Information Extraction

Lecture #21

Introduction to Natural Language Processing
CMPSCI 585, Spring 2004
University of Massachusetts Amherst

Andrew McCallum

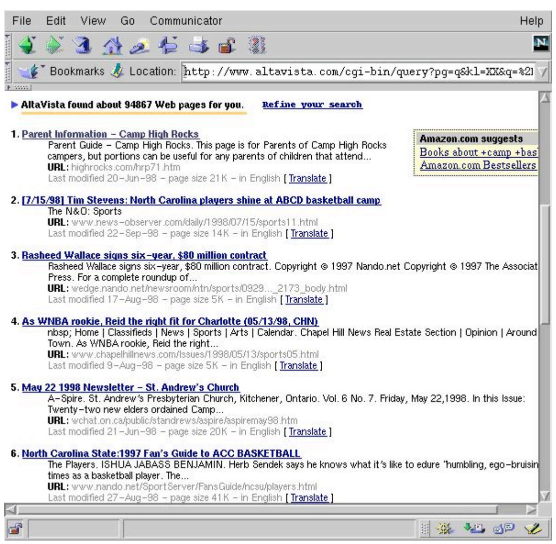
Main Points

- Why IE?
- Components of the IE problem and solution
- Approaches to IE segmentation and classification
  - Sliding window
  - Finite state machines
- IE for the Web
- Semi-supervised IE

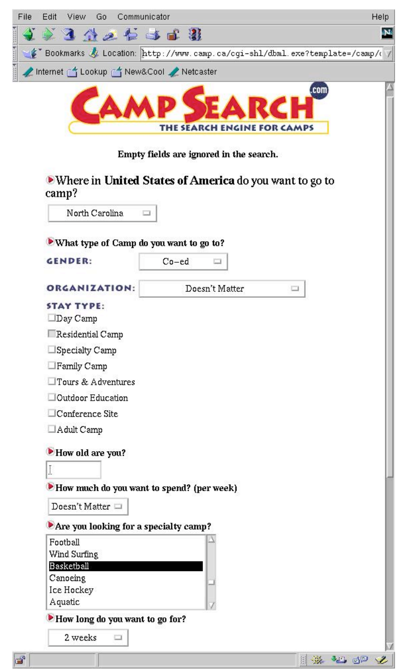
- Next time: relation extraction and coreference
- Optional class: CRFs for IE & coreference

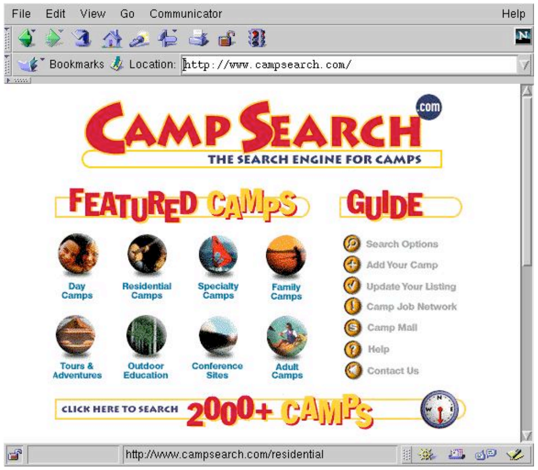
Query to General-Purpose Search Engine:

+camp +basketball "north carolina" "two weeks"

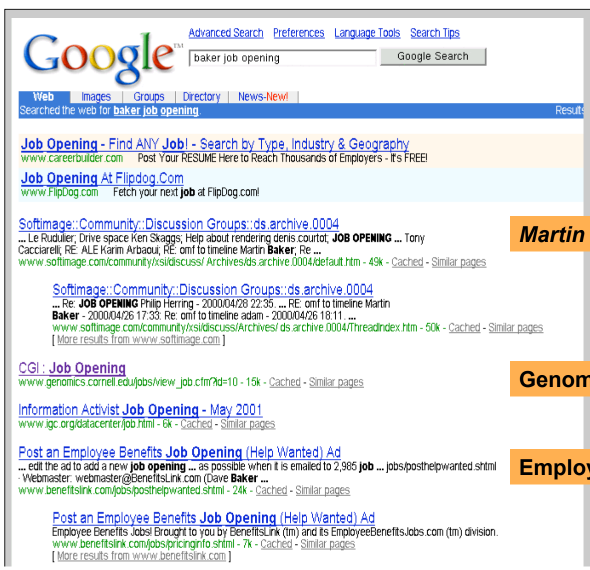


Domain-Specific Search Engine





Example: The Problem



***Martin Baker*, a person**

Genomics job

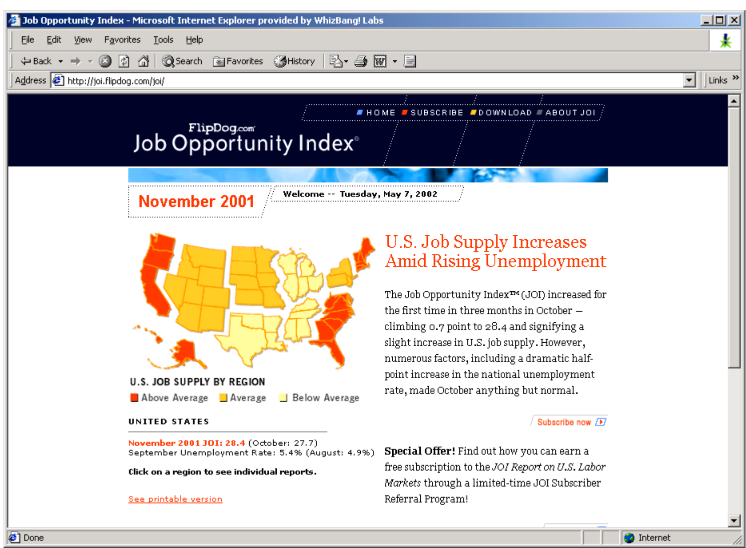
Employers job posting form

Example: A Solution

Extracting Job Openings from the Web

Job Openings:
Category = Food Services
Keyword = Baker
Location = Continental U.S.

Data Mining the Extracted Job Information



What is "Information Extraction"

As a task: **Filling slots in a database from sub-segments of text.**

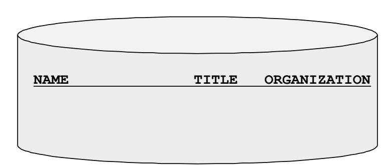
October 14, 2002, 4:00 a.m. PT

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..

What is “Information Extraction”

As a family of techniques: **Information Extraction = segmentation + classification + clustering + association**

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…

Microsoft Corporation
CEO
Bill Gates
Microsoft
Gates
Microsoft
Bill Veghte
Microsoft
VP
Richard Stallman
founder
Free Software Foundation

What is “Information Extraction”

As a family of techniques: **Information Extraction = segmentation + classification + association + clustering**

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…

Microsoft Corporation
CEO
Bill Gates
Microsoft
Gates
Microsoft
Bill Veghte
Microsoft
VP
Richard Stallman
founder
Free Software Foundation

What is “Information Extraction”

As a family of techniques: **Information Extraction = segmentation + classification + association + clustering**

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…

Microsoft Corporation
CEO
Bill Gates

Microsoft
Gates

Microsoft

Bill Veghte
Microsoft
VP

Richard Stallman
founder
Free Software Foundation

What is “Information Extraction”

As a family of techniques:

Information Extraction = segmentation + classification + association + clustering

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…

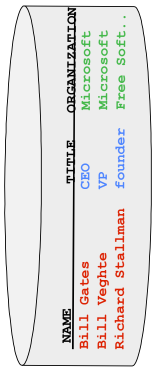
* Microsoft Corporation
CEO
Bill Gates

* Microsoft
Gates

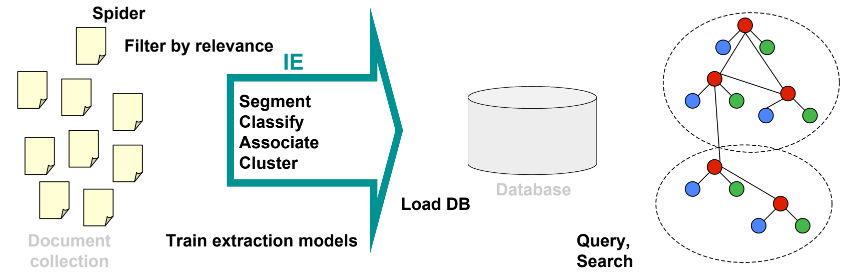
* Microsoft

Bill Veghte
* Microsoft
VP

Richard Stallman
founder
Free Software Foundation



IE in Context



IE History

Pre-Web

- Mostly news articles
  - De Jong’s *FRUMP* [1982]
    - Hand-built system to fill Schank-style “scripts” from news wire
  - *Message Understanding Conference (MUC)* DARPA [’87-’95], *TIPSTER* [’92-’96]
- Most early work dominated by hand-built models
  - E.g. SRI’s *FASTUS*, hand-built FSMs.
  - But by 1990’s, some machine learning: Lehnert, Cardie, Grishman and then HMMs: Elkan [Leek ’97], BBN [Bikel et al ’98]

Web

- AAAI ’94 Spring Symposium on “Software Agents”
  - Much discussion of ML applied to Web. Maes, Mitchell, Etzioni.
- Tom Mitchell’s WebKB, ‘96
  - Build KB’s from the Web.
- Wrapper Induction
  - Initially hand-build, then ML: [Soderland ’96], [Kushmeric ’97],…

What makes IE from the Web Different?

Less grammar, but more formatting & linking

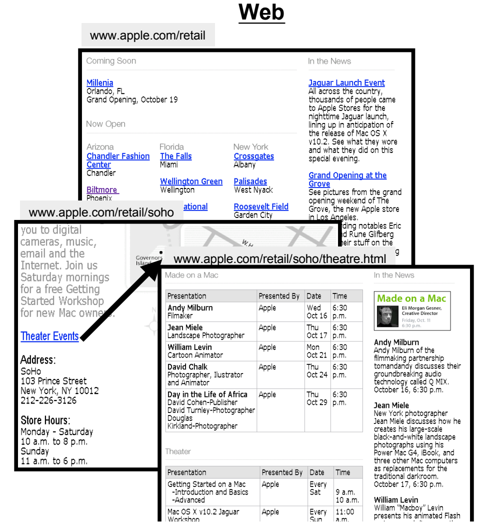
Newswire

Apple to Open Its First Retail Store
in New York City

MACWORLD EXPO, NEW YORK--July 17, 2002--Apple's first retail store in New York City will open in Manhattan's SoHo district on Thursday, July 18 at 8:00 a.m. EDT. The SoHo store will be Apple's largest retail store to date and is a stunning example of Apple's commitment to offering customers the world's best computer shopping experience.

"Fourteen months after opening our first retail store, our 31 stores are attracting over 100,000 visitors each week," said Steve Jobs, Apple's CEO. "We hope our SoHo store will surprise and delight both Mac and PC users who want to see everything the Mac can do to enhance their digital lifestyles."

Web



The directory structure, link structure, formatting & layout of the Web is its own new grammar.

Landscape of IE Tasks (1/4): Pattern Feature Domain

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

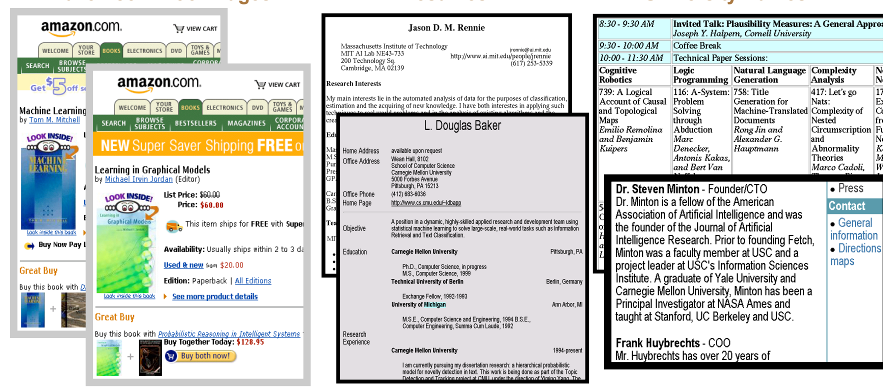
Barto, Andrew G.	(413) 545-2109	barto@cs.umass.edu	CS276
Professor. Computational neuroscience, reinforcement learning, adaptive motor control, artificial neural networks, adaptive and learning control, motor development.			
Berger, Emery D.	(413) 577-4211	emery@cs.umass.edu	CS344
Assistant Professor.			
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*	131: A Comparative Study of Logic Programs with	246: Dealing with Dependencies between Content Planning and	470: A Perspective on Knowledge Compilation	258: Violation-Guided Learning for Constrained	353: Temporal Difference Learning Applied to a

Landscape of IE Tasks (2/4): Pattern Scope

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



Landscape of IE Tasks (3/4): Pattern 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): Pattern Combinations

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

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

"Named entity" extraction

Evaluation of Single Entity Extraction

TRUTH:

Michael Kearns and Sebastian Seung will start Monday's tutorial, followed by Richard M. Karpe and Martin Cooke.

PRED:

Michael Kearns and Sebastian Seung will start Monday's tutorial, followed by Richard M. Karpe and Martin Cooke.

$$\text{Precision} = \frac{\#\text{ correctly predicted segments}}{\#\text{ predicted segments}} = \frac{2}{6}$$

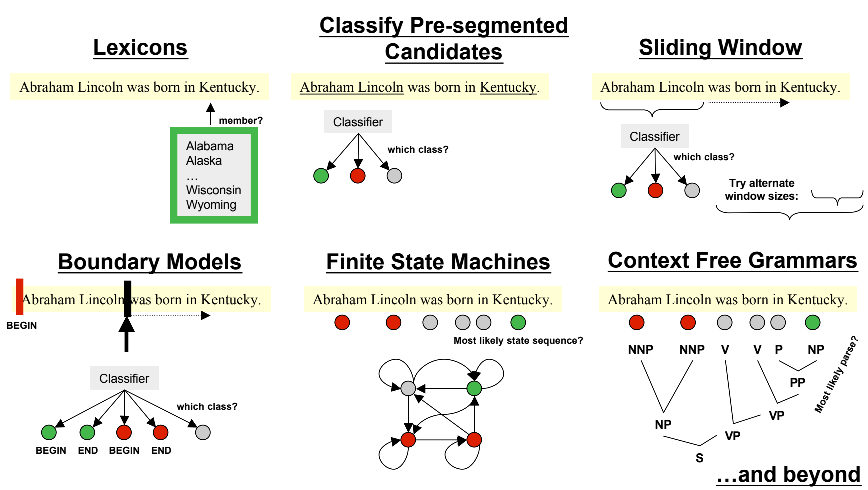
$$\text{Recall} = \frac{\#\text{ correctly predicted segments}}{\#\text{ true segments}} = \frac{2}{4}$$

$$\text{F1} = \text{Harmonic mean of Precision \& Recall} = \frac{1}{((1/P) + (1/R))\,/\,2}$$

State of the Art Performance

- Named entity recognition
  - Person, Location, Organization, …
  - F1 in high 80's or low- to mid-90's
- Binary relation extraction
  - Contained-in (Location1, Location2)
    Member-of (Person1, Organization1)
  - F1 in 60's or 70's or 80's
- Wrapper induction
  - Extremely accurate performance obtainable
  - Human effort (~30min) required on each site

Landscape of IE Techniques (1/1): Models



Any of these models can be used to capture words, formatting or both.

Sliding Windows

Extraction by Sliding Window

**E.g.
Looking for
seminar
location**

```
        GRAND CHALLENGES FOR MACHINE LEARNING

               Jaime Carbonell
          School of Computer Science
         Carnegie Mellon University

                  3:30 pm
               7500 Wean Hall

Machine learning has evolved from obscurity
in the 1970s into a vibrant and popular
discipline in artificial intelligence
during the 1980s and 1990s.   As a result
of its success and growth, machine learning
is evolving into a collection of related
disciplines: inductive concept acquisition,
analytic learning in problem solving (e.g.
analogy, explanation-based learning),
learning theory (e.g. PAC learning),
genetic algorithms, connectionist learning,
hybrid systems, and so on.
```

CMU UseNet Seminar Announcement

Extraction by Sliding Window

**E.g.
Looking for
seminar
location**

```
        GRAND CHALLENGES FOR MACHINE LEARNING

               Jaime Carbonell
          School of Computer Science
         Carnegie Mellon University

                  3:30 pm
               7500 Wean Hall

Machine learning has evolved from obscurity
in the 1970s into a vibrant and popular
discipline in artificial intelligence
during the 1980s and 1990s.   As a result
of its success and growth, machine learning
is evolving into a collection of related
disciplines: inductive concept acquisition,
analytic learning in problem solving (e.g.
analogy, explanation-based learning),
learning theory (e.g. PAC learning),
genetic algorithms, connectionist learning,
hybrid systems, and so on.
```

CMU UseNet Seminar Announcement

Extraction by Sliding Window

**E.g.
Looking for
seminar
location**

```
        GRAND CHALLENGES FOR MACHINE LEARNING

               Jaime Carbonell
          School of Computer Science
         Carnegie Mellon University

                  3:30 pm
               7500 Wean Hall

Machine learning has evolved from obscurity
in the 1970s into a vibrant and popular
discipline in artificial intelligence
during the 1980s and 1990s.   As a result
of its success and growth, machine learning
is evolving into a collection of related
disciplines: inductive concept acquisition,
analytic learning in problem solving (e.g.
analogy, explanation-based learning),
learning theory (e.g. PAC learning),
genetic algorithms, connectionist learning,
hybrid systems, and so on.
```

CMU UseNet Seminar Announcement

Extraction by Sliding Window

**E.g.
Looking for
seminar
location**

```
        GRAND CHALLENGES FOR MACHINE LEARNING

               Jaime Carbonell
          School of Computer Science
         Carnegie Mellon University

                  3:30 pm
               7500 Wean Hall

Machine learning has evolved from obscurity
in the 1970s into a vibrant and popular
discipline in artificial intelligence
during the 1980s and 1990s.   As a result
of its success and growth, machine learning
is evolving into a collection of related
disciplines: inductive concept acquisition,
analytic learning in problem solving (e.g.
analogy, explanation-based learning),
learning theory (e.g. PAC learning),
genetic algorithms, connectionist learning,
hybrid systems, and so on.
```

CMU UseNet Seminar Announcement

A “Naïve Bayes” Sliding Window Model

[Freitag 1997]

… **00 : pm Place : Wean Hall Rm 5409 Speaker : Sebastian Thrun** …

$\underbrace{w_{t-m} \quad \ldots \quad w_{t-1}}_{\text{prefix}} \quad \underbrace{w_{t} \quad \ldots \quad w_{t+n}}_{\text{contents}} \quad \underbrace{w_{t+n+1} \quad \ldots \quad w_{t+n+m}}_{\text{suffix}}$

P(“Wean Hall Rm 5409” = LOCATION) =

Ä

Prior probability of start position | Prior probability of length | Probability prefix words | Probability contents words | Probability suffix words

Try all start positions and reasonable lengths

Estimate these probabilities by (smoothed) counts from labeled training data.

If P(“Wean Hall Rm 5409” = LOCATION) **is above some threshold, extract it.**

Other examples of sliding window: ***[Baluja et al 2000]***
(decision tree over individual words & their context)

“Naïve Bayes” Sliding Window Results

Domain: CMU UseNet Seminar Announcements

```
   GRAND CHALLENGES FOR MACHINE LEARNING

            Jaime Carbonell
        School of Computer Science
      Carnegie Mellon University

                 3:30 pm
              7500 Wean Hall

Machine learning has evolved from obscurity
in the 1970s into a vibrant and popular
discipline in artificial intelligence during
the 1980s and 1990s.   As a result of its
success and growth, machine learning is
evolving into a collection of related
disciplines: inductive concept acquisition,
analytic learning in problem solving (e.g.
analogy, explanation-based learning),
learning theory (e.g. PAC learning), genetic
algorithms, connectionist learning, hybrid
systems, and so on.
```

Field	F1
Person Name:	30%
Location:	61%
Start Time:	98%

SRV: a realistic sliding-window-classifier IE system

[Frietag AAAI ‘98]

- What windows to consider?
  - *all* windows containing **as many** tokens as the shortest example, but **no more** tokens than the longest example
- How to represent a classifier? It might:
  - Restrict the **length** of window;
  - Restrict the **vocabulary** or formatting used before/after/inside window;
  - Restrict the **relative order** of tokens;
  - Etc…

<title>Course Information for CS213</title>
<h1>CS 213 C++ Programming</h1>

“A token followed by a 3-char numeric token just after the title”

SRV: a rule-learner for sliding-window classification

- Top-down rule learning:

let RULES = ;;
while (*there are uncovered positive examples*) {
  // construct a rule R to add to RULES
  let R be a rule covering all examples;
  while (R *covers too many negative examples)* {
    let C = argmax_C VALUE(R, R **Æ** C, *uncoveredExamples*)
      over some set of candidate conditions C
    let R = R **Æ** C;
  }
  let RULES = RULES **[** {R};
}

SRV: a rule-learner for sliding-window classification

Search metric: SRV algorithm greedily adds conditions to maximize "information gain" of *R*

VALUE(R,R',Data) = lData|*p (p log p – p' log p')

where *p (p')* is fraction of data covered by *R (R')*

To prevent overfitting:

rules are built on 2/3 of data, then their false positive rate is estimated with a Dirichlet on the 1/3 holdout set.

Candidate conditions: …

Learning "first-order" rules

- A sample "zero-th" order rule set:
  (tok1InTitle **Æ** **:**tok1StartsPara **Æ** tok2triple)
  Ç (prevtok2EqCourse **Æ** prevtok1EqNumber) **Ç** **…**
- First-order "rules" can be learned the same way—with additional search to find best "condition"
  phrase(X) **Ã** firstToken(X,A), **:**startPara(A), nextToken(A,B), triple(B)
  phrase(X) **Ã** firstToken(X,A), prevToken(A,C), eq(C,'number'), prevToken(C,D), eq(D,'course')
- Semantics:
  "p(X) **Ã** q(X),r(X,Y),s(Y)" = "{X : **9** Y : q(X) **Æ** r(X,Y) **Æ** s(Y)}"

SRV: a rule-learner for sliding-window classification

- Primitive predicates used by SRV:
  - *token(X,W), allLowerCase(W), numerical(W), …*
  - *nextToken(W,U), previousToken(W,V)*
- HTML-specific predicates:
  - *inTitleTag(W), inH1Tag(W), inEmTag(W),…*
  - *emphasized(W) = "inEmTag(W) or inBTag(W) or …"*
  - *tableNextCol(W,U) = "U is some token in the column after the column W is in"*
  - *tablePreviousCol(W,V), tableRowHeader(W,T),…*

SRV: a rule-learner for sliding-window classification

- Non-primitive "conditions" used by SRV:
  - *every(+X, f, c)* = ***8*** *W**2**X : f(W)=c*
    - variables tagged "+" must be used in earlier conditions
    - underlined values will be replaced by constants, e.g., "every(X, isCapitalized, true)"
  - *some(+X, W, <$f_1,\ldots,f_k$>, g, c)*= ***9*** *W: g(f_k(…(f_1(W)…))=c*
    - *e.g., some(X, W, [prevTok,prevTok],inTitle,false)*
    - *set of "paths" <$f_1,\ldots,f_k$> considered grows over time.*
  - *tokenLength(+X, relop, c):*
  - *position(+W,direction,relop, c):*
    - *e.g., tokenLength(X,>,4), position(W,fromEnd,<,2)*

Utility of non-primitive conditions in greedy rule search

Greedy search for first-order rules is hard because useful conditions can give **no** immediate benefit:

phrase(X) Ã token(X,A), prevToken(A,B),inTitle(B),
nextToken(A,C), tripleton(C)

"A token followed by a 3-char numeric token just after the title"

<title>Course Information for CS213</title>
<h1>CS 213 C++ Programming</h1>

courseNumber(X) Ã
tokenLength(X,=,2),
every(X, inTitle, false),
some(X, A, <previousToken>, inTitle, true),
some(X, B, <>. tripleton, true)

Non-primitive conditions make greedy search easier

Rapier: an alternative approach

[Califf & Mooney, AAAI '99]

A bottom-up rule learner:

initialize RULES to be one rule per example;
repeat {
randomly pick *N* pairs of rules (R_i,R_j);
let $\{G_{1\ldots},G_N\}$ be the consistent pairwise generalizations;
let $G^* = argmin_G\ COST(G,RULES)$;
let RULES = RULES **[** {G*} – {R': G* ¶ R'}
}
where *COST(G,RULES) = size of RULES-* {R': G ¶ R'} and
"G¶ R" means every *example* matching G matches R

<title>Course Information for CS213</title>
<h1>CS 213 C++ Programming</h1> …

Differences dropped

courseNum(window1) Ã token(window1,'CS'), doubleton('CS'), prevToken('CS','CS213'), inTitle('CS213'), nextTok('CS','213'), numeric('213'), tripleton('213'), nextTok('213','C++'), tripleton('C++'), ….

<title>Syllabus and meeting times for Eng 214</title>
<h1>Eng 214 Software Engineering for Non-programmers </h1>…

courseNum(window2) Ã token(window2,'Eng'), tripleton('Eng'), prevToken('Eng','214'), inTitle('214'), nextTok('Eng','214'), numeric('214'), tripleton('214'), nextTok('214','Software'), …

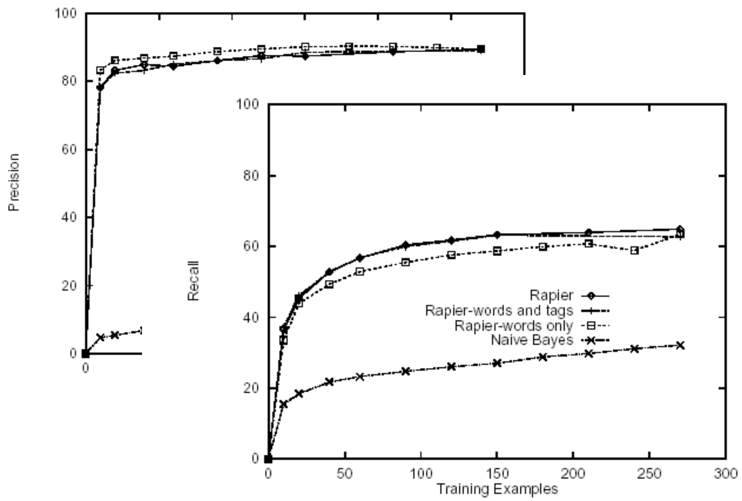
courseNum(X) Ã token(X,A), prevToken(A, B), inTitle(B), nextTok(A,C)), numeric(C), tripleton(C), nextTok(C,D), …

Common conditions carried over to generalization

Rapier: an alternative approach

- Combines top-down and bottom-up learning
  - Bottom-up to find common restrictions on **content**
  - Top-down greedy addition of restrictions on **context**
- Use of part-of-speech and semantic features (from WORDNET).
- Special "pattern-language" based on sequences of tokens, each of which satisfies one of a set of given constraints
  - < <tok2{'ate','hit'},POS2{'vb'}>, <tok2{'the'}>, <POS2{'nn'>>

Rapier: results – precision/recall



Rapier – results *vs.* SRV

System	stime		etime		loc		speaker	
	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
RAPIER	93.9	92.9	95.8	94.6	91.0	60.5	80.9	39.4
RAP-WT	96.5	95.3	94.9	94.4	91.0	61.5	79.0	40.0
RAP-W	96.5	95.9	96.8	96.6	90.0	54.8	76.9	29.1
NAIBAY	98.2	98.2	49.5	95.7	57.3	58.8	34.5	25.6
SRV	98.6	98.4	67.3	92.6	74.5	70.1	54.4	58.4
WHISK	86.2	100.0	85.0	87.2	83.6	55.4	52.6	11.1
WH-PR	96.2	100.0	89.5	87.2	93.8	36.1	0.0	0.0

Rule-learning approaches to sliding-window classification: Summary

- SRV, Rapier, and WHISK *[Soderland KDD '97]*
  - Representations for **classifiers** allow restriction of the **relationships** between tokens, etc
  - Representations are carefully chosen **subsets** of **even more powerful** representations based on logic programming (ILP and Prolog)
  - Use of these "heavyweight" representations is **complicated**, but seems to pay off in results
- Can simpler representations for classifiers work?

BWI: Learning to detect boundaries

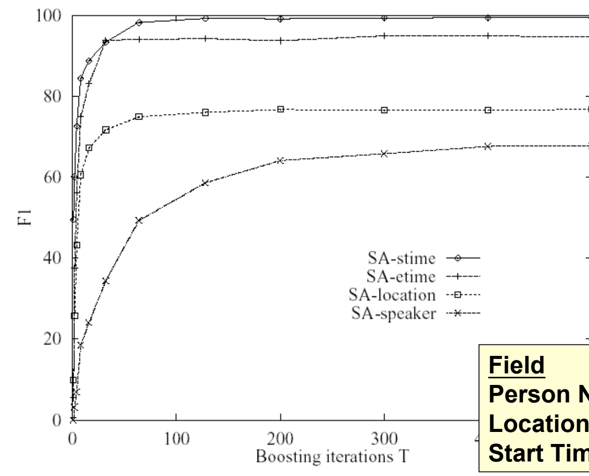
[Freitag & Kushmerick, AAAI 2000]

- Another formulation: learn **three** probabilistic classifiers:
  - *START(i)* = Prob(position *i* starts a field)
  - *END(j)* = Prob(position *j* ends a field)
  - *LEN(k)* = Prob(an extracted field has length *k*)
- Then score a possible extraction *(i,j)* by
  *START(i) * END(j) * LEN(j-i)*
- *LEN(k)* is estimated from a histogram

BWI: Learning to detect boundaries

- BWI uses **boosting** to find “detectors” for *START* and *END*
- Each weak detector has a *BEFORE* and *AFTER* pattern (on tokens before/after position *i).*
- Each “pattern” is a sequence of tokens and/or wildcards like: anyAlphabeticToken, anyToken, anyUpperCaseLetter, anyNumber, …
- Weak learner for “patterns” uses greedy search (+ lookahead) to repeatedly extend a pair of empty *BEFORE,AFTER* patterns

BWI: Learning to detect boundaries



Field	F1
Person Name:	**30%**
Location:	**61%**
Start Time:	**98%**

Problems with Sliding Windows and Boundary Finders

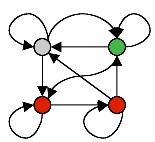
- Decisions in neighboring parts of the input are made independently from each other.
  - Naïve Bayes Sliding Window may predict a “seminar end time” before the “seminar start time”.
  - It is possible for two *overlapping* windows to both be above threshold.
  - In a Boundary-Finding system, left boundaries are laid down independently from right boundaries, and their pairing happens as a separate step.

Finite State Machines

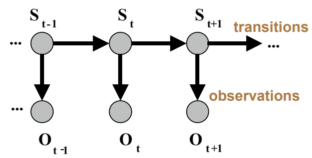
Hidden Markov Models

HMMs are the standard sequence modeling tool in genomics, music, speech, NLP, …

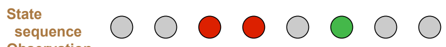
Finite state model



Graphical model



Generates:



o_1 o_2 o_3 o_4 o_5 o_6 o_7 o_8

$$P(\vec{s},\vec{o}) \propto \prod_{t=1}^{|\vec{o}|} P(s_t \mid s_{t-1}) P(o_t \mid s_t)$$

Parameters: for all states $S=\{s_1,s_2,\ldots\}$

- **Start state probabilities:** $P(s_t)$
- **Transition probabilities:** $P(s_t|s_{t-1})$
- **Observation (emission) probabilities:** $P(o_t|s_t)$ Usually a multinomial over atomic, fixed alphabet

Training:

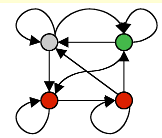
- **Maximize probability of training observations (w/ prior)**

IE with Hidden Markov Models

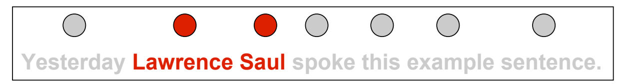
Given a sequence of observations:

Yesterday Lawrence Saul spoke this example sentence.

and a trained HMM:



Find the most likely state sequence: (Viterbi)



Any words said to be generated by the designated "person name" state extract as a person name:

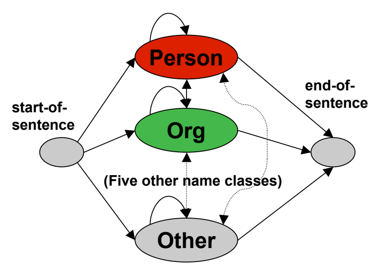
Person name: Lawrence Saul

HMMs for IE:
A richer model, with backoff

HMM Example: "Nymble"

[Bikel, et al 1998], [BBN "IdentiFinder"]

Task: Named Entity Extraction



Transition probabilities	Observation probabilities
$P(s_t \mid s_{t-1}, o_{t-1})$	$P(o_t \mid s_t, s_{t-1})$ or $P(o_t \mid s_t, o_{t-1})$
Back-off to:	**Back-off to:**
$P(s_t \mid s_{t-1})$	$P(o_t \mid s_t)$
$P(s_t)$	$P(o_t)$

Train on 450k words of news wire text.

Results:

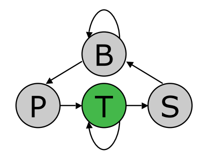
Case	Language	F1 .
Mixed	English	93%
Upper	English	91%
Mixed	Spanish	90%

Other examples of shrinkage for HMMs in IE: *[Freitag and McCallum '99]*

HMMs for IE: Augmented finite-state structures with linear interpolation

Simple HMM structure for IE

- 4 state types:
  - **B**ackground (generates words not of interest),
  - **T**arget (generates words to be extracted),
  - **P**refix (generates typical words preceding target)
  - **S**uffix (words typically following target)



- Properties:
  - Extracts one type of target (e.g. target = person name), we will build one model for each extracted type.
  - Models different Markov-order n-grams for different predicted state contexts.
  - even thought there are multiple states for “Background”, state-path given labels is unambiguous. Therefore model parameters can all be computed using counts from labeled training data

More rich prefix and suffix structures

- In order to represent more context, add more state structure to prefix, target and suffix.
- But now overfitting becomes more of a problem.

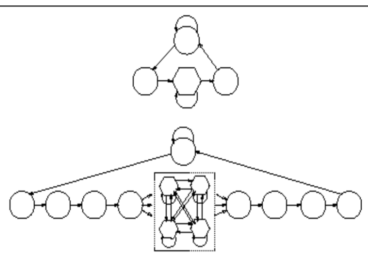
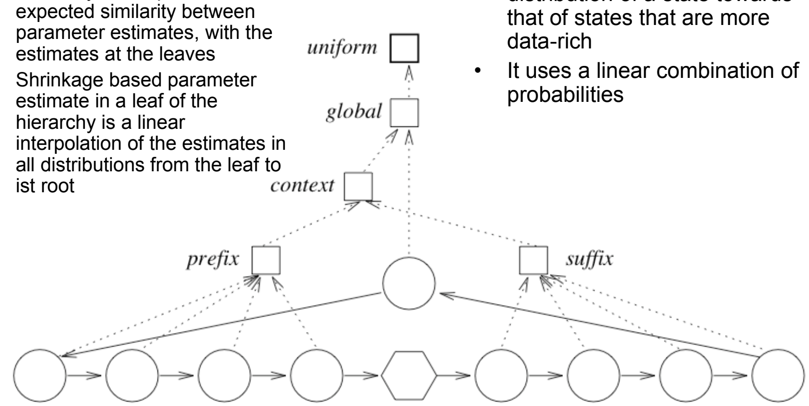


Figure 1: Two example HMM structures. Circle nodes represent non-target states; hexagon nodes represent target states.

Linear interpolation across states

- Is defined in terms of some hierarchy that represents the expected similarity between parameter estimates, with the estimates at the leaves
- Shrinkage based parameter estimate in a leaf of the hierarchy is a linear interpolation of the estimates in all distributions from the leaf to ist root



- Shrinkage smoothes the distribution of a state towards that of states that are more data-rich
- It uses a linear combination of probabilities

Evaluation of linear interpolation

- Data set of seminar announcements.

	speaker	*location*	*stime*	*etime*
None	0.513	0.735	0.991	0.814
Uniform	0.614	0.776	0.991	0.933
Global	0.711	0.839	0.991	0.595
Hier.	0.672	0.850	0.987	0.584

Table 4: Effect on F1 performance of different shrinkage configurations on four seminar announcement fields, given a topology with a window size of four and four parallel length-differentiated target paths.

IE with HMMs: Learning Finite State Structure

Information Extraction from Research Papers

References

Leslie Pack Kaelbling, Michael L. Littman and Andrew W. Moore. Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research, pages 237-285, May 1996.

Headers

Journal of Artificial Intelligence Research 4 (1996) 237-285 Submitted 9/95; published 5/96

Reinforcement Learning: A Survey

Leslie Pack Kaelbling LPK@CS.BROWN.EDU
Michael L. Littman MLITTMAN@CS.BROWN.EDU
Computer Science Department, Box 1910, Brown University
Providence, RI 02912-1910 USA

Andrew W. Moore AWM@CS.CMU.EDU
Smith Hall 221, Carnegie Mellon University, 5000 Forbes Avenue
Pittsburgh, PA 15213 USA

Abstract

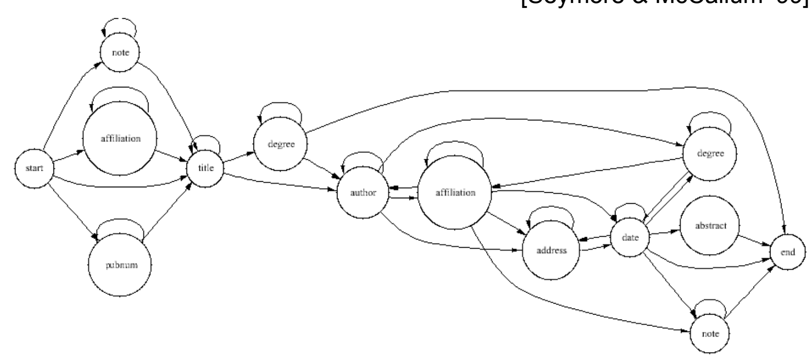
This paper surveys the field of reinforcement learning from a computer-science perspective. It is written to be accessible to researchers familiar with machine learning. Both the historical basis of the field and a broad selection of current work are summarized. Reinforcement learning is the problem faced by an agent that learns behavior through trial-and-error interactions with a dynamic environment. The work described here has a resemblance to work in psychology, but differs considerably in the details and in the use of the word "reinforcement." The paper discusses central issues of reinforcement learning, including trading off exploration and exploitation, establishing the foundations of the field via Markov decision theory, learning from delayed reinforcement, constructing empirical models to accelerate learning, making use of generalization and hierarchy, and coping with hidden state. It concludes with a survey of some implemented systems and an assessment of the practical utility of current methods for reinforcement learning.

1. Introduction

Reinforcement learning dates back to the early days of cybernetics and work in statistics,

Information Extraction with HMMs

[Seymore & McCallum '99]



Importance of HMM Topology

- Certain structures better capture the observed phenomena in the prefix, target and suffix sequences
- Building structures by hand does not scale to large corpora
- Human intuitions don't always correspond to structures that make the best use of HMM potential

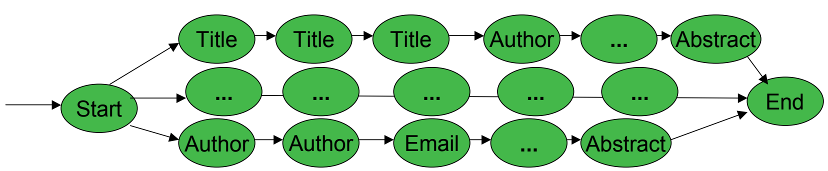
Structure Learning

Two approaches

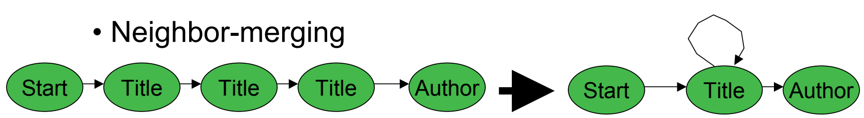
- Bayesian Model Merging
  - Neighbor-Merging
  - V-Merging
- Stochastic Optimization
  - Hill Climbing in the possible structure space by spiltting states and gauging performance on a validation set

Bayesian Model Merging

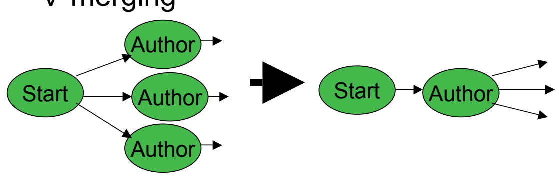
- Maximally Spesific Model



- Neighbor-merging



- V-merging

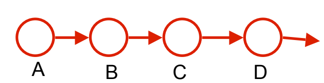


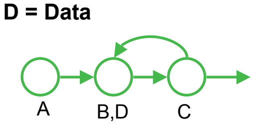
Bayesian Model Merging

- Iterates merging states until an optimal tradeoff between fit to the data and model size has been reached

P(M | D) ~ P(D | M) P(M)

M = Model
D = Data

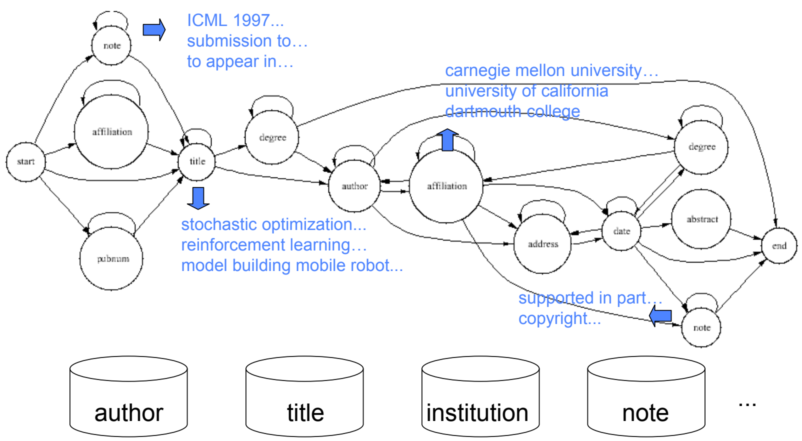




P(D | M) can be calculated with the Forward algorithm

P(M) model prior can be formulated to reflect a preference for smaller models

HMM Emissions



2 million words of BibTeX data from the Web

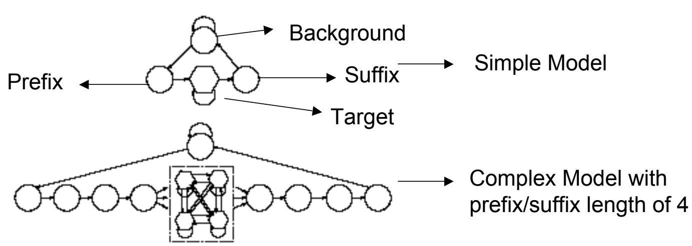
HMM Information Extraction Results

Per-word error rate

	Headers	References
One state/class Labeled data only	**0.095**	
Model Merging Labeled data only	**0.087 *(8% better)***	
One state/class +BibTeX data	**0.076 *(20% better)***	
Model Merging +BibTeX	**0.071 *(25% better)***	**0.066**

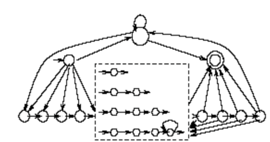
Stochastic Optimization

- Start with a simple model
- Perform hill-climbing in the space of possible structures
- Make several runs and take the average to avoid local optima



State Operations

- Lengthen a prefix
- Split a prefix
- Lengthen a suffix
- Split a suffix
- Lengthen a target string
- Split a target string
- Add a background state



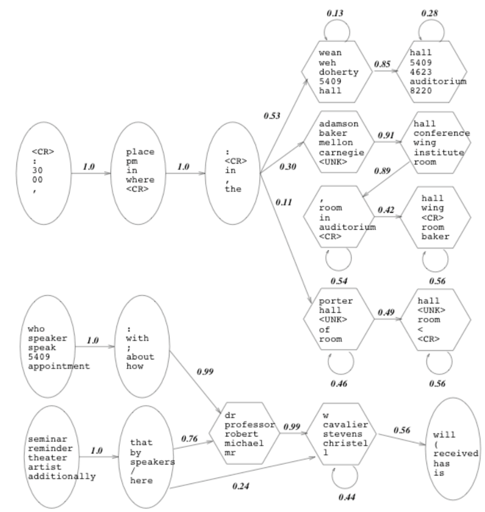
LearnStructure Algorithm

```
procedure LearnStructure(LabeledSet, Ops)
    ValidSet ← 1/3 of LabeledSet
    TrainSet ← LabeledSet − ValidSet
    CurModel ← the simple model
    Keepers ← {CurModel}
    I ← 0
    while I < 20 and CurModel has fewer than 25 states
        Candidates ← {M | M ∈ op(CurModel) ∧ op ∈ Ops}
        for M ∈ Candidates
            score(M) ← average of 3 runs trained on
                TrainSet and scored for F1 on ValidSet
        CurModel ← M ∈ Candidates with highest score
        Keepers ← Keepers ∪ {CurModel}
        I ← I + 1
    for M ∈ Keepers
        score(M) ← average F1 from
            3-fold cross-validation on LabeledSet
    return M ∈ Keepers with highest score
```

Part of Example Learned Structure

Locations

Speakers



Accuracy of Automatically-Learned Structures

	speaker	*location*	*acquired*	*dlramt*	*title*	*company*	*conf*	*deadline*	***Average***
Grown HMM	76.9	87.5	41.3	54.4	58.3	65.4	27.2	46.5	57.2
vs. SRV	+19.8	+16.0	+1.1	-1.6	—	—	—	—	+8.8
vs. Rapier	+23.9	+14.8	+12.5	+15.1	-11.7	+24.9	—	—	+13.3
vs. Simple HMM	+24.3	+5.6	+14.3	+5.6	+5.7	+11.1	+15.7	+6.7	+11.1
vs. Complex HMM	-2.1	+6.7	+7.5	-0.3	-0.3	+19.1	+0.0	-6.8	+3.0

Table 2: Difference in F1 performance between the HMM using a learned structure and other methods. The + numbers indicate how much better our Grown HMM did than the alternative method.

Limitations of HMM/CRF models

- HMM/CRF models have a **linear** structure
- Web documents have a **hierarchical** structure
  - Are we suffering by not modeling this structure more explicitly?
- How can one learn a **hierarchical** extraction model?
  - Coming up: STALKER, a hierarchical **wrapper-learner**
  - But first: how do we train wrapper-learners?

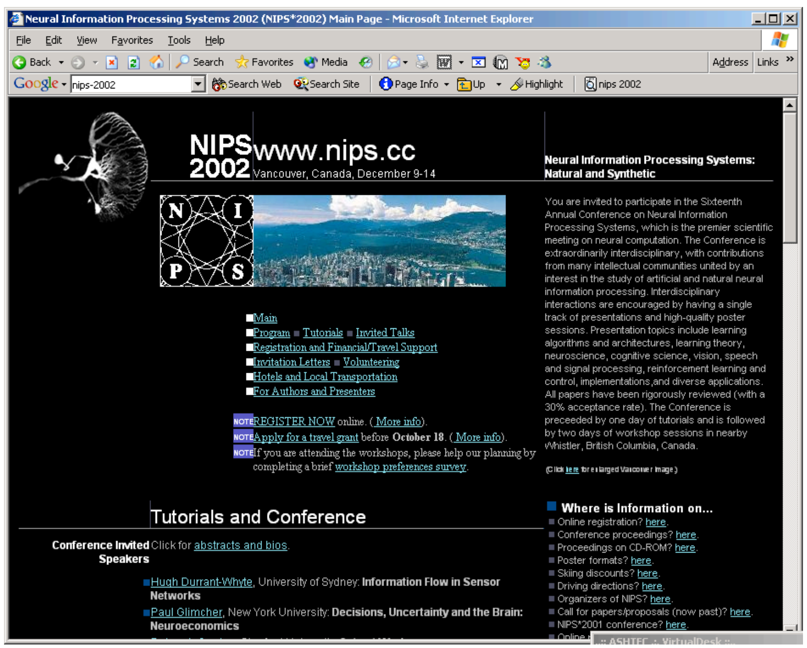
Tree-based Models

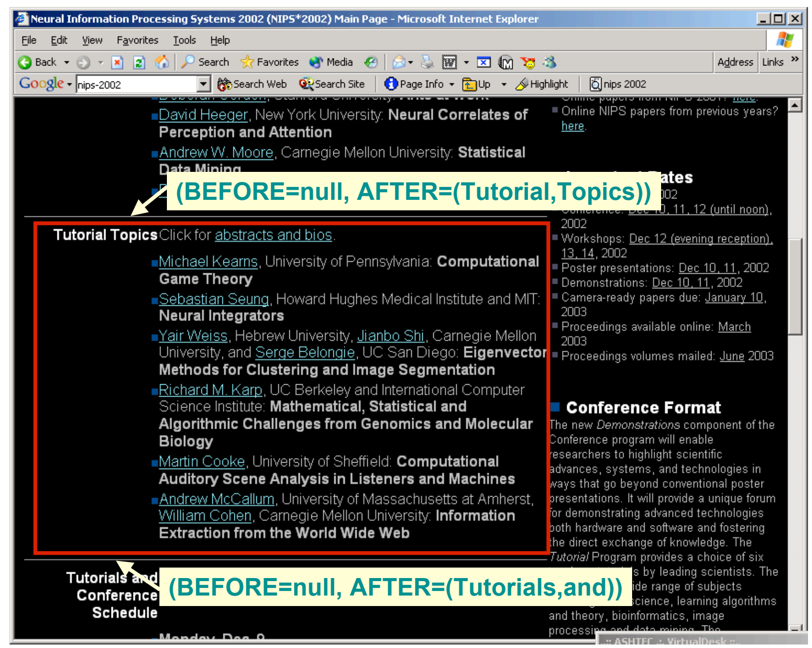
- Extracting from **one** web site
  - Use *site-specific* formatting information: e.g., “the JobTitle is a bold-faced paragraph in column 2”
  - For large well-structured sites, like parsing a **formal language**
- Extracting from **many** web sites:
  - Need general solutions to entity extraction, grouping into records, etc.
  - Primarily use *content* information
  - Must deal with a *wide range* of ways that users present data.
  - Analogous to parsing **natural language**
- Problems are **complementary:**
  - Site-dependent learning can collect training data for a site-independent learner
  - Site-dependent learning can boost accuracy of a site-independent learner on selected key sites

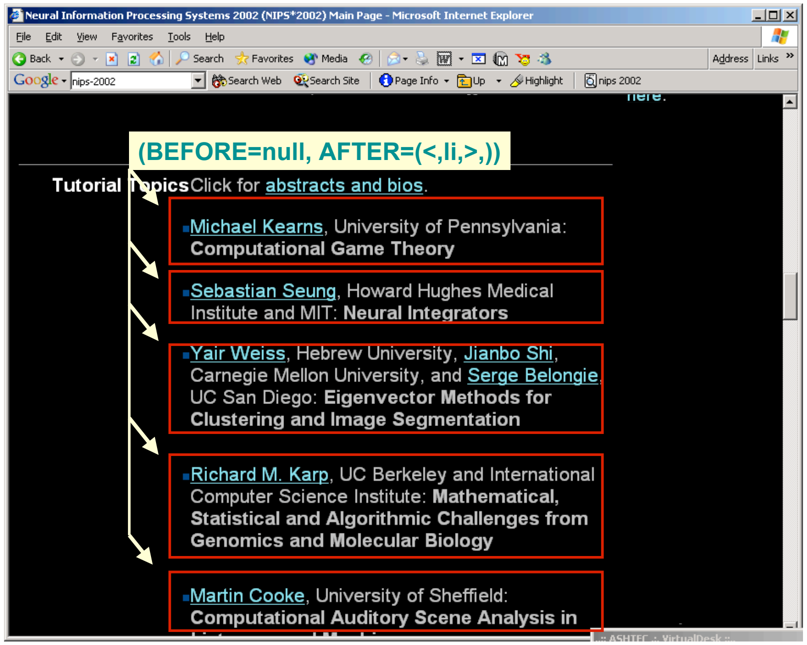
STALKER: Hierarchical boundary finding

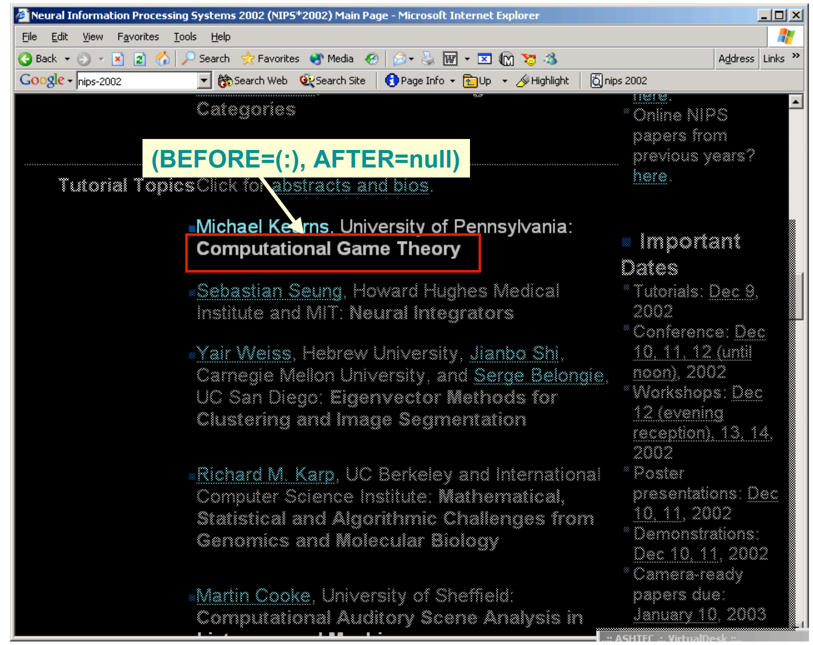
[Muslea,Minton & Knoblock 99]

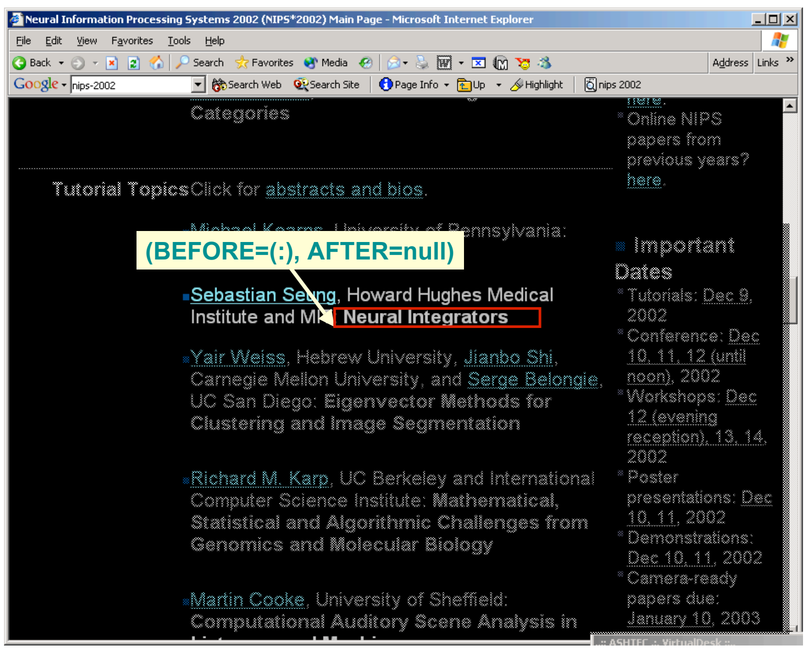
- Main idea:
  - To train a hierarchical extractor, pose a **series** of learning problems, one for each node in the hierarchy
  - At each stage, extraction is simplified by knowing about the "**context**."

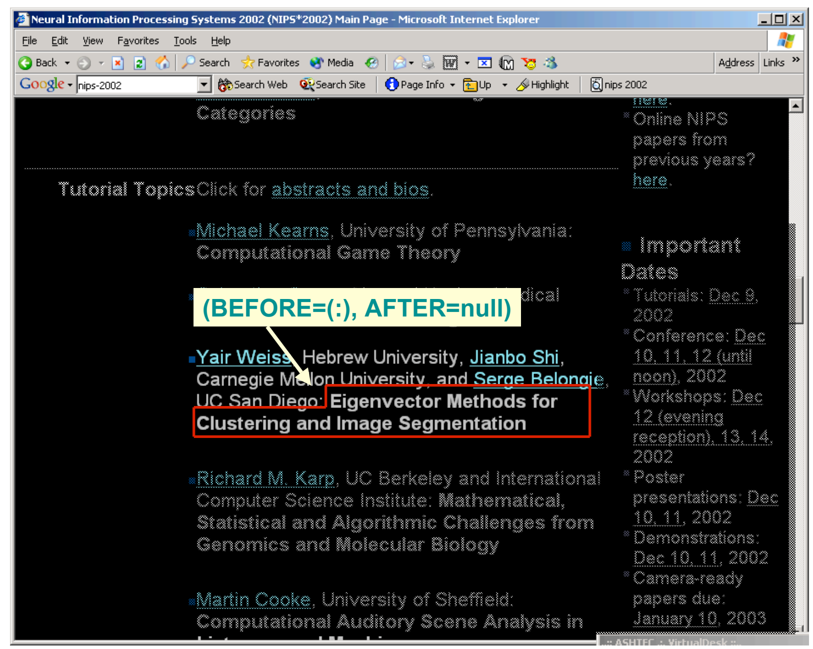




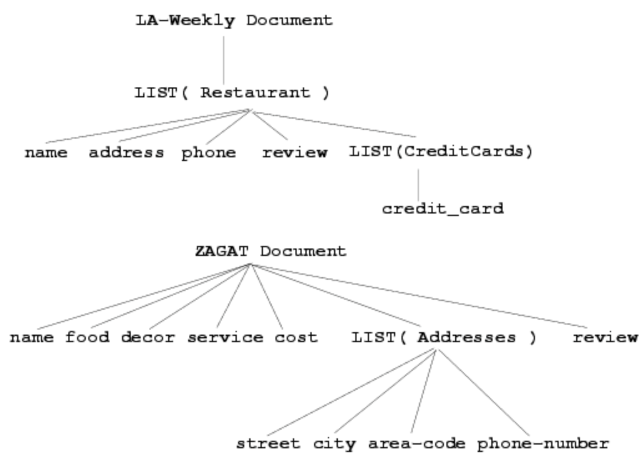

Neural Information Processing Systems 2002 (NIPS*2002) Main Page - Microsoft Internet Explorer
(BEFORE=null, AFTER=(<,li,>,))
Tutorial Topics Click for abstracts and bios.
Michael Kearns, University of Pennsylvania: Computational Game Theory
Sebastian Seung, Howard Hughes Medical Institute and MIT: Neural Integrators
Yair Weiss, Hebrew University, Jianbo Shi, Carnegie Mellon University, and Serge Belongie, UC San Diego: Eigenvector Methods for Clustering and Image Segmentation
Richard M. Karp, UC Berkeley and International Computer Science Institute: Mathematical, Statistical and Algorithmic Challenges from Genomics and Molecular Biology
Martin Cooke, University of Sheffield: Computational Auditory Scene Analysis in


Neural Information Processing Systems 2002 (NIPS*2002) Main Page - Microsoft Internet Explorer
(BEFORE=(:), AFTER=null)
Tutorial Topics Click for abstracts and bios.
Michael Kearns, University of Pennsylvania: Computational Game Theory
Important Dates


Neural Information Processing Systems 2002 (NIPS*2002) Main Page - Microsoft Internet Explorer
(BEFORE=(:), AFTER=null)
Sebastian Seung, Howard Hughes Medical Institute and MIT: Neural Integrators
Important Dates


Neural Information Processing Systems 2002 (NIPS*2002) Main Page - Microsoft Internet Explorer
(BEFORE=(:), AFTER=null)
Yair Weiss, Hebrew University, Jianbo Shi, Carnegie Mellon University, and Serge Belongie, UC San Diego: Eigenvector Methods for Clustering and Image Segmentation
Important Dates

Stalker: hierarchical decomposition of two web sites

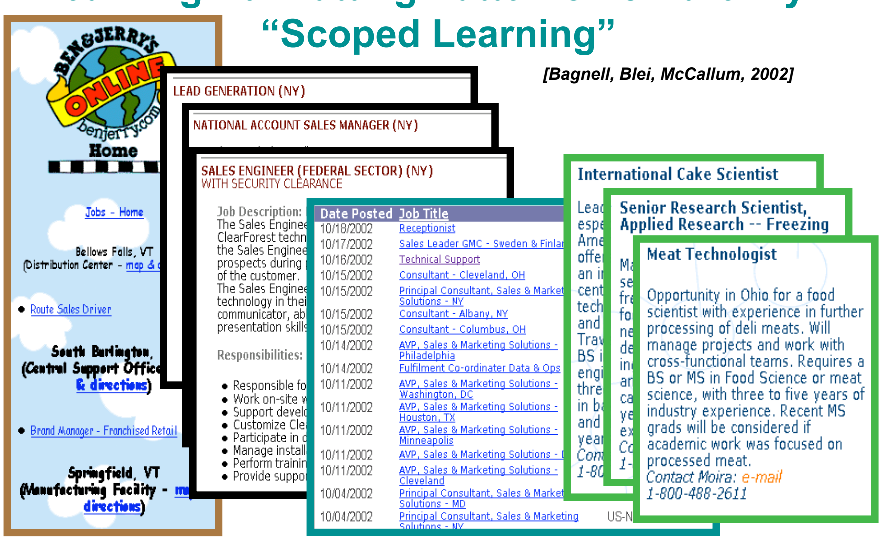


Stalker: summary and results

- Rule format:
  - "landmark automata" format for rules
    - E.g.: <a>W. Cohen CMU: Web IE
    - STALKER: BEGIN = SkipTo(<, /, a, >), SkipTo(:)
- Top-down rule learning algorithm
  - Carefully chosen ordering between types of rule specializations
- Very fast learning: e.g. 8 examples *vs. 274*
- **A lesson:** we often control the IE training data!

Learning Formatting Patterns "On the Fly": "Scoped Learning"

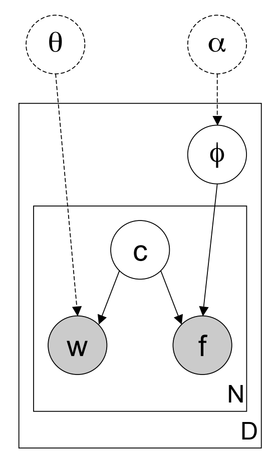
[Bagnell, Blei, McCallum, 2002]



Formatting is regular on each site, but there are too many different sites to wrap. Can we get the best of both worlds?

Scoped Learning Generative Model

1. For each of the D documents:
   a) Generate the multinomial formatting feature parameters ϕ from $p(\phi|\alpha)$
2. For each of the N words in the document:
   a) Generate the *n*th category c_n from $p(c_n)$.
   b) Generate the *n*th word (global feature) from $p(w_n|c_n,\theta)$
   c) Generate the *n*th formatting feature (local feature) from $p(f_n|c_n,\phi)$



$$p(\phi,\mathbf{c},\mathbf{w},\mathbf{f}) = p_\alpha(\phi)\prod_{n=1}^{N} p(c_n)p_\theta(w_n|c_n)p(f_n|c_n,\phi)$$

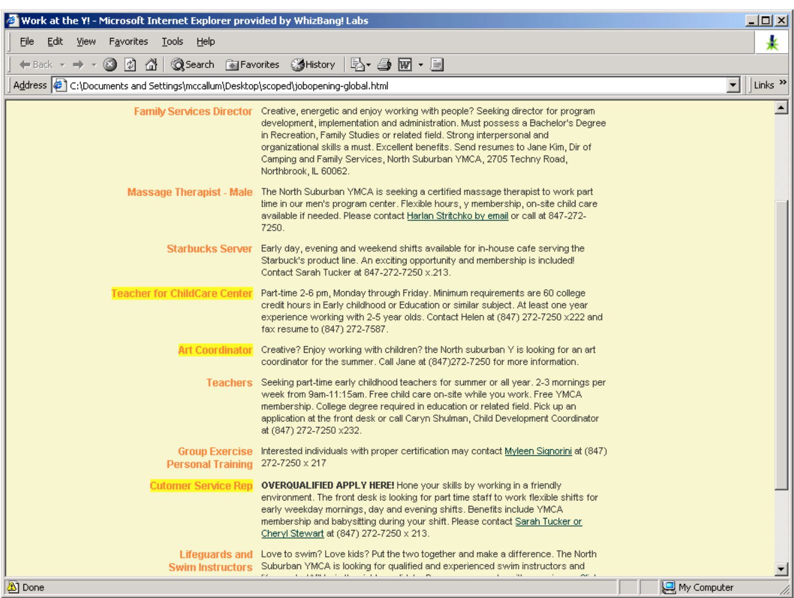
Inference

Given a new web page, we would like to classify each word resulting in **c** = {c_1, c_2,…, c_n}

$$p(\mathbf{c}|\mathbf{w},\mathbf{f}) = \frac{\int \prod_{n=1}^{N} p(w_n|c_n)p(f_n|c_n,\phi)p(c_n)p(\phi)d\phi}{\int \prod_{n=1}^{N} \sum_{c_n} p(w_n|c_n)p(f_n|c_n,\phi)p(c_n)p(\phi)d\phi}$$

This is not feasible to compute because of the integral and sum in the denominator. We experimented with two approximations:

- MAP point estimate of ϕ
- Variational inference



Global Extractor: Precision = 46%, Recall = 75%

MAP Point Estimate

If we approximate ϕ with a point estimate, $\hat{\phi}$, then the integral disappears and c decouples. We can then label each word with:

$$\hat{c}_n = \arg\max_{c_n} p(w_n|c_n)p(f_n|c_n,\hat{\phi})p(c_n)$$

A natural point estimate is the posterior mode: a maximum likelihood estimate for the local parameters given the document in question:

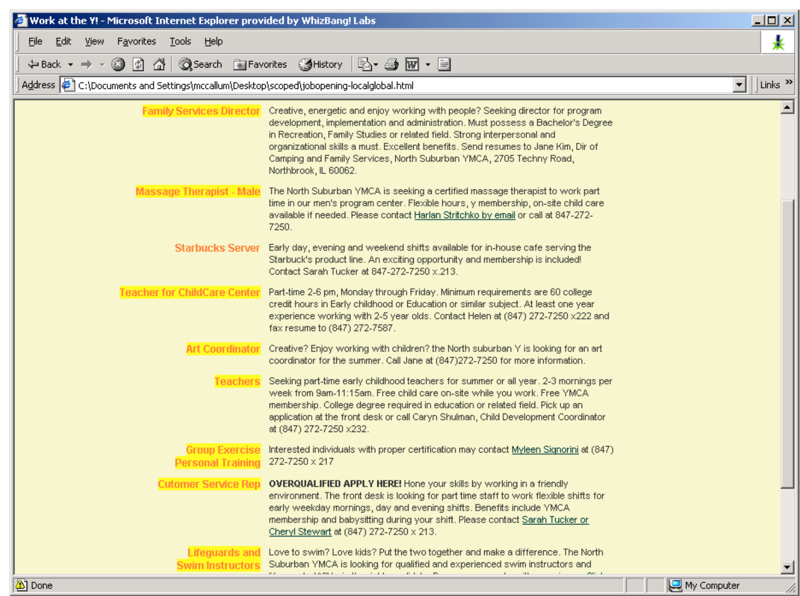
$$\hat{\phi} = \arg\max_{\phi} p(\phi|\mathbf{f},\mathbf{w})$$

E-step:

$$p^{(t+1)}(c_n|w_n,f_n;\phi) \propto p^{(t)}(f_n|c_n;\phi)p(w_n|c_n)p(c_n)$$

M-step:

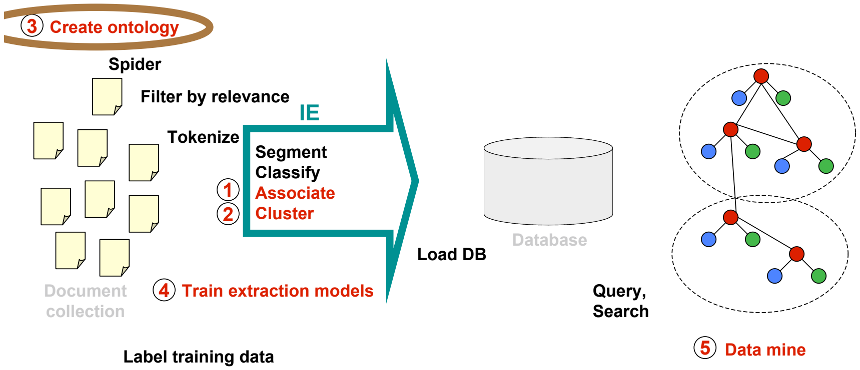
$$\hat{\phi}_{c,f} = p^{(t+1)}(f|c;\phi) \propto \sum_{\{n:c_n=c,f_n=f\}} p^{(t)}(c_n|f_n,w_n)$$



Scoped Learning Extractor: Precision = 58%, Recall = 75% ***Δ Error = -22%***

Broader View

Now touch on some other issues



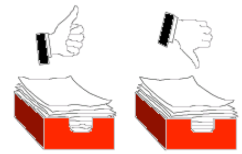
(3) Automatically Inducing an Ontology

[Riloff, '95]

Two inputs:

(1)

preclassified texts



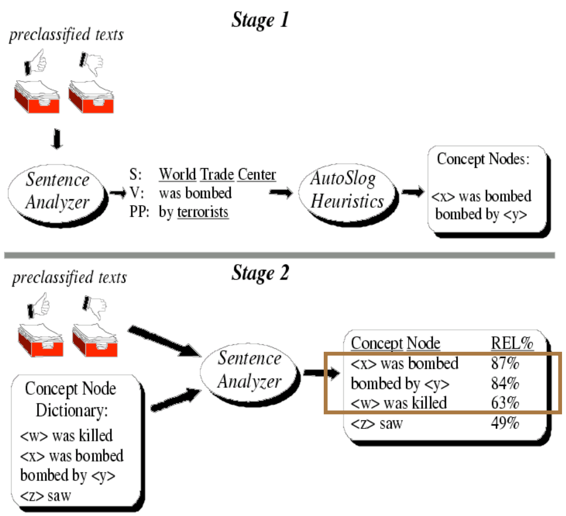
(2)

Heuristic "interesting" meta-patterns.

Linguistic Pattern	Example
1. <subject> active-verb	<perpetrator> bombed
2. <subject> active-verb direct-object[3]	<perpetrator> claimed responsibility
3. <subject> passive-verb	<victim> was murdered
4. <subject> verb infinitive	<perpetrator> attempted to kill
5. <subject> auxiliary noun	<victim> was victim
6. active-verb <direct-object>	bombed <target>
7. passive-verb <direct-object>[4]	killed <victim>
8. infinitive <direct-object>	to kill <victim>
9. verb infinitive <direct-object>	threatened to attack <target>
10. gerund <direct-object>	killing <victim>
11. noun auxiliary <direct-object>	fatality was <victim>
12. noun preposition <noun-phrase>	bomb against <target>
13. active-verb preposition <noun-phrase>	killed with <instrument>
14. passive-verb preposition <noun-phrase>	was aimed at <target>
15. infinitive preposition <noun-phrase>[3]	to fire at <victim>

(3) Automatically Inducing an Ontology

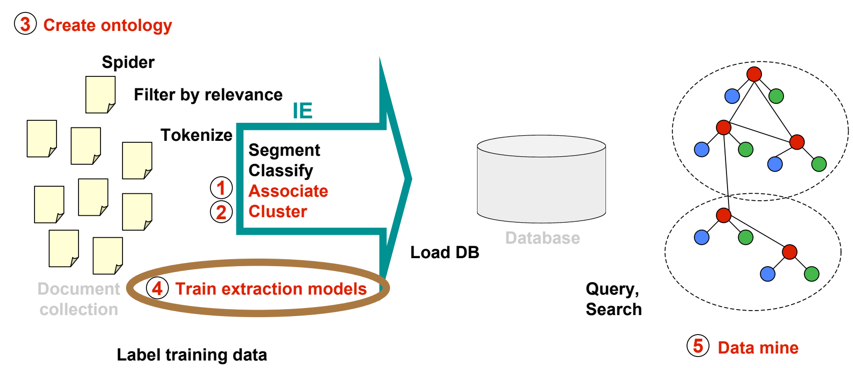
[Riloff, '95]



Subject/Verb/Object patterns that occur more often in the relevant documents than the irrelevant ones.

Broader View

Now touch on some other issues



(4) Training IE Models using Unlabeled Data

[Collins & Singer, 1999]

...says Mr. Cooper, a vice president of ...

NNP NNP appositive phrase, head=president

Use two independent sets of features:

Contents: full-string=*Mr._Cooper*, contains(*Mr.*), contains(*Cooper*)
Context: context-type=*appositive*, appositive-head=*president*

1. Start with just seven rules: and ~1M sentences of NYTimes

full-string=New_York	\ Location
fill-string=California	\ Location
full-string=U.S.	\ Location
contains(Mr.)	\ Person
contains(Incorporated)	\ Organization
full-string=Microsoft	\ Organization
full-string=I.B.M.	\ Organization

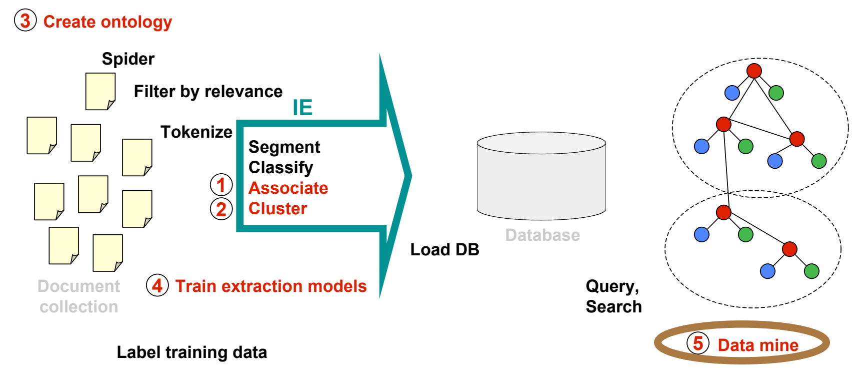
2. Alternately train & label using each feature set.

3. Obtain 83% accuracy at finding *person*, *location*, *organization* & *other* in appositives and prepositional phrases!

See also *[Brin 1998], [Riloff & Jones 1999]*

Broader View

Now touch on some other issues



(5) Data Mining: Working with IE Data

- Some special properties of IE data:
  - It is based on extracted text
  - It is “dirty”, (missing extraneous facts, improperly normalized entity names, etc.
  - May need cleaning before use
- What operations can be done on dirty, unnormalized databases?
  - Query it directly with a language that has “soft joins” across similar, but not identical keys. *[Cohen 1998]*
  - Construct features for learners *[Cohen 2000]*
  - Infer a “best” underlying clean database *[Cohen, Kautz, MacAllester, KDD2000]*

(5) Data Mining: Mutually supportive IE and Data Mining

[Nahm & Mooney, 2000]

Extract a large database
Learn rules to predict the value of each field from the other fields.
Use these rules to increase the accuracy of IE.

Example DB record

Filled Job Template
title: Senior DBMS Consultant
salary: Up to $55K
state: TX
city: Dallas
country: US
language: Powerbuilder, Progress, C, C++, Visual Basic
platform: UNIX, NT
application: SQL Server, Oracle
area: Electronic Commerce, Customer Service
required years of experience: 3
desired years of experience: 5
required degree: BS

Sample Learned Rules

platform:AIX & !application:Sybase & application:DB2
\application:Lotus Notes

language:C++ & language:C & application:Corba & title=SoftwareEngineer
\ platform:Windows

language:HTML & platform:WindowsNT & application:ActiveServerPages
\ area:Database

Language:Java & area:ActiveX & area:Graphics
\ area:Web