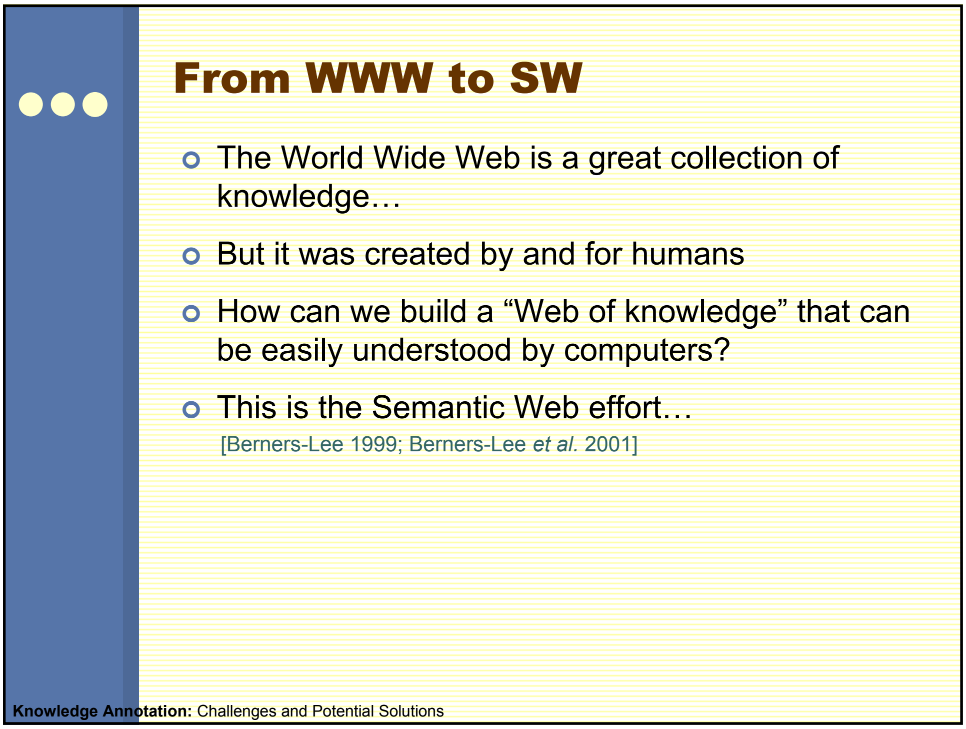
Extracting Relations: Issues

- How useful are these techniques?
- Can we extract relations that we don’t already have lists for?

  {author, title}: Amazon.com or the Library of Congress already possess comprehensive book catalogs

  {organization, headquarter}: Sites like Yahoo! Finance contains such information in a convenient form

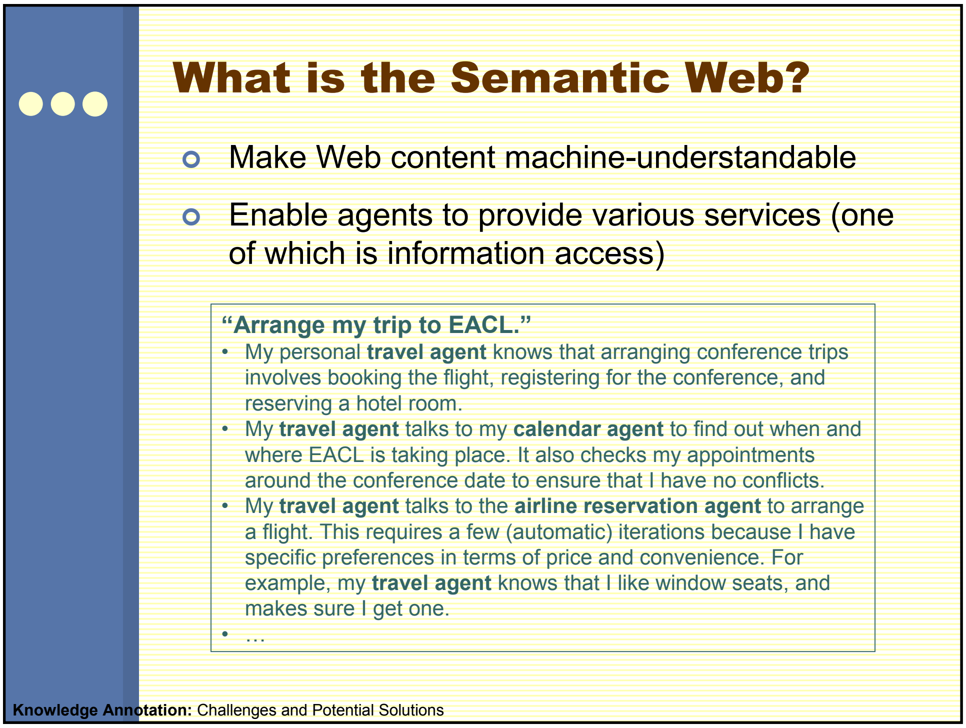
- Can we extract relations that have hierarchical structure? It is an open research question



From WWW to SW

- The World Wide Web is a great collection of knowledge…
- But it was created by and for humans
- How can we build a "Web of knowledge" that can be easily understood by computers?
- This is the Semantic Web effort…

  [Berners-Lee 1999; Berners-Lee *et al.* 2001]

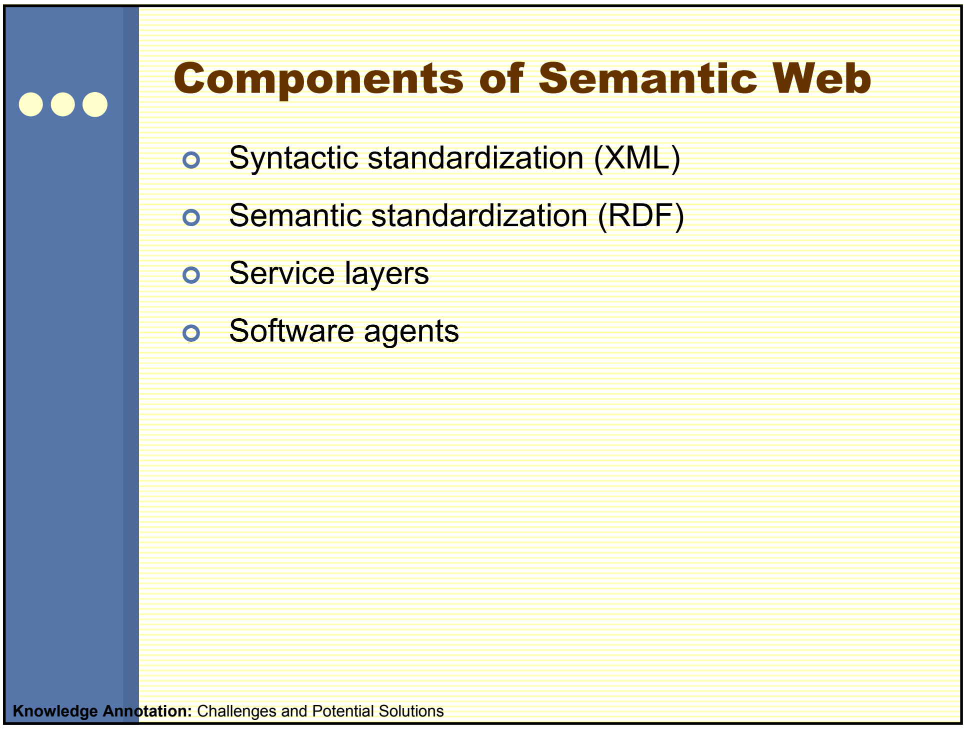


What is the Semantic Web?

- Make Web content machine-understandable
- Enable agents to provide various services (one of which is information access)

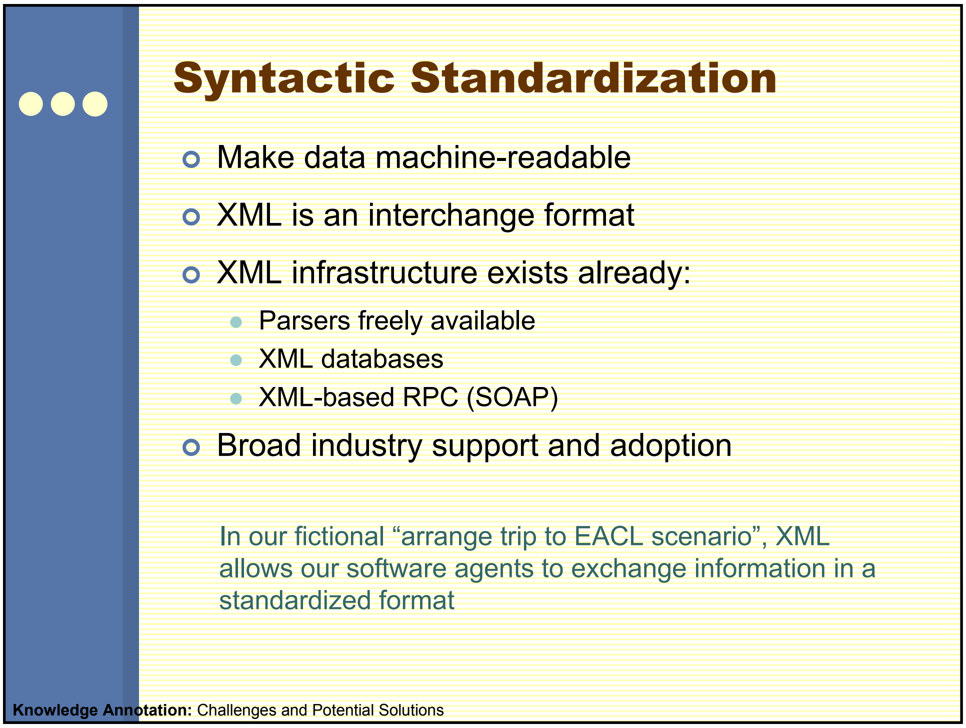
"Arrange my trip to EACL."

- My personal **travel agent** knows that arranging conference trips involves booking the flight, registering for the conference, and reserving a hotel room.
- My **travel agent** talks to my **calendar agent** to find out when and where EACL is taking place. It also checks my appointments around the conference date to ensure that I have no conflicts.
- My **travel agent** talks to the **airline reservation agent** to arrange a flight. This requires a few (automatic) iterations because I have specific preferences in terms of price and convenience. For example, my **travel agent** knows that I like window seats, and makes sure I get one.
- …



Components of Semantic Web

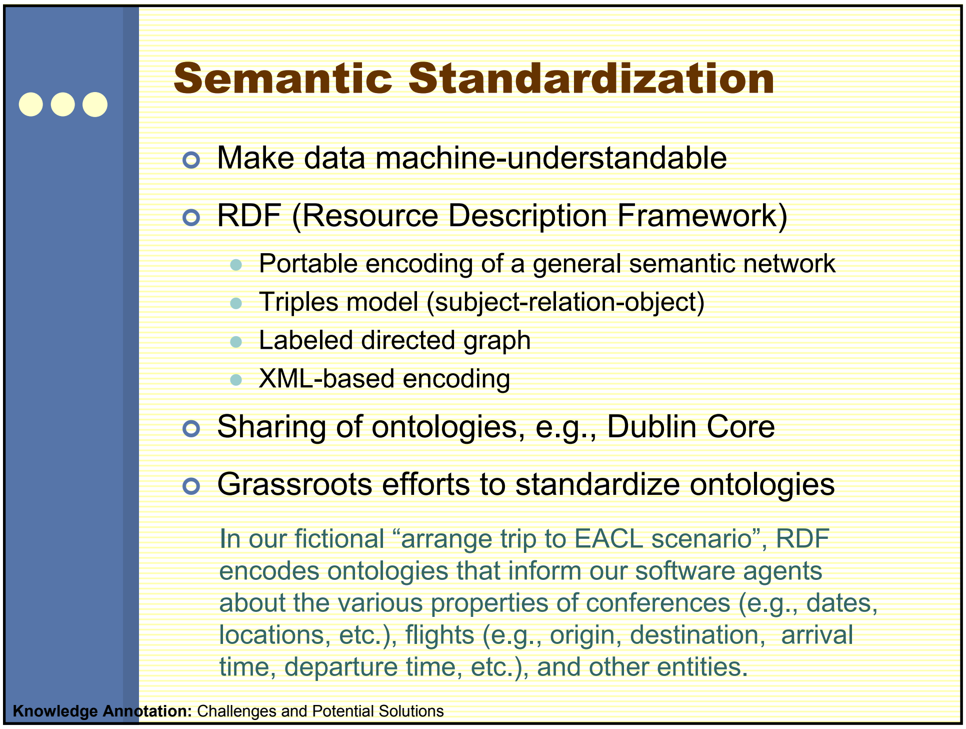
- Syntactic standardization (XML)
- Semantic standardization (RDF)
- Service layers
- Software agents



Syntactic Standardization

- Make data machine-readable
- XML is an interchange format
- XML infrastructure exists already:
  - Parsers freely available
  - XML databases
  - XML-based RPC (SOAP)
- Broad industry support and adoption

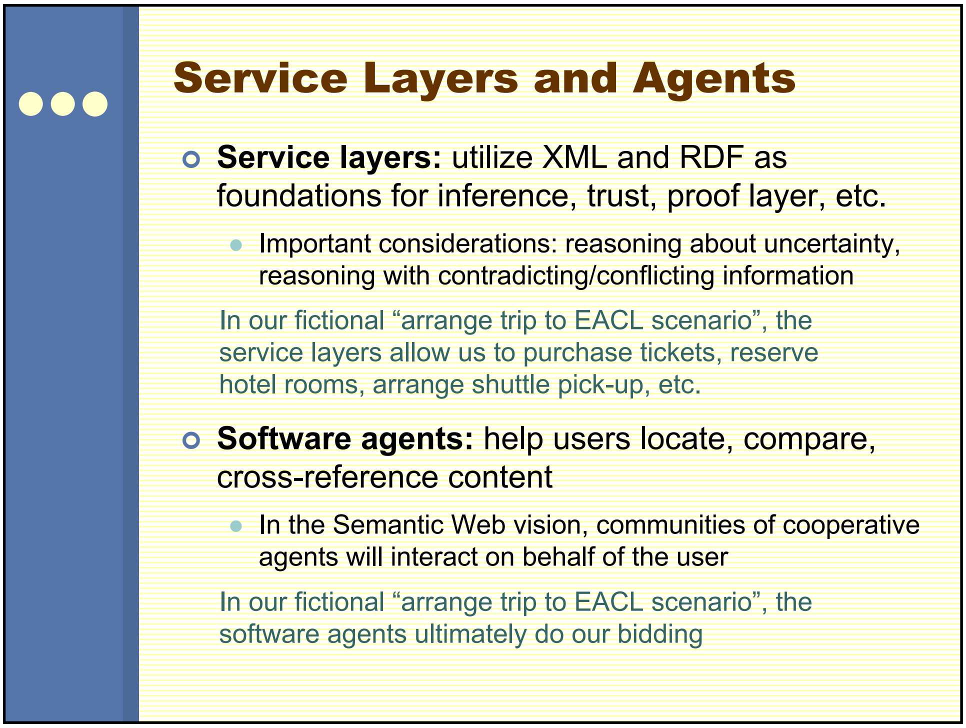
In our fictional “arrange trip to EACL scenario”, XML allows our software agents to exchange information in a standardized format



Semantic Standardization

- Make data machine-understandable
- RDF (Resource Description Framework)
  - Portable encoding of a general semantic network
  - Triples model (subject-relation-object)
  - Labeled directed graph
  - XML-based encoding
- Sharing of ontologies, e.g., Dublin Core
- Grassroots efforts to standardize ontologies

In our fictional “arrange trip to EACL scenario”, RDF encodes ontologies that inform our software agents about the various properties of conferences (e.g., dates, locations, etc.), flights (e.g., origin, destination, arrival time, departure time, etc.), and other entities.



Service Layers and Agents

- **Service layers:** utilize XML and RDF as foundations for inference, trust, proof layer, etc.
  - Important considerations: reasoning about uncertainty, reasoning with contradicting/conflicting information

In our fictional “arrange trip to EACL scenario”, the service layers allow us to purchase tickets, reserve hotel rooms, arrange shuttle pick-up, etc.

- **Software agents:** help users locate, compare, cross-reference content
  - In the Semantic Web vision, communities of cooperative agents will interact on behalf of the user

In our fictional “arrange trip to EACL scenario”, the software agents ultimately do our bidding

Semantic Web: What's Missing?

- Where in the loop is the human?
- How will we communicate with our software agents?
- How will we access information on the Semantic Web?

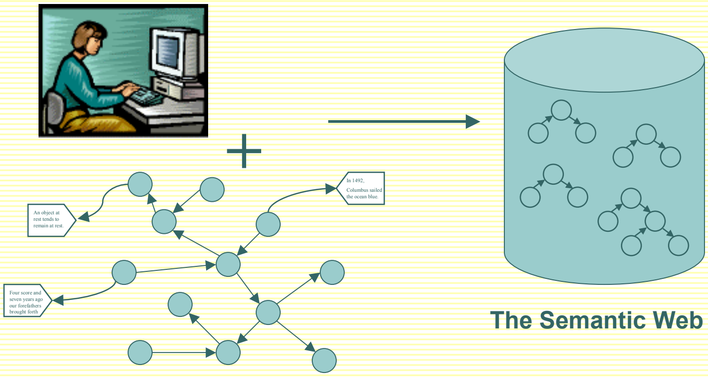
Obviously, we cannot expect ordinary Semantic Web users to manually manipulate ontologies, query with formal logic expressions, etc.

We would like to communicate with software agents in natural language…

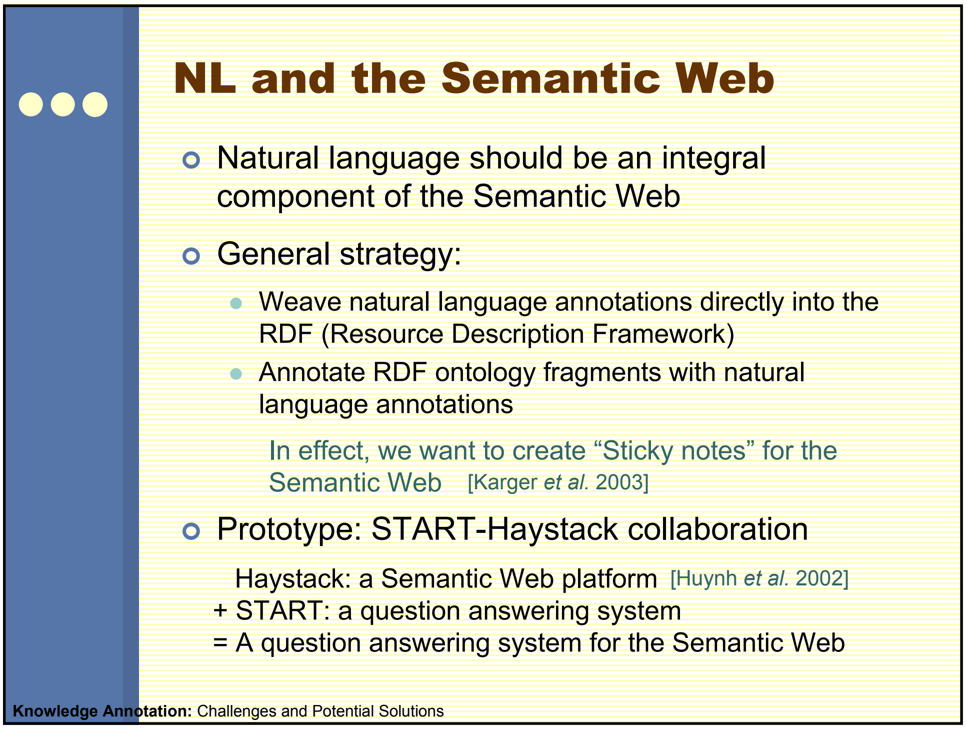
What is the role of natural language in the Semantic Web?

RDF + NL Annotations

[Katz and Lin 2002a; Katz *et al.* 2002c; Karger *et al.* 2003]


+
An object at rest tends to remain at rest.
Four score and seven years ago our forefathers brought forth
In 1492, Columbus sailed the ocean blue.
The Semantic Web

Annotate RDF as if it were any other type of content segment, i.e., describe RDF fragments with natural language sentences and phrases

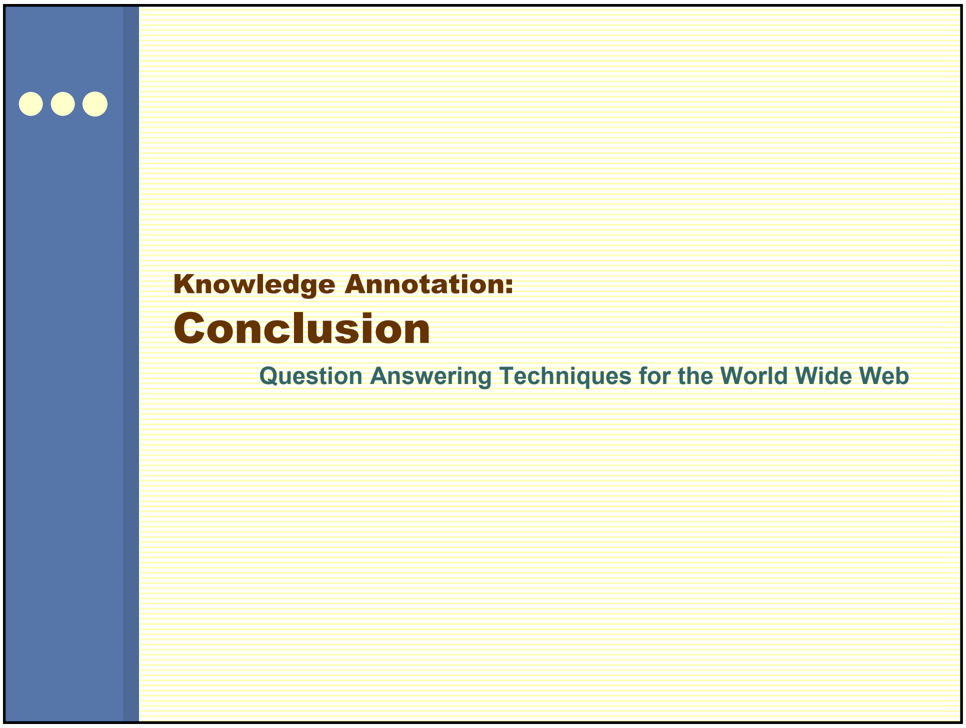


NL and the Semantic Web

- Natural language should be an integral component of the Semantic Web
- General strategy:
  - Weave natural language annotations directly into the RDF (Resource Description Framework)
  - Annotate RDF ontology fragments with natural language annotations

  In effect, we want to create "Sticky notes" for the Semantic Web [Karger *et al.* 2003]
- Prototype: START-Haystack collaboration

  Haystack: a Semantic Web platform [Huynh *et al.* 2002]
  + START: a question answering system
  = A question answering system for the Semantic Web



Knowledge Annotation:

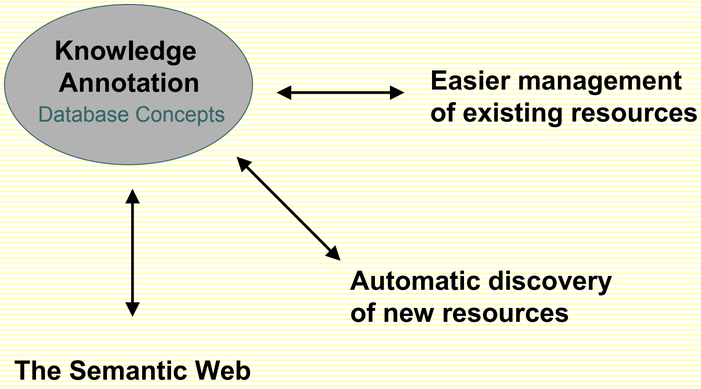
Conclusion

Question Answering Techniques for the World Wide Web

Summary

- Structured and semistructured Web resources can be organized to answer natural language questions
- Linguistically-sophisticated techniques for connecting questions with resources permit high precision question answering
- Knowledge annotation brings together many related fields of research, most notably NLP and database systems
- Future research focuses on discovery and management of semistructured resources, and the Semantic Web

The Future



Knowledge Annotation

Database Concepts

Easier management of existing resources

Automatic discovery of new resources

The Semantic Web

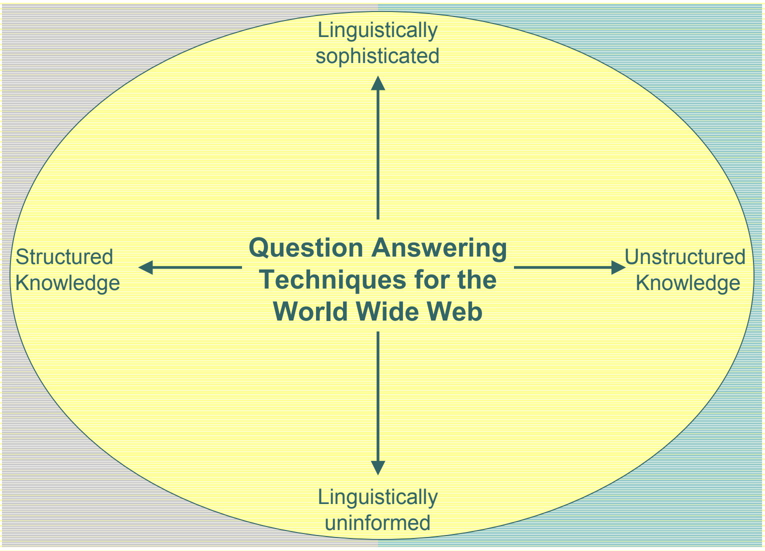
Conclusion

Question Answering Techniques for the World Wide Web

The Future of Web QA

- Two dimensions for organizing Web-based question answering strategies
  - Nature of the information
  - Nature of the technique
- The Web-based question answering system of the future…
  - Will be able to utilize the entire spectrum of available information from free text to highly structured databases
  - Will be able to seamlessly integrate robust, simple techniques with highly accurate linguistically-sophisticated ones

The Future of Web QA



Question Answering Techniques for the World Wide Web

Linguistically sophisticated

Structured Knowledge

Unstructured Knowledge

Linguistically uninformed

Acknowledgements

- We would like to thank Aaron Fernandes, Vineet Sinha, Stefanie Tellex, and Olzem Uzuner for their comments on earlier drafts of these slides. All remaining errors are, of course, our own.

References

Steven Abney, Michael Collins, and Amit Singhal. 2000. Answer extraction. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000)*.

Steven P. Abney. 1996. Partial parsing via finite-state cascades. *Journal of Natural Language Engineering*, 2(4):337–344.

Brad Adelberg. 1998. NoDoSE—a tool for semi-automatically extracting structured and semistructured data from text documents. *SIGMOD Record*, 27:283–294.

Brad Adelbery and Matt Denny. 1999. Building robust wrappers for text sources. Technical report, Northwestern University.

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries (DL'00)*.

Eugene Agichtein, Steve Lawrence, and Luis Gravano. 2001. Learning search engine specific query transformations for question answering. In *Proceedings of the Tenth International World Wide Web Conference (WWW10)*.

Diego Mollá Aliod, Jawad Berri, and Michael Hess. 1998. A real world implementation of answer extraction. In *Proceedings of 9th International Conference on Database and Expert Systems, Natural Language and Information Systems Workshop (NLIS'98)*.

Arne Andersson, N. Jesper Larsson, and Kurt Swanson. 1999. Suffix trees on words. *Algorithmica*, 23(3):246–260.

Evan L. Antworth. 1990. PC-KIMMO: A two-level processor for morphological analysis. Occasional Publications in Academic Computing 16, Summer Institute of Linguistics, Dallas, Texas.

Naveen Ashish and Craig Knoblock. 1997. Wrapper generation for semi-structured internet sources. In *Proceedings of the Workshop on Management of Semistructured Data at PODS/SIGMOD'97*.

Paolo Atzeni, Giansalvatore Mecca, and Paolo Merialdo. 1997. To weave the Web. In *Proceedings of the 23rd International Conference on Very Large Databases (VLDB 1997)*.

Michele Banko and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*.

Michele Banko, Eric Brill, Susan Dumais, and Jimmy Lin. 2002. AskMSR: Question answering using the World Wide Web. In *Proceedings of 2002 AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*.

Adam Berger and John Lafferty. 1999. Information retrieval as statistical translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-1999)*.

Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The Semantic Web. *Scientific American*, 284(5):34–43.

Tim Berners-Lee. 1999. *Weaving the Web*. Harper, New York.

Douglas Biber. 1986. Spoken and written textual dimensions in English: Resolving the contradictory findings. *Language*, 62(2):384–413.

Eric Breck, Marc Light, Gideon S. Mann, Ellen Riloff, Brianne Brown, Pranav Anand, Mats Rooth, and Michael Thelen. 2001. Looking under the hood: Tools for diagnosing your question answering engine. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001) Workshop on Open-Domain Question Answering*.

Eric Brill, Jimmy Lin, Michele Banko, Susan Dumais, and Andrew Ng. 2001. Data-intensive question answering. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*.

Eric Brill, Susan Dumais, and Michele Banko. 2002. An analysis of the AskMSR question-answering system. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the Sixth International World Wide Web Conference (WWW6)*.

Sergey Brin. 1998. Extracting patterns and relations from the World Wide Web. In *Proceedings of the WebDB Workshop—International Workshop on the Web and Databases, at EDBT '98*.

Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Sabine Buchholz. 2001. Using grammatical relations, answer frequencies and the World Wide Web for question answering. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*.

Chris Buckley and A. F. Lewit. 1985. Optimization of inverted vector searches. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-1985)*.

Robin D. Burke, Kristian J. Hammond, Vladimir A. Kulyukin, Steven L. Lytinen, Noriko Tomuro, and Scott Schoenberg. 1997. Question answering from frequently-asked question files: Experiences with the FAQ Finder system. Technical Report TR-97-05, University of Chicago.

Eugene Charniak. 1999. A Maximum-Entropy-Inspired parser. Technical Report CS-99-12, Brown University, Computer Science Department.

Jennifer Chu-Carroll, John Prager, Christopher Welty, Krzysztof Czuba, and David Ferrucci. 2002. A multi-strategy and multi-source approach to question answering. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*.

Charles Clarke, Gordon Cormack, and Thomas Lynam. 2001a. Exploiting redundancy in question answering. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2001)*.

Charles Clarke, Gordon Cormack, Thomas Lynam, C.M. Li, and Greg McLearn. 2001b. Web reinforced question answering (MultiText experiments for TREC 2001). In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*.

Charles Clarke, Gordon Cormack, Graeme Kemkes, Michael Laszlo, Thomas Lynam, Egidio Terra, and Philip Tilker. 2002. Statistical selection of exact answers (MultiText experiments for TREC 2002). In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*.

William Cohen. 2000. WHIRL: A word-based information representation language. *Artificial Intelligence*, 118(1–2):163–196.

Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam, and Sean Slattery. 1998a. Automatically deriving structured knowledge bases from on-line dictionaries. Technical Report CMU-CS-98-122, Carnegie Mellon University.

Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam, and Sean Slattery. 1998b. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-1998)*.

David Day, John Aberdeen, Lynette Hirschman, Robyn Kozierok, Patricia Robinson, and Marc Vilain. 1997. Mixed-initiative development of language processing systems. In *Proceedings of the Fifth ACL Conference on Applied Natural Language Processing (ANLP-1997)*.

Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. 2002. Web question answering: Is more always better? In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2002)*.

Sharon Flank, David Garfield, and Deborah Norkin. 1995. Digital image libraries: An innovating method for storage, retrieval, and selling of color images. In *Proceedings of the First International Symposium on Voice, Video, and Data Communications of the Society of Photo-Optical Instrumentation Engineers (SPIE)*.

Xiaoying Gao and Leon Sterling. 1999. AutoWrapper: automatic wrapper generation for multiple online services. In *Proceedings of Asia Pacific Web Conference 1999 (APWeb99)*.

Bert Green, Alice Wolf, Carol Chomsky, and Kenneth Laughery. 1961. BASEBALL: An automatic question answerer. In *Proceedings of the Western Joint Computer Conference*.

Jean-Robert Gruser, Louiqa Raschid, Marı́a Esther Vidal, and Laura Bright. 1998. Wrapper generation for web accessible data sources. In *Proceedings of the 3rd IFCIS International Conference on Cooperative Information Systems (CoopIS 1998)*.

Dan Gusfield. 1997. Linear time construction of suffix trees. In *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. University of Cambridge.

Laura M. Haas, Donald Kossmann, Edward L. Wimmers, and Jun Yang. 1997. Optimizing queries across diverse data sources. In *Proceedings of 23rd International Conference on Very Large Data Bases (VLDB 1997)*.

Joachim Hammer, Jason McHugh, and Hector Garcia-Molina. 1997. Semistructured data: The TSIMMIS experience. In *Proceedings of the First East-European Symposium on Advances in Databases and Information Systems (ADBIS'97)*.

Sanda Harabagiu and Dan Moldovan. 2001. Open-domain textual question answering: Tutorial given at naacl-2001.

Sanda Harabagiu and Dan Moldovan. 2002. Open-domain textual question answering: Tutorial given at coling-2002.

Sanda Harabagiu, Dan Moldovan, Marius Paşca, Rada Mihalcea, Mihai Surdeanu, Răzvan Bunescu, Roxana G îrju, Vasile Rus, and Paul Morărescu. 2000a. FALCON: Boosting knowledge for answer engines. In *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*.

Sanda Harabagiu, Marius Paşca, and Steven Maiorano. 2000b. Experiments with open-domain textual question answering. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*.

Gary G. Hendrix. 1977a. Human engineering for applied natural language processing. Technical Note 139, SRI International.

Gary G. Hendrix. 1977b. Human engineering for applied natural language processing. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI-77)*.

Ulf Hermjakob, Abdessamad Echihabi, and Daniel Marcu. 2002. Natural language based reformulation resource and Web exploitation for question answering. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*.

Lynette Hirschman and Robert Gaizauskas. 2001. Natural language question answering: The view from here. *Journal of Natural Language Engineering, Special Issue on Question Answering*, Fall–Winter.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001a. Towards semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research (HLT 2001)*.

Eduard Hovy, Ulf Hermjakob, and Chin-Yew Lin. 2001b. The use of external knowledge in factoid QA. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*.

Eduard Hovy, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2002. Using knowledge to facilitate factoid answer pinpointing. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002)*.

Chun-Nan Hsu and Chien-Chi Chang. 1999. Finite-state transducers for semi-structured text mining. In *Proceedings of the IJCAI-99 Workshop on Text Mining: Foundations, Techniques, and Applications*.

Chun-Nan Hsu. 1998. Initial results on wrapping semistructured Web pages with finite-state transducers and contextual rules. In *Proceedings of AAAI-1998 Workshop on AI and Information Integration*.

David Huynh, David Karger, and Dennis Quan. 2002. Haystack: A platform for creating, organizing and visualizing information using RDF. In *Proceedings of the Eleventh World Wide Web Conference Semantic Web Workshop*.

Frederick Jelinek. 1997. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, Massachusetts.

Hideo Joho and Mark Sanderson. 2000. Retrieving descriptive phrase from large amounts of free text. In *Proceedings of the Ninth International Conference on Information and Knowledge Management (CIKM 2000)*.

David Karger, Boris Katz, Jimmy Lin, and Dennis Quan. 2003. Sticky notes for the Semantic Web. In *Proceedings of the 2003 International Conference on Intelligent User Interfaces (IUI 2003)*.

Boris Katz and Beth Levin. 1988. Exploiting lexical regularities in designing natural language systems. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING-1988)*.

Boris Katz and Jimmy Lin. 2002a. Annotating the Semantic Web using natural language. In *Proceedings of the 2nd Workshop on NLP and XML at COLING-2002*.

Boris Katz and Jimmy Lin. 2002b. START and beyond. In *Proceedings of 6th World Multiconference on Systemics, Cybernetics, and Informatics (SCI 2002)*.

Boris Katz and Jimmy Lin. 2003. Selectively using relations to improve precision in question answering. In *Proceedings of the EACL-2003 Workshop on Natural Language Processing for Question Answering*.

Boris Katz, Deniz Yuret, Jimmy Lin, Sue Felshin, Rebecca Schulman, Adnan Ilik, Ali Ibrahim, and Philip Osafo-Kwaako. 1999. Integrating large lexicons and Web resources into a natural language query systen. In *Proceedings of the International Conference on Multimedia Computing and Systems (IEEE ICMCS '99)*.

Boris Katz, Sue Felshin, Deniz Yuret, Ali Ibrahim, Jimmy Lin, Gregory Marton, Alton Jerome McFarland, and Baris Temelkuran. 2002a. Omnibase: Uniform access to heterogeneous data for question answering. In *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB 2002)*.

Boris Katz, Jimmy Lin, and Sue Felshin. 2002b. The START multimedia information system: Current technology and future directions. In *Proceedings of the International Workshop on Multimedia Information Systems (MIS 2002)*.

Boris Katz, Jimmy Lin, and Dennis Quan. 2002c. Natural language annotations for the Semantic Web. In *Proceedings of the International Conference on Ontologies, Databases, and Application of Semantics (ODBASE 2002)*.

Boris Katz. 1988. Using English for indexing and retrieving. In *Proceedings of the 1st RIAO Conference on User-Oriented Content-Based Text and Image Handling (RIAO '88)*.

Boris Katz. 1990. Using English for indexing and retrieving. In Patrick Henry Winston and Sarah Alexandra Shellard, editors, *Artificial Intelligence at MIT: Expanding Frontiers*, volume 1. MIT Press.

Boris Katz. 1997. Annotating the World Wide Web using natural language. In *Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet (RIAO '97)*.

Brett Kessler, Geoffrey Nunberg, and Hinrich Schütze. 1997. Automatic detection of text genre. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL/EACL-1997)*.

Craig Knoblock, Steven Minton, Jose Luis Ambite, Naveen Ashish, Ion Muslea, Andrew Philpot, and Sheila Tejada. 2001. The Ariadne approach to Web-based information integration. *International Journal on Cooperative Information Systems (IJCIS) Special Issue on Intelligent Information Agents: Theory and Applications*, 10(1/2):145–169.

Nickolas Kushmerick, Daniel Weld, and Robert Doorenbos. 1997. Wrapper induction for information extraction. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*.

Nickolas Kushmerick. 1997. *Wrapper Induction for Information Extraction*. Ph.D. thesis, Department of Computer Science, University of Washington.

Cody Kwok, Oren Etzioni, and Daniel S. Weld. 2001. Scaling question answering to the Web. In *Proceedings of the Tenth International World Wide Web Conference (WWW10)*.

Steve Lawrence and C. Lee Giles. 1998. Context and page analysis for improved Web search. *IEEE Internet Computing*, 2(4):38–46.

Wendy G. Lehnert. 1977. A conceptual theory of question answering. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI-77)*.

Wendy G. Lehnert. 1981. A computational theory of human question answering. In Aravind K. Joshi, Bonnie L. Webber, and Ivan A. Sag, editors, *Elements of Discourse Understanding*, pages 145–176. Cambridge University Press, Cambridge, England.

Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. 1996. Querying heterogeneous information sources using source descriptions. In *Proceedings of 22nd International Conference on Very Large Data Bases (VLDB 1996)*.

Marc Light, Gideon S. Mann, Ellen Riloff, and Eric Breck. 2001. Analyses for elucidating current question answering technology. *Journal of Natural Language Engineering, Special Issue on Question Answering*, Fall–Winter.

Dekang Lin and Patrick Pantel. 2001a. DIRT—discovery of inference rules from text. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Dekang Lin and Patrick Pantel. 2001b. Discovery of inference rules for question answering. *Journal of Natural Language Engineering, Special Issue on Question Answering*, Fall–Winter.

Jimmy Lin, Aaron Fernandes, Boris Katz, Gregory Marton, and Stefanie Tellex. 2002. Extracting answers from the Web using knowledge annotation and knowledge mining techniques. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*.

Jimmy Lin, Dennis Quan, Vineet Sinha, Karun Bakshi, David Huynh, Boris Katz, and David R. Karger. 2003. The role of context in question answering systems. In *Proceedings of the 2003 Conference on Human Factors in Computing Systems (CHI 2003)*.

Jimmy Lin. 2001. Indexing and retrieving natural language using ternary expressions. Master's thesis, Massachusetts Institute of Technology.

Chin-Yew Lin. 2002a. The effectiveness of dictionary and Web-based answer reranking. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002)*.

Jimmy Lin. 2002b. The Web as a resource for question answering: Perspectives and challenges. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*.

John B. Lowe. 2000. What's in store for question answering? (invited talk). In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*.

Bernardo Magnini and Roberto Prevete. 2000. Exploiting lexical expansions and boolean compositions for Web querying. In *Proceedings of the ACL-2000 Workshop on Recent Advances in NLP and IR*.

Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. 2001. Multilingual question answering: the DIOGENE system. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*.

Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. 2002a. Is it the right answer? Exploiting Web redundancy for answer validation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*.

Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. 2002b. Mining knowledge from repeated co-occurrences: DIOGENE at TREC 2002. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*.

Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. 2002c. Towards automatic evaluation of Question/Answering systems. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*.

Gideon Mann. 2001. A statistical method for short answer extraction. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001) Workshop on Open-Domain Question Answering*.

Gideon Mann. 2002. Learning how to answer question using trivia games. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002)*.

Jason McHugh, Serge Abiteboul, Roy Goldman, Dallan Quass, and Jennifer Widom. 1997. Lore: A database management system for semistructured data. Technical report, Stanford University Database Group, February.

Mandar Mitra, Amit Singhal, and Chris Buckley. 1998. Improving automatic query expansion. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-1998)*.

Dan Moldovan, Sanda Harabagiu, Roxana Gîrju, Paul Morărescu, Finley Lacatusu, Adrian Novischi, Adriana Badulescu, and Orest Bolohan. 2002. LCC tools for question answering. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*.

Ion Muslea, Steve Minton, and Craig Knoblock. 1999. A hierarchical approach to wrapper induction. In *Proceedings of the 3rd International Conference on Autonomous Agents*.

John Prager, Dragomir Radev, Eric Brown, Anni Coden, and Valerie Samn. 1999. The use of predictive annotation for question answering in TREC8. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*.

Hong Qi, Jahna Otterbacher, Adam Winkel, and Dragomir R. Radev. 2002. The University of Michigan at TREC2002: question answering and novelty tracks. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*.

J. Ross Quinlan and R. Mike Cameron-Jones. 1993. FOIL: A midterm report. In *Proceedings of the 12th European Conference on Machine Learning*.

Dragomir Radev, Hong Qi, Zhiping Zheng, Sasha Blair-Goldensohn, Zhu Zhang, Waiguo Fan, and John Prager. 2001. Mining the Web for answers to natural language questions. In *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM 2001)*.

Dragomir Radev, Weiguo Fan, Hong Qi, Harris Wu, and Amardeep Grewal. 2002. Probabilistic question answering on the Web. In *Proceedings of the Eleventh International World Wide Web Conference (WWW2002)*.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*.

Steven E. Robertson and Steve Walker. 1997. On relevance weights with little relevance information. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-1997)*.

Stephen E. Robertson, Steve Walker, and Micheline Hancock-Beaulieu. 1998. Okapi at TREC-7: Automatic ad hoc, filtering, VLC and interactive. In *Proceedings of the 7th Text REtrieval Conference (TREC-7)*.

Arnaud Sahuguet and Fabien Azavant. 1999. WysiWyg Web Wrapper Factory (W4F). In *Proceedings of the Eighth International World Wide Web Conference (WWW8)*.

Gerard Salton. 1971. *The Smart Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Englewood Cliffs, New Jersey.

Daniel Sleator and Davy Temperly. 1991. Parsing English with a link grammar. Technical Report CMU-CS-91-196, Carnegie Mellon University, Department of Computer Science.

Daniel Sleator and Davy Temperly. 1993. Parsing English with a link grammar. In *Proceedings of the Third International Workshop on Parsing Technology*.

Alan F. Smeaton and Ian Qigley. 1996. Experiments on using semantic distances between words in image caption retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-1996)*.

Martin M. Soubbotin and Sergei M. Soubbotin. 2001. Patterns of potential answer expressions as clues to the right answers. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*.

Martin M. Soubbotin and Sergi M. Soubbotin. 2002. Use of patterns for detection of likely answer strings: A systematic approach. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*.

Rohini Srihari and Wei Li. 1999. Information extraction supported question answering. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*.

Gerald J. Sussman. 1973. A computational model of skill acquisition. Technical Report 297, MIT Artificial Intelligence Laboratory.

Anthony Tomasic, Louiqa Raschid, and Patrick Valduriez. 1996. Scaling heterogeneous distributed databases and the design of Disco. In *Proceedings of the 16th International Conference on Distributed Computing Systems*.

Ellen M. Voorhees and Dawn M. Tice. 1999. The TREC-8 question answering track evaluation. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*.

Ellen M. Voorhees and Dawn M. Tice. 2000a. Overview of the TREC-9 question answering track. In *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*.

Ellen M. Voorhees and Dawn M. Tice. 2000b. The TREC-8 question answering track evaluation. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC-2000)*.

Ellen M. Voorhees. 1994. Query expansion using lexical-semantics relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-1994)*.

Ellen M. Voorhees. 2001. Overview of the TREC 2001 question answering track. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*.

Ellen M. Voorhees. 2002a. The evaluation of question answering systems: Lessons learned from the TREC QA track. In *Proceedings of the Question Answering: Strategy and Resources Workshop at LREC-2002*.

Ellen M. Voorhees. 2002b. Overview of the TREC 2002 question answering track. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*.

David L. Waltz. 1973. Understanding line drawings of scenes with shadows. In Patrick H. Winston, editor, *Psychology of Computer Vision*. MIT Press, Cambridge, Massachusetts.

Robert Wilensky, David Ngi Chin, Marc Luria, James H. Martin, James Mayfield, and Dekai Wu. 1989. The Berkeley UNIX Consultant project. Technical Report CSD-89-520, Computer Science Division, the University of California at Berkeley.

Robert Wilensky. 1982. Talking to UNIX in English: An overview of an on-line UNIX consultant. Technical Report CSD-82-104, Computer Science Division, the University of California at Berkeley.

Terry Winograd. 1972. *Understanding Natural Language*. Academic Press, New York, New York.

Patrick H. Winston, Boris Katz, Thomas O. Binford, and Michael R. Lowry. 1983. Learning physical descriptions from functional definitions, examples, and precedents. In *Proceedings of the Third National Conference on Artificial Intelligence (AAAI-1983)*.

Patrick H. Winston. 1975. Learning structural descriptions from examples. In Patrick H. Winston, editor, *The Psychology of Computer Vision*. McGraw-Hill Book Company, New York, New York.

William A. Woods, Ronald M. Kaplan, and Bonnie L. Nash-Webber. 1972. The lunar sciences natural lanaugage information system: Final report. Technical Report 2378, BBN.

Jinxi Xu and W. Bruce Croft. 2000. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems*, 18(1):79–112.

Jinxi Xu, Ana Licuanan, Jonathan May, Scott Miller, and Ralph Weischedel. 2002. TREC2002 QA at BBN: Answer selection and confidence estimation. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*.

Hui Yang and Tat-Seng Chua. 2002. The integration of lexical knowledge and external resources for question answering. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*.

Jeonghee Yi and Neel Sundaresan. 1999. Mining the Web for acronyms using the duality of patterns and relations. In *Proceedings of the 1999 Workshop on Web Information and Data Management*.

Rémi Zajac. 2001. Towards ontological question answering. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001) Workshop on Open-Domain Question Answering*.

Dell Zhang and Wee Sun Lee. 2002. Web based pattern mining and matching approach to question answering. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*.

Zhiping Zheng. 2002a. AnswerBus question answering system. In *Proceeding of 2002 Human Language Technology Conference (HLT 2002)*.

Zhiping Zheng. 2002b. Developing a Web-based question answering system. In *Proceedings of the Eleventh International World Wide Web Conference (WWW2002)*.