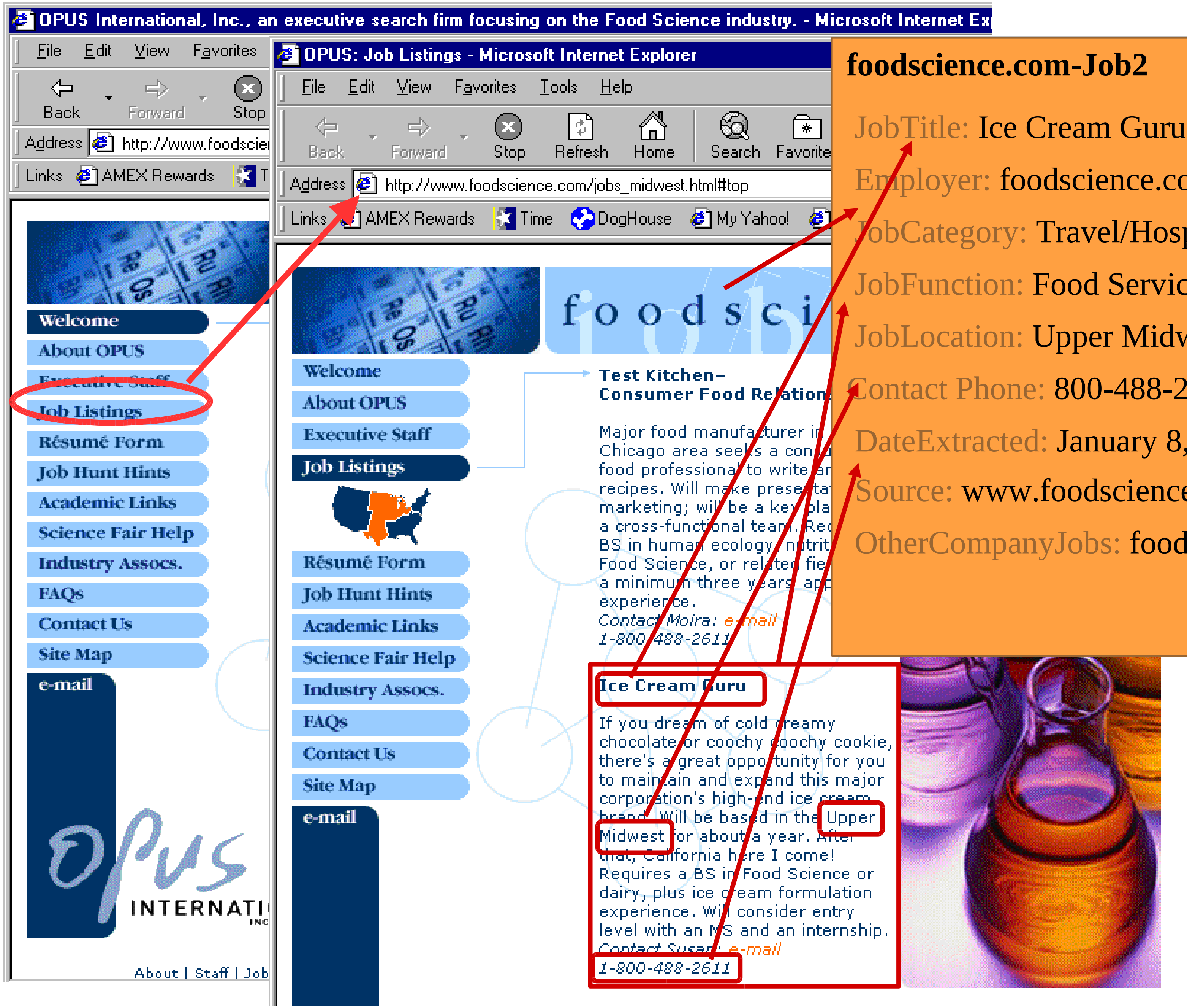
Information Extraction

- **Definition** (after Grishman 1997, Eikvil 1999):
  "The identificiation and extraction of instances of a particular class of events or relationships <u>in a natural language text</u> and their transformation into a structured representation (e.g. a database)."
  - *IR* retrieves *relevant documents* from collections
  - *IE* retrieves *relevant information* from documents
- Example: **AutoSlog** (Riloff)
  - input:
    - general syntactic patterns
    - annotated (marked-up) training documents
  - ouput:
    - instantiated patterns that extract particular information
  - **Autoslog-TS**: Extension that replaces need for annotated corpus with manual post-processing of sorted pattern list
- On the Web: natural language text -> (semi-)structured text

Slide taken from William Cohen

Extracting Job Openings from the Web



foodscience.com-Job2

JobTitle: Ice Cream Guru

Employer: foodscience.com

JobCategory: Travel/Hospitality

JobFunction: Food Services

JobLocation: Upper Midwest

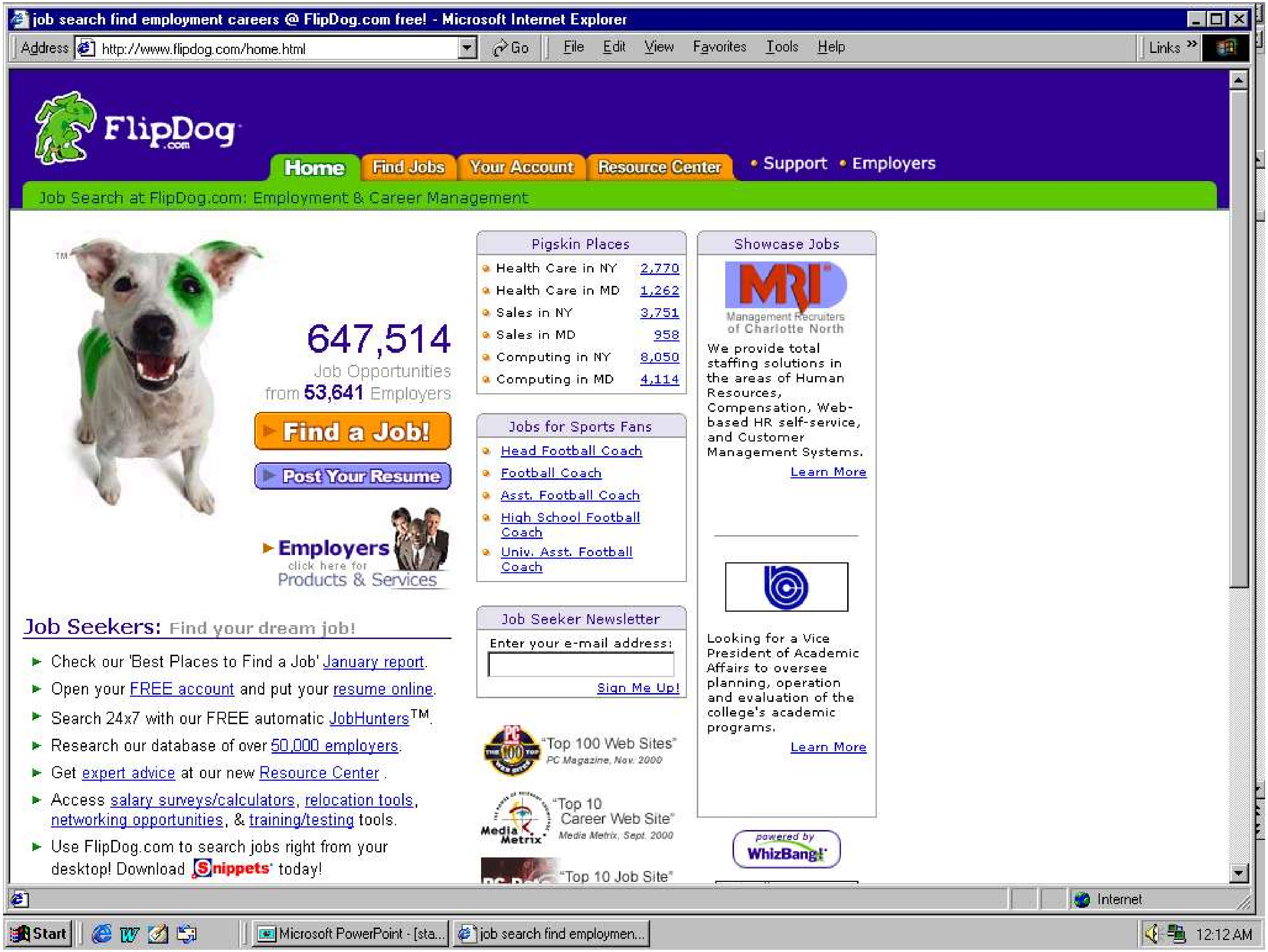
Contact Phone: 800-488-2611

DateExtracted: January 8, 2001

Source: www.foodscience.com/jobs_midwest.html

OtherCompanyJobs: foodscience.com-Job1

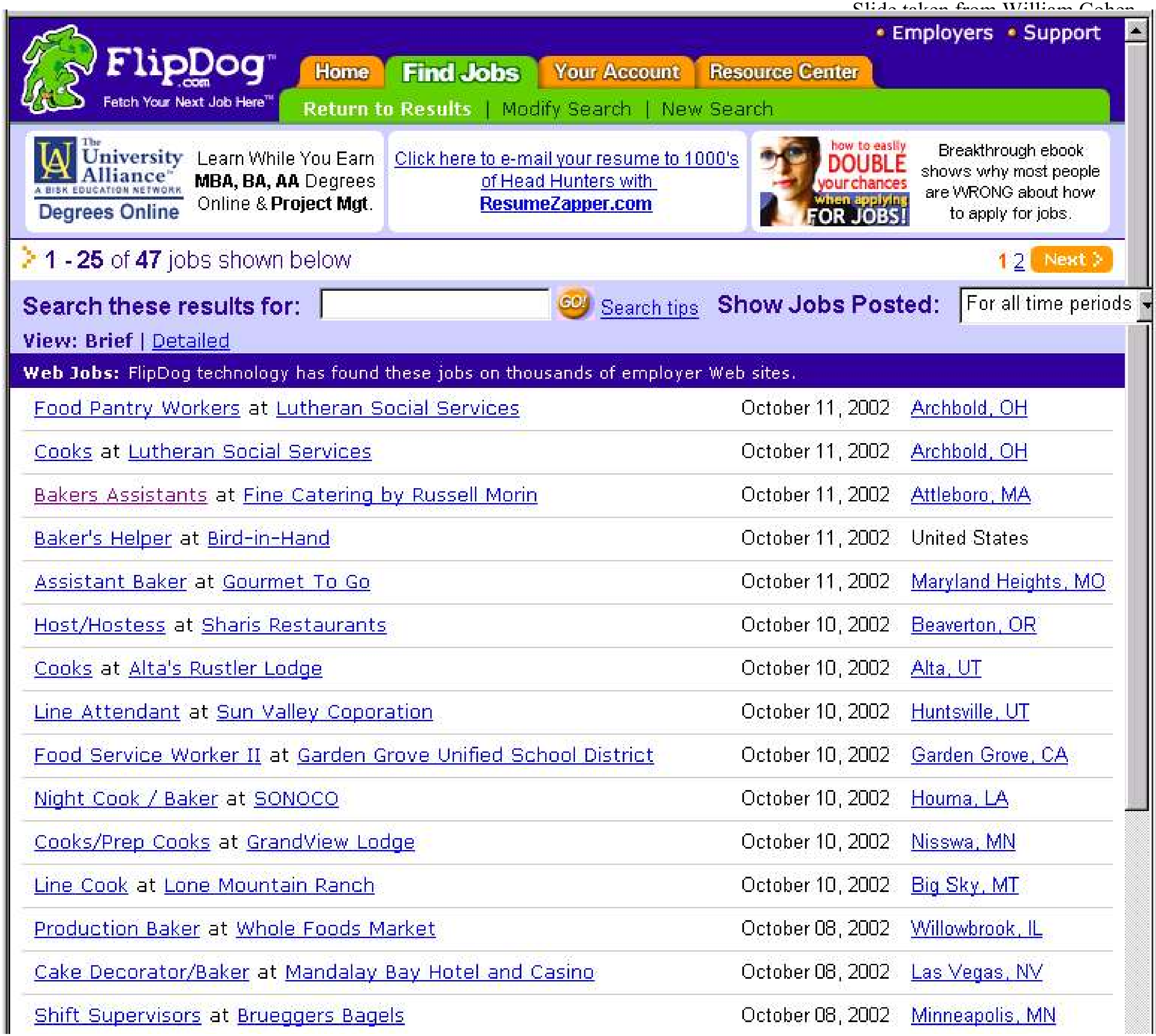
Example: A Solution



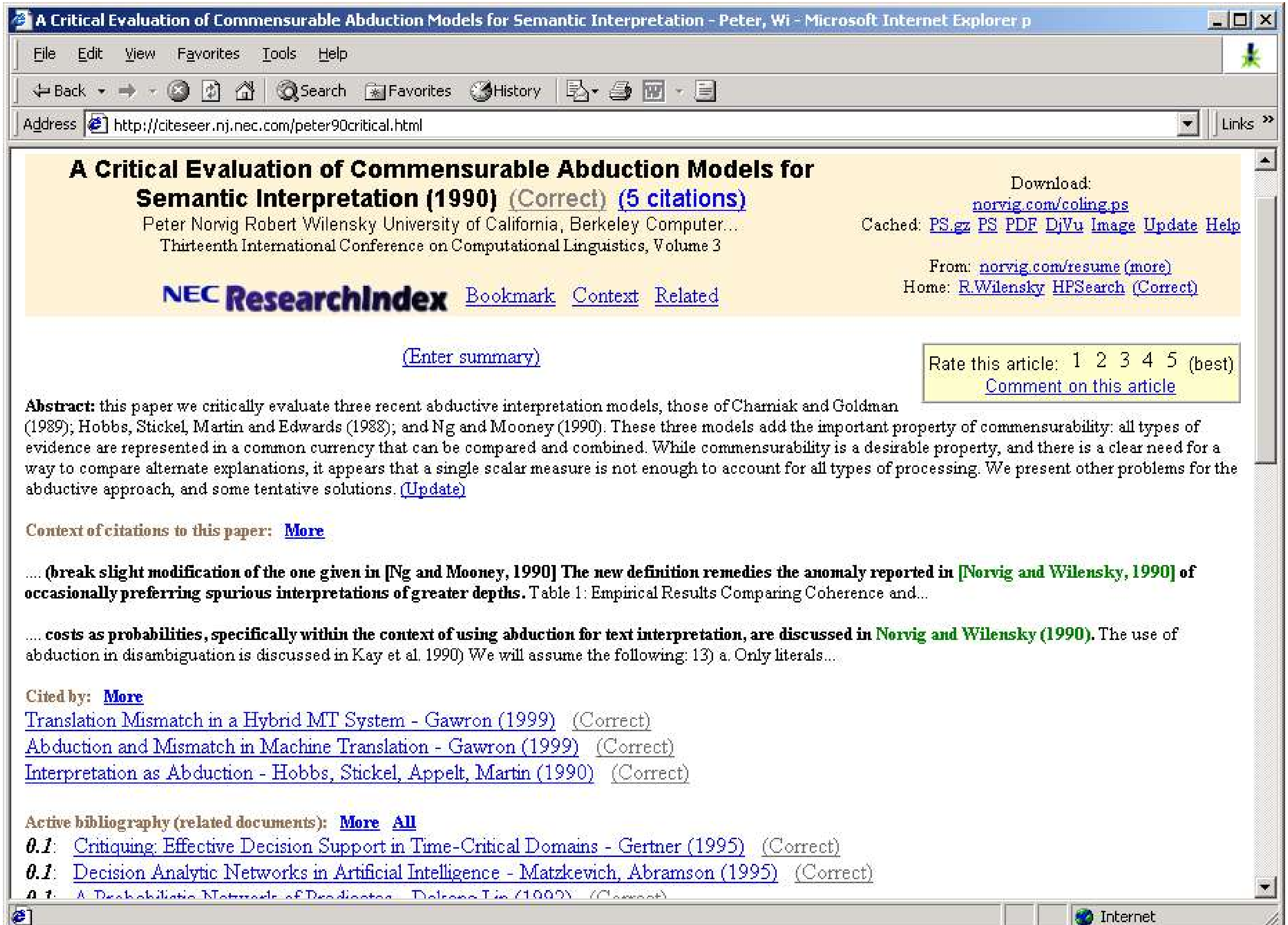
Slide taken from William Cohen

Job Openings:

Category = Food Services
Keyword = Baker
Location = Continental U.S.



IE from Research Papers



A Critical Evaluation of Commensurable Abduction Models for Semantic Interpretation - Peter, Wi - Microsoft Internet Explorer p

File Edit View Favorites Tools Help

Back Search Favorites History

Address http://citeseer.nj.nec.com/peter90critical.html

Links

A Critical Evaluation of Commensurable Abduction Models for Semantic Interpretation (1990) (Correct) (5 citations)

Peter Norvig Robert Wilensky University of California, Berkeley Computer...

Thirteenth International Conference on Computational Linguistics, Volume 3

NEC ResearchIndex Bookmark Context Related

Download: norvig.com/coling.ps

Cached: PS.gz PS PDF DjVu Image Update Help

From: norvig.com/resume (more)

Home: R.Wilensky HPSearch (Correct)

(Enter summary)

Rate this article: 1 2 3 4 5 (best)

Comment on this article

Abstract: this paper we critically evaluate three recent abductive interpretation models, those of Charniak and Goldman (1989); Hobbs, Stickel, Martin and Edwards (1988); and Ng and Mooney (1990). These three models add the important property of commensurability: all types of evidence are represented in a common currency that can be compared and combined. While commensurability is a desirable property, and there is a clear need for a way to compare alternate explanations, it appears that a single scalar measure is not enough to account for all types of processing. We present other problems for the abductive approach, and some tentative solutions. (Update)

Context of citations to this paper: More

.... **(break slight modification of the one given in [Ng and Mooney, 1990] The new definition remedies the anomaly reported in [Norvig and Wilensky, 1990] of occasionally preferring spurious interpretations of greater depths.** Table 1: Empirical Results Comparing Coherence and...

.... **costs as probabilities, specifically within the context of using abduction for text interpretation, are discussed in Norvig and Wilensky (1990).** The use of abduction in disambiguation is discussed in Kay et al. 1990) We will assume the following: 13) a. Only literals...

Cited by: More

Translation Mismatch in a Hybrid MT System - Gawron (1999) (Correct)

Abduction and Mismatch in Machine Translation - Gawron (1999) (Correct)

Interpretation as Abduction - Hobbs, Stickel, Appelt, Martin (1990) (Correct)

Active bibliography (related documents): More All

0.1: Critiquing: Effective Decision Support in Time-Critical Domains - Gertner (1995) (Correct)

0.1: Decision Analytic Networks in Artificial Intelligence - Matzkevich, Abramson (1995) (Correct)

Internet

What is "Information Extraction"

As a task:

Filling slots in a database from sub-segments of text.

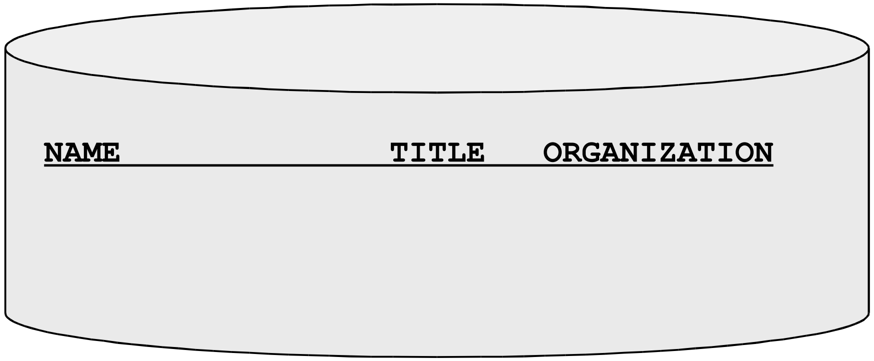
October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access.“

Richard Stallman, founder of the Free Software Foundation, countered saying…



What is “Information Extraction”

As a task:

Filling slots in a database from sub-segments of text.

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access.“

Richard Stallman, founder of the Free Software Foundation, countered saying…

IE

NAME	TITLE	ORGANIZATION
Bill Gates	CEO	Microsoft
Bill Veghte	VP	Microsoft
Richard Stallman	founder	Free Soft..

Landscape of IE Tasks (1/4):
Degree of Formatting

Text paragraphs without formatting

Astro Teller is the CEO and co-founder of BodyMedia. Astro holds a Ph.D. in Artificial Intelligence from Carnegie Mellon University, where he was inducted as a national Hertz fellow. His M.S. in symbolic and heuristic computation and B.S. in computer science are from Stanford University. His work in science, literature and business has appeared in international media from the New York Times to CNN to NPR.

Grammatical sentences and some formatting & links

Dr. Steven Minton - Founder/CTO
Dr. Minton is a fellow of the American Association of Artificial Intelligence and was the founder of the Journal of Artificial Intelligence Research. Prior to founding Fetch, Minton was a faculty member at USC and a project leader at USC's Information Sciences Institute. A graduate of Yale University and Carnegie Mellon University, Minton has been a Principal Investigator at NASA Ames and taught at Stanford, UC Berkeley and USC.

Frank Huybrechts - COO
Mr. Huybrechts has over 20 years of

- Press

Contact

- General information
- Directions maps

Non-grammatical snippets, rich formatting & links

Barto, Andrew G.	(413) 545-2109	barto@cs.umass.edu	CS276
Professor. Computational neuroscience, reinforcement learning, adaptive motor control, artificial neural networks, adaptive and learning control, motor development.			
Berger, Emery D.	(413) 577-4211	emery@cs.umass.edu	CS344
Assistant Professor.			
Brock, Oliver	(413) 577-0334	oli@cs.umass.edu	CS246
Assistant Professor.			
Clarke, Lori A.	(413) 545-1328	clarke@cs.umass.edu	CS304
Professor. Software verification, testing, and analysis; software architecture and design.			
Cohen, Paul R.	(413) 545-3638	cohen@cs.umass.edu	CS278
Professor. Planning, simulation, natural language, agent-based systems, intelligent data analysis, intelligent user interfaces.			

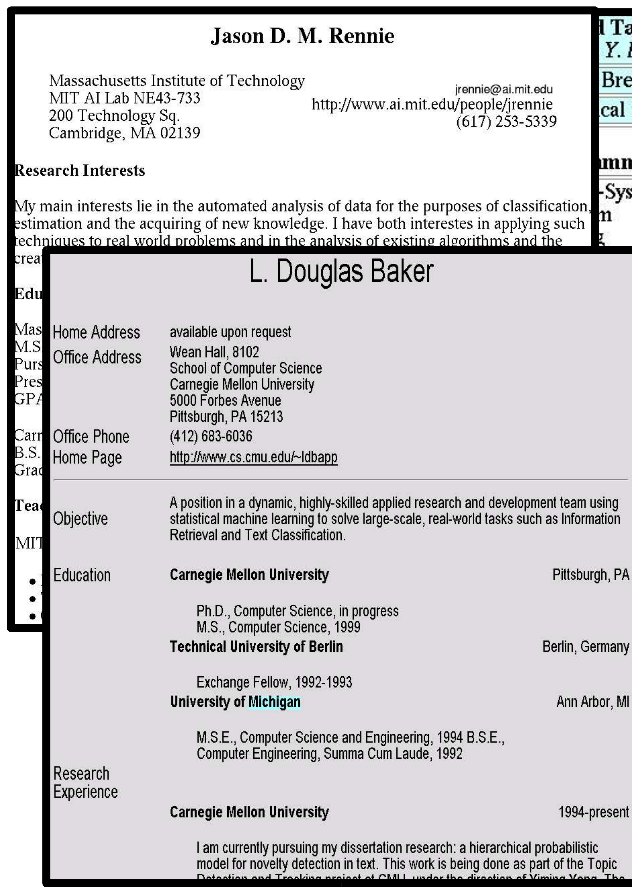
Tables

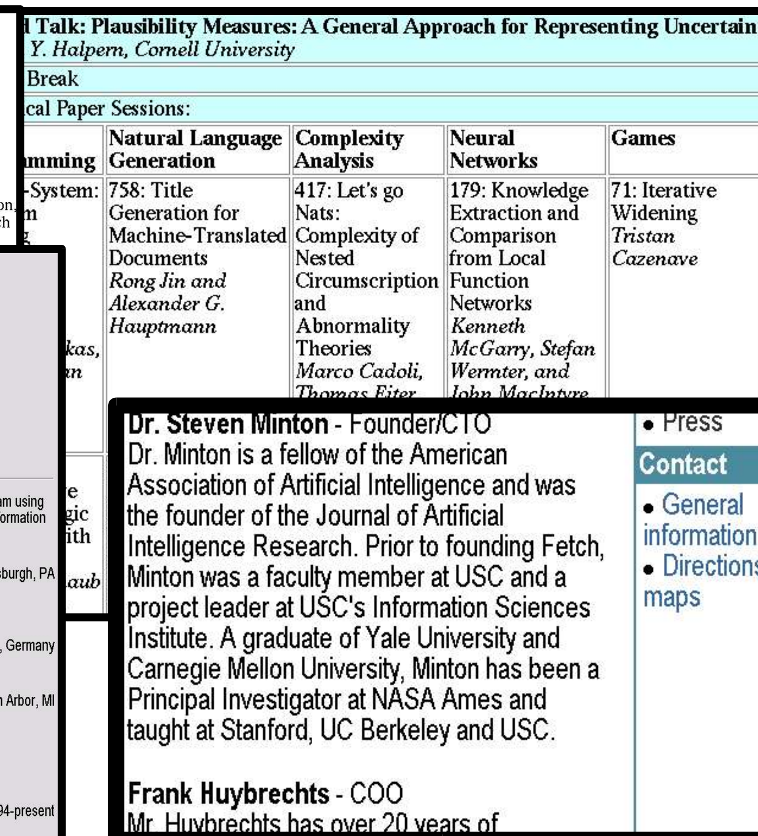
8:30 - 9:30 AM	**Invited Talk: Plausibility Measures: A General Approach for Representing Uncertainty** *Joseph Y. Halpern, Cornell University*				
9:30 - 10:00 AM	Coffee Break				
10:00 - 11:30 AM	Technical Paper Sessions:				
Cognitive Robotics	**Logic Programming**	**Natural Language Generation**	**Complexity Analysis**	**Neural Networks**	**Games**
739: A Logical Account of Causal and Topological Maps *Emilio Remolina and Benjamin Kuipers*	116: A-System: Problem Solving through Abduction *Marc Denecker, Antonis Kakas, and Bert Van Nuffelen*	758: Title Generation for Machine-Translated Documents *Rong Jin and Alexander G. Hauptmann*	417: Let's go Nats: Complexity of Nested Circumscription and Abnormality Theories *Marco Cadoli, Thomas Eiter, and Georg Gottlob*	179: Knowledge Extraction and Comparison from Local Function Networks *Kenneth McGarry, Stefan Wermter, and John MacIntyre*	71: Iterative Widening *Tristan Cazenave*
549: Online-Execution of ccGolog Plans *Henrik Grosskreutz and Gerhard Lakemeyer*	131: A Comparative Study of Logic Programs with Preference *Torsten Schaub and Kewen*	246: Dealing with Dependencies between Content Planning and Surface Realisation in a Pipeline Generation	470: A Perspective on Knowledge Compilation *Adnan Darwiche and Pierre Marquis*	258: Violation-Guided Learning for Constrained Formulations in Neural-Network Time-Series	353: Temporal Difference Learning Applied to a High Performance Game-Playing

Slide taken from William Cohen

Landscape of IE Tasks (2/4):
Intended Breadth of Coverage

Web site specific	**Genre specific**	**Wide, non-specific**
Formatting	Layout	Language
Amazon.com Book Pages	Resumes	University Names





Landscape of IE Tasks (3/4): Complexity

E.g. word patterns:

Closed set

U.S. states

He was born in Alabama…

The big Wyoming sky…

Regular set

U.S. phone numbers

Phone: (413) 545-1323

The CALD main office can be reached at 412-268-1299

Complex pattern

U.S. postal addresses

University of Arkansas
P.O. Box 140
Hope, AR 71802

Headquarters:
1128 Main Street, 4th Floor
Cincinnati, Ohio 45210

Ambiguous patterns, needing context and many sources of evidence

Person names

…was among the six houses sold by Hope Feldman that year.

Pawel Opalinski, Software Engineer at WhizBang Labs.

Landscape of IE Tasks (4/4): Single Field/Record

Jack Welch will retire as CEO of General Electric tomorrow. The top role at the Connecticut company will be filled by Jeffrey Immelt.

Single entity

Person: **Jack Welch**

Person: **Jeffrey Immelt**

Location: **Connecticut**

"Named entity" extraction

Binary relationship

Relation: **Person-Title**
Person: **Jack Welch**
Title: **CEO**

Relation: **Company-Location**
Company: **General Electric**
Location: **Connecticut**

N-ary record

Relation: **Succession**
Company: **General Electric**
Title: **CEO**
Out: **Jack Welsh**
In: **Jeffrey Immelt**

Recognizers

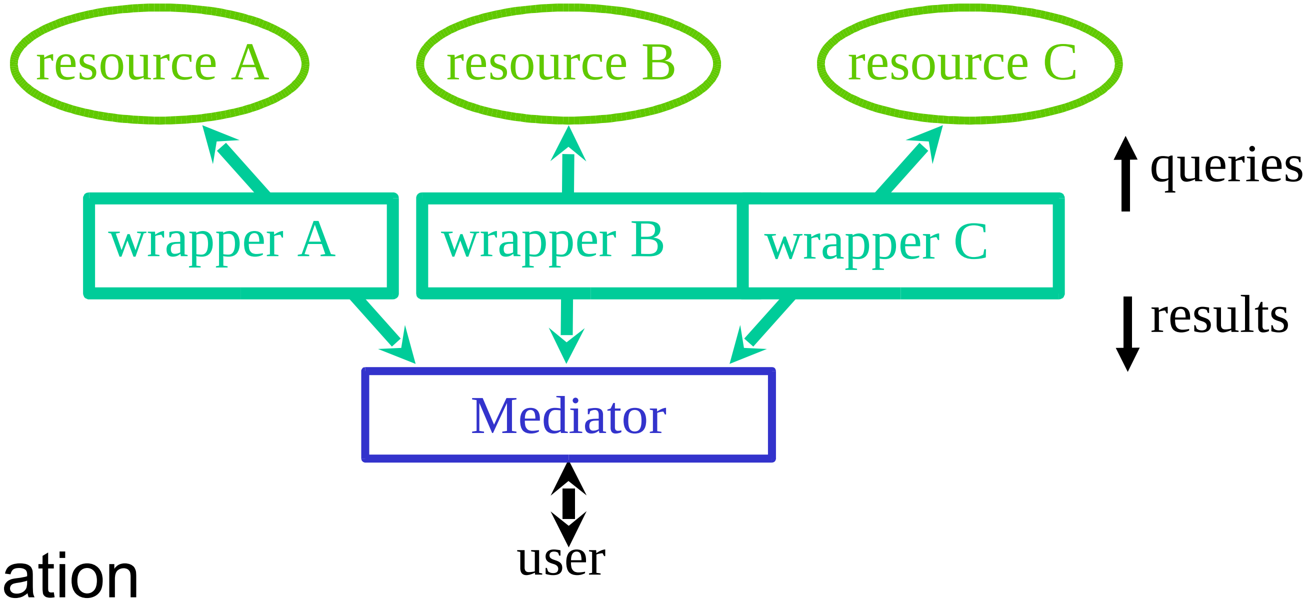
- Simple procedures to find pieces of information based on its appearance
  - e-mail addresses (easy)
  - telephone numbers (tricky)
  - street addresses (difficult)
- Examples:
  - Simple Web Crawlers can (and do) collect huge databases of e-mail addresses
  - Can also be used to automatically generate training examples for wrapper induction (Kushmerick, 2000)

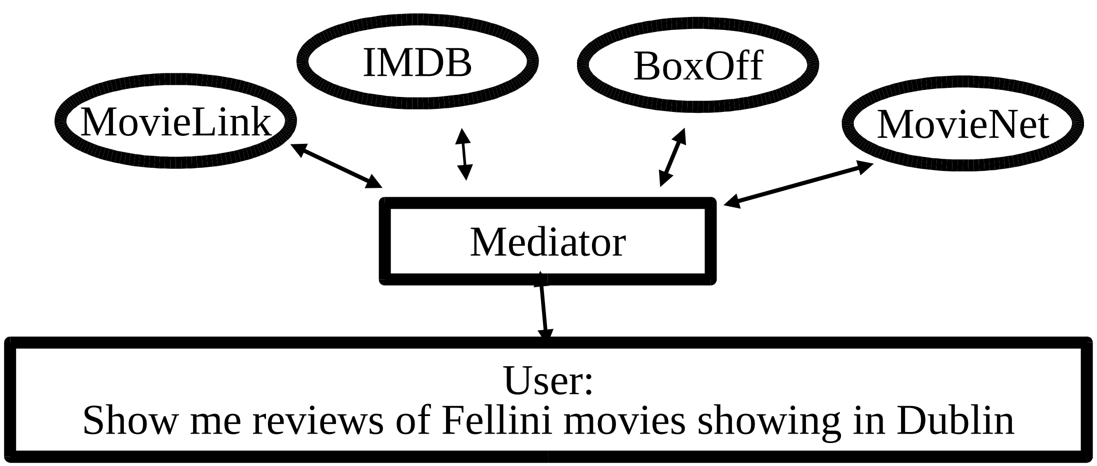
Wrappers

- Wrapper: (in an Information Extraction context)
  - A procedure that extracts certain pieces of information from (semi-)structured text (HTML)
- Examples:
  - Comparison Shoppers (Junglee, Shopbot/Jango, mySimon)
  - Meta-Search engines (citeseer, metacrawler)
  - News Agents (google news)
- Building Wrappers by hand:
  - time-consuming and error-prone (=> expensive)
  - Web-sites change frequently
    - mean-time to failure of wrappers: 1 month (Weld, 1998)
    - monthly failure rates of wrappers: 8% (Norvig, 1998)

Wrapper Induction: Motivation

- Wrappers
  - parse the contents of several sites
- Mediators
  - integrate the extracted information
- Example:





Wrapper Induction

- Automatic generation of wrappers from a few (annotated) sample pages
- Assumptions:
  - regularity in presentation of information
  - often machine-generated answers to queries
    - same header
    - same tail
    - inbetween a table/list of items that constitute the answer to the query
- Learn the delimiters between items of interest

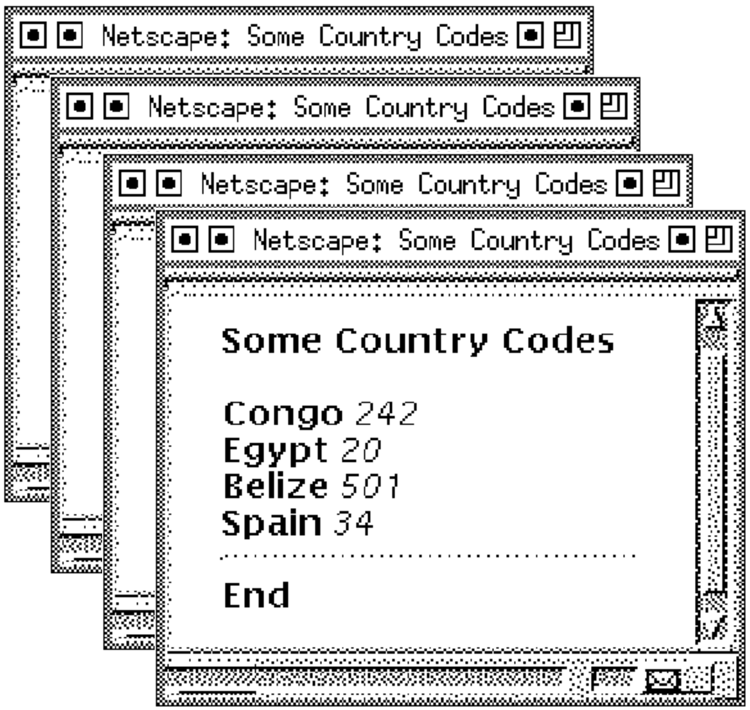
LR Wrappers (Kushmerick 2000)

- Very simple but nevertheless powerful wrapper class
- Assume that
  - only one "database" per page
  - information can be separated into tuples (records)
  - each tuple contains exactly k items (attributes)
- Wrapper consists of k delimiter pairs $<l_i, r_i>$,
  - li and *ri* are patterns that have to matched in the text

```
repeat
  foreach <l_i, r_i> ∈ {<l_1, r_1>, ..., <l_k, r_k>}
        find next occurrence of l_i
        find next occurrence of r_i
        extract text inbetween and store as the i-th value for this tuple
 until no more occurrences of l_1
```

Induction of LR Wrappers

Web Pages



Web Pages Labeled for Extraction

<HTML><HEAD>Some Country Codes</HEAD>
<B>**Congo**</B> <I>**242**</I>
<B>**Egypt**</B> <I>**20**</I>
<B>**Belize**</B> <I>**501**</I>
<B>**Spain**</B> <I>**34**</I>
</BODY></HTML>

Extracted Wrapper

⟨<B>, </B>, <I>, </I>⟩

$\langle\ l_1,\quad r_1,\quad l_2,\quad r_2\ \rangle$

Induction of LR Wrappers

- Heads: text before first tuple for each page
- Tails: text after last tuple for each page
- Separators: text between subsequent attributes

- Candidate delimiters:
  - Left: suffixes of the shortest of all separators to the left
    (including heads for $i = 1$)
  - Right: prefixes of the shortest of all separators to the right
    (including tails for $i = k$)

- Among the candidate delimiters, any one that satisfies a set of constraints can be selected
  - Constraints must ensure that the wrapper does not try to extract irrelevant parts of text (false positives)

Constraints for Delimiters

- the left delimiter l_i
  - must be a *proper* suffix of the text before each instance of the target
    - a proper suffix of a string means that
      - it is a suffix of the string
      - and it does not occur in any other place of the string
    - Example:
      - `cde` is a proper suffix of `deabcde`, `de` is a suffix but not proper
  - l_1 must not be part of any pages tail
    - otherwise extraction of a new tuple will be started at the end
- the right delimiter r_i
  - must be a prefix of the text after each instance of the target
  - must not be part of any value for attribute i
    - otherwise extraction will terminate prematurely

A Problem with LR-Wrappers

- Distracting text in Head or Tail

l_1 fires

```
<HTML><TITLE>Some Country Codes</TITLE>
  <BODY><B>Some Country Codes</B><P>
  <B>Congo</B> <I>242</I><BR>
  <B>Egypt</B> <I>20</I><BR>
  <B>Belize</B> <I>501</I><BR>
  <B>Spain</B> <I>34</I><BR>
  <HR><B>End</B></BODY></HTML>
```

- an LR-Wrapper cannot learn an extractor for this case
  - every candidate delimiter for l_1 occurs in the head
  - every candidate delimiter for l_1 occurs in the tail

HLRT-Wrappers

- Head-Tail-Left-Right Wrappers:
  - learn a separate delimiter for identifying head and tail

*Ignore page's **head** and **tail***

end of head

```
<HTML><TITLE>Some Country Codes</TITLE>
<BODY><B>Some Country Codes</B><P>
<B>Congo</B> <I>242</I><BR>
<B>Egypt</B> <I>20</I><BR>
<B>Belize</B> <I>501</I><BR>
<B>Spain</B> <I>34</I><BR>
<HR><B>End</B></BODY></HTML>
```

} head

} body

} tail

start of tail

More Expressive Wrapper Classes

- HLRT:
  - learn 2 additional delimiters to separate the head and the tail
  - ignores occurrence of l_i and r_i before h and after t
  - allows to process multiple "databases" in one document
- OCLR and HOCLRT:
  - for each tuple: learn an opening and closing delimiter
- N-LR and N-HLRT:
  - allows multi-valued attributes
  - allows optional attributes
    - RESTRICTION: if a value is specified, all previous values (of this tuple) must also be specified.

Evaluation

- Study on 30 randomly selected Web-sites from `www.search.com` (at that time a catalogue of hubs for various topics)
  - LR Wrapper was able to wrap 53%
  - LR + HLRT wrapped 60%
  - Addition of OC wrapping did not bring improvements
  - Addition of N-HLRT improved to 70%
- LR Wrappers are not limited to HTML-documents
  - any string can be extracted for delimiters, not just HTML tags
- All wrapper classes are PAC learnable
- Constraints become hard to handle

SoftMealy (Hsu & Dung, 1998)

- Problems with LR-Wrappers:
  - no permutations of attributes allowed
  - delimiters may not be sufficient to identify texts
- SoftMealy provides a general solution to problems with
  - missing attributes
  - attributes with multiple values
  - variable order of attributes
- Approach:
  - learn a finite-state transducer (FST) that encodes all possible sequences of attributes
  - each <u>state</u> represents a fact to be extracted
  - dummy states are used to skip parts of text
  - use *separators* ("invisible" borders) instead of delimiters
  - learn to recognize separators by defining their left and right context with *contextual rules* (<u>state transitions</u>)

Labelled Web Page

U (URL)

<LI><A HREF="mani.html">

N (Name) A (Academic title)

Mani Chandy</A>, <I>Professor of Computer Science</I> and

M (Admin title)

<I>Executive Officer for Computer Science</I>

U (URL)

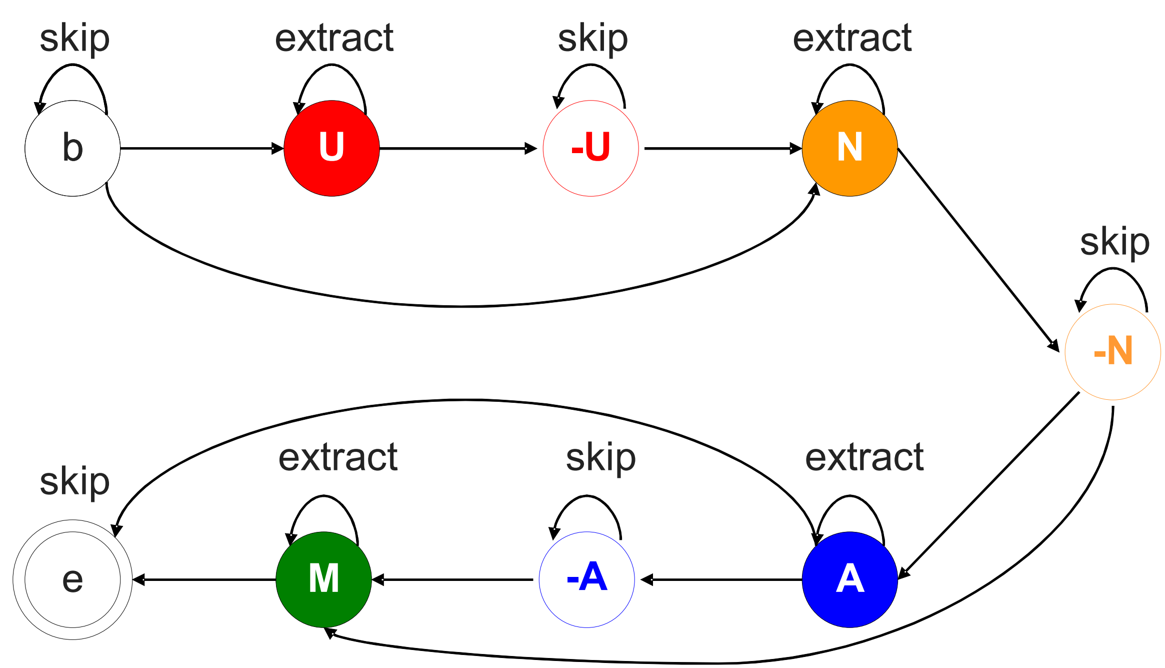
<LI><A HREF="david.html">

N (Name) M (Admin title)

David E. Breen</A>, <I>Assistant Director of Computer Graphics

Laboratory</I>

Sample FST



- **Contextual rule looks like:**

TRANSFER FROM state N **TO** state -N **IF**

left context = capitalized string

right context = HTML tag "</A>"

Wrapper Induction by Inductive Rule Learning

- Training Examples:
  - treat each slot independently (single slot extraction)
  - generate training example that represent the context of the slot (tokens before, after, and in the slot)
- Features are extracted from the context of a slot:
  - *token type:* word, number, punctuation, html-tag, ...
  - *formatting:* capitalized, italics, bold, font, ...
  - *location:* after/before line break, paragraph, ...
  - *html structure:* h1, a, href, table, td, center, ...
  - *relative position:* previous token, next token
- Learn Rules:
  - evaluate rules by counting correct matches as positive, wrong matches as negative (e.g., Laplace heuristic)

Example Systems

- RAPIER (Califf & Mooney, 1997):
  - based in a logic framework (ILP)
  - integrates some NLP (part-of-speech tags)
  - bottom-up learning with *lgg*: select two examples and compute the minimal generalization that covers both
- SRV (Freitag, 1998):
  - uses a large variety of features both for structured and unstructured text
  - top-down rule learning (Ripper-like)
- Expressive, general rule learning systems (e.g., ILP) could be used as well, but would lack domain-specific optimizations

WHISK (Soderland, 1999)

- multi-slot extraction
- rules represented as perl-like regular expressions
- can handle (semi-)structured and unstructured text

- top-down rule learning with seed instance (AQ-like)
  - choose a random training example
  - start with the most general rule
  - refine the rule using heuristics as in RIPPER-like algorithms (e.g., Laplace accuracy)
  - but only with conditions that appear in the training example

- use of user-specified semantic classes
  - e.g. BEDROOM = {brs|br|bds|bdrm|bd|bedroom|bedrooms|bed}
- integrated with interactive training based on a simple form of active learning

Example - WHISK

Training example:

```
<B>Capitol Hill -</B> 1 bedroom twnhme. fplc D/W
W/D. Undergrnd pkg incl. $675. 3 BR, 2nd flr of
turn of ctry HOME. incl. gar, grt N. Hill loc
$995. (206) 999-9999 <br>
```

Label:

- Rental:
  - area: Capitol Hill
  - bedrooms: 1
  - price: 675
- Rental:
  - area: Capitol Hill
  - bedrooms: 3
  - price 995

Starting Rule:

```
* ( * ) * ( * ) * ( * ) *
```

Final Rule:

(after seeing several examples):

```
START<B> ( * ) ' -' * ( DIGIT )
BEDROOM * '$' ( NUMBER ) *
```

Example - WHISK

Training example:

```
<B>Capitol Hill -</B> 1 bedroom twnhme. fplc D/W
W/D. Undergrnd pkg incl. $675. 3 BR, 2nd flr of
turn of ctry HOME. incl. gar, grt N. Hill loc
$995. (206) 999-9999 <br>
```

```
START<B> ( * ) ' -' * ( DIGIT )BEDROOM * '$' ( NUMBER ) *
```

```
BEDROOM = {brs|br|bds|bdrm|bd|bedroom|bedrooms|bed}
```

Example - WHISK

Training example:

```
<B>Capitol Hill -</B> 1 bedroom twnhme. fplc D/W
W/D. Undergrnd pkg incl. $675. 3 BR, 2nd flr of
turn of ctry HOME. incl. gar, grt N. Hill loc
$995. (206) 999-9999 <br>
```

```
START<B> ( * ) ' -' * ( DIGIT )BEDROOM * '$' ( NUMBER ) *
```

```
BEDROOM = {brs|br|bds|bdrm|bd|bedroom|bedrooms|bed}
```

Information Integration

- Data Integration (Data Warehousing):
  - Join different databases into a single view
  - Problem: Information may be encoded in different ways
- Information Integration:
  - Join information originating from different wrappers
  - Problem: extracted information is still free text
- Example:
  - *Data source 1:* Wrapper for Movie database
  - *Data source 2:* Wrapper Local movie show times
  - *Task:* Generate a page that integrates reviews into the local show times
  - *Problem:* Key relation (movie titles) will not match exactly

WHIRL (Cohen 1998)

- extension of DATALOG (or SQL) database queries that allows to deal with free text
  - models the information extracted by a wrapper as a relational table
- adresses the problem that
  - wrappers may not be able to extract the exact text
    - e.g., irrelevant information (directors, ratings, actors, etc.) might be extracted with title
  - text may be formulated differently on different Web-Sites
    - e.g., order and/or abbreviations of first, middle and last names
- Approach:
  - uses vector space model to represents textual fields
  - uses *similarity literals* to specify approximate matches
- `http://www.cs.cmu.edu/~wcohen/whirl/`

DATALOG vs. WHIRL

- Hard Queries:
  - items in a join must match exactly
- Items match or do not match
- Return all matches satisfying the query

- Soft Queries:
  - items in a join need only be "similar"
- Use cosine similarity to compute the degree of match [0,1]
- Return the best matches according to similarity
  - Use efficient A*-like search to find the r best matches according to similarity score (r-materialization)

WHIRL - Example

- Given two wrapped relations:
  - `review(Movie,Review)`
  - `showtime(Cinema,Movie,Time)`
- Sample Queries:
  - Hard Query (DATALOG):
    `showtime(C,M,T) & review(M,R)`

    M1 is similar to M2
  - Soft Query:
    `showtime(C,M1,T) & review(M2,R) & M1 ~ M2`
  - If the titles of the reviews could not be wrapped:
    `showtime(C,M,T) & review(R) & M ~ R`
  - Free text queries:
    `showtime(C,M1,T) & review(M2,R) & M1 ~ M2 & R~"excellent comedy with Bruce Willis"`

WHIRL - Scoring

- Possible answers Θ to queries Q are scored, i.e., a function $SCORE(Q,\Theta)$ is computed
- For a regular literal: $SCORE(B,\Theta)=s$
  if $B\Theta$ is a ground fact, 0 otherwise
  (usually $s = 1$, "degree of belief in the proposition")
- For a similarity literal $X{\sim}Y$:
  $$SCORE(X\sim Y,\Theta)=\mathrm{sim}(X\Theta,Y\Theta)$$
- Conjunctive Query $Q = B_1$ & ... & B_n
  $$SCORE(Q,\Theta)=\prod_i SCORE(B_i)$$
- A definite clause `Head :- B1, B2, ... Bn.`
  $$SCORE(Head)=1-\prod_i(1-SCORE(B_i))$$

Using WHIRL as Text Classifier

- represent labelled training documents in relation `train(Document,Class)`
- The following clause returns labels C ordered by similarity score of D to D1
  `classify(D,C) :- train(D1,C), D ~ D1.`
  - NOTE: multiple ground instantiations of the head (i.e, multiple bindings to the head) are combined using the definite clause similarity score!
- very similar to nearest neighbor classification
  - minor differences in combining evidence (similarity score)
- experimentally very competitive to conventional approaches