

# CS276B

Web Search and Mining  
Winter 2005

## Lecture 3

(includes slides borrowed from Andrew McCallum and Nick Kushmerick)

# Recap: Project and Practicum

- We hope you've been thinking about projects!
- Still time to revise and concretize plans over the next week, though...

# Plan for IE

- Today
  - Introduction to the IE problem
  - Wrappers
  - Wrapper Induction
  - Traditional NLP-based IE
- Second IE class
  - Probabilistic/machine learning methods for information extraction

# What is Information Extraction?

- First note this semantic slippage:
  - Information Retrieval *doesn't* retrieve **information**
  - You have an information need, but what you get back isn't *information* but *documents*, which you hope have the information
- Information extraction is *one* approach to going further for a special case:
  - There's some relation you're interested in
  - Your query is for elements of that relation
  - A limited form of natural language understanding
- But this is a common scenario...

# Extracting Corporate Information

For more information contact:

Street address: 10 Mageau Road, Suite 330  
City: Lexington  
State: MA  
Zip code: 02421-3112  
Telephone: 781-674-0000 (7)  
Fax: 781-674-0048 (777)  
Email: info@marketsoft.com  
SIC code: 7372 (Prepackaged software) (777)

People/Titles	Addresses	Company
Chris Evans - President & CEO	10 Mageau Road, Suite 330, Lexington MA 02421-3112	CEO, MarketSoft
Markus Sol		

Source: Whizbang! Labs/  
Andrew McCallum

E.g., information need: Who is the CEO of MarketSoft?

Google Search: lego fire engine

Results 1 - 99 of about 3 confirmed / 6,830 total results for lego fire engine. (0.79 seconds)

LEGO Micro Minis Rescue Fire Engine  
LEGO Vehicles Set

LEGO fire engine

## Commercial information...

A book, Not a toy

Title

Need this price

## Canonicalization: Product information

## Product information

## Product information

## Product info

- CNET markets this information
- How do they get most of it?
  - Sometimes data feeds
  - Phone calls
  - Typing

## It's difficult because of textual inconsistency: digital cameras

- Image Capture Device: 1.68 million pixel 1/2-inch CCD sensor
- Image Capture Device Total Pixels Approx. 3.34 million Effective Pixels Approx. 3.24 million
- Image sensor Total Pixels: Approx. 2.11 million-pixel
- Imaging sensor Total Pixels: Approx. 2.11 million, 1,688 (H) x 1,248 (V)
- CCD Total Pixels: Approx. 3,340,000 (2,140[H] x 1,560 [V])
  - Effective Pixels: Approx. 3,240,000 (2,088 [H] x 1,550 [V])
  - Recording Pixels: Approx. 3,145,000 (2,048 [H] x 1,536 [V])
- *These all came off the same manufacturer's website!!*
- And this is a very technical domain. Try sofa beds.



## Classified Advertisements (Real Estate)

### Background:

- Advertisements are plain text
- Lowest common denominator: only thing that 70+ newspapers with 20+ publishing systems can all handle

```
<ADNUM>2067206v1</ADNUM>
<DATE>March 02, 1998</DATE>
<ADTITLE>MADDINGTON
$89,000</ADTITLE>
<ADTEXT>
OPEN 1.00 - 1.45<BR>
U 11 / 10 BERTRAM ST<BR>
NEW TO MARKET Beautiful<BR>
3 brm freestanding<BR>
villa, close to shops & bus<BR>
Owner moved to Melbourne<BR>
ideally suit 1st home
buyer,<BR>
investor & 55 and over.<BR>
Brian Hazel den 0418 958 996<BR>
R WHITE LEEMING 9332 3477
</ADTEXT>
```



## Why doesn't text search (IR) work?

What you search for in real estate advertisements:

- Towns. You might think easy, but:
  - Real estate agents: Coldwell Banker, Mosman
  - Phrases: Only 45 minutes from Parramatta
  - Multiple property ads have different towns
- Money: want a range not a textual match
  - Multiple amounts: was \$155K, now \$145K
  - Variations: offers in the high 700s [but not rents for \$270]
- Bedrooms: similar issues (br, bdr, beds, B/R)

## Task: Information Extraction

- Information extraction systems
  - Find and understand the limited relevant parts of texts
    - Clear, factual information (*who did what to whom when?*)
  - Produce a structured representation of the relevant information: *relations* (in the DB sense)
  - Combine knowledge about language and a domain
  - Automatically extract the desired information
- E.g.
  - Gathering earnings, profits, board members, etc. from company reports
  - Learn drug-gene product interactions from medical research literature
  - "Smart Tags" (Microsoft) inside documents

## Aside: What about XML?

- Don't XML, RDF, OIL, SHOE, DAML, XSchema, ... obviate the need for information extraction?!?!?
- Yes:
  - IE is sometimes used to "reverse engineer" HTML database interfaces; extraction would be much simpler if XML were exported instead of HTML.
  - Ontology-aware editors will make it easier to enrich content with metadata.
- No:
  - Terabytes of legacy HTML.
  - Data consumers forced to accept ontological decisions of data providers (eg, <NAME>John Smith</NAME> vs. <NAME first="John" last="Smith"/>).
  - A lot of these pages are PR aimed at humans
  - Will you annotate every email you send? Every memo you write? Every photograph you scan?

## "Wrappers"

- If we think of things from the database point of view
  - We want to be able to database-style queries
  - But we have data in some horrid textual form/content management system that doesn't allow such querying
  - We need to "wrap" the data in a component that understands database-style querying
    - Hence the term "wrappers"
- Many people have "wrapped" many web sites
  - Commonly something like a Perl script
  - Often easy to do as a one-off
- But handcoding wrappers in Perl isn't very viable
  - Sites are numerous, and their surface structure mutates rapidly (around 10% failures each month)

## Amazon Book Description

```
.....
</td></tr>
</table>
<b class="sans">The Age of Spiritual Machines : When Computers Exceed Human Intelligence</b><br>
<font face=verdana,arial,Helvetica size=1>
by <a href="/exec/obidos/search-handle-url/index=books&field-author=
Kurzweil%2C%20Ray/002-6235079-4593641">
Ray Kurzweil</a><br>
</font>
<br>
<a href="http://images.amazon.com/images/P/0140282025.01.LZZZZZZZ.jpg">
</a>
<font face=verdana,arial,Helvetica size=1>
<span class="small">
<span class="small">
<b>List Price:</b> <span class=listprice>$14.95</span><br>
<b>Our Price:</b> <font color=#990000>$11.96</font><br>
<b>You Save:</b> <font color=#990000><b>$2.99 </b>
(20%)</font><br>
</span>
<p><b>...
```

## Extracted Book Template

Title: **The Age of Spiritual Machines :  
When Computers Exceed Human Intelligence**  
Author: **Ray Kurzweil**  
List-Price: **\$14.95**  
Price: **\$11.96**  
:  
:

## Template Types

- Slots in template typically filled by a substring from the document.
- Some slots may have a fixed set of pre-specified possible fillers that may not occur in the text itself.
  - Terrorist act: threatened, attempted, accomplished.
  - Job type: clerical, service, custodial, etc.
  - Company type: SEC code
- Some slots may allow multiple fillers.
  - Programming language
- Some domains may allow multiple extracted templates per document.
  - Multiple apartment listings in one ad

## Wrapper tool-kits

- Wrapper toolkits: Specialized programming environments for writing & debugging wrappers by hand
- Ugh! The links to examples I used in 2003 are all dead now ... here's one I found:
  - <http://www.cc.gatech.edu/projects/disl/XWRAPelite/elite-home.html>
- Aging Examples
  - World Wide Web Wrapper Factory (W4F)
  - Java Extraction & Dissemination of Information (JEDI)
  - Junglee Corporation
  - Survey:  
<http://www.netobjectdays.org/pdf/02/papers/node/0188.pdf>

## Task: Wrapper Induction

- Learning wrappers is **wrapper induction**
  - Sometimes, the relations are structural.
    - Web pages generated by a database.
    - Tables, lists, etc.
  - Can't computers automatically learn the patterns a human wrapper-writer would use?
  - Wrapper induction is usually regular relations which can be expressed by the *structure* of the document:
    - the item in bold in the 3<sup>rd</sup> column of the table is the price
- Wrapper induction techniques can also learn:
  - If there is a page about a research project X and there is a link near the word 'people' to a page that is about a person Y then Y is a member of the project X.
    - [e.g, Tom Mitchell's Web->KB project]

## Wrappers: Simple Extraction Patterns

- Specify an item to extract for a slot using a regular expression pattern.
  - Price pattern: "**\b\\$\d+(\.\d{2})?\b**"
- May require preceding (pre-filler) pattern to identify proper context.
  - Amazon list price:
    - Pre-filler pattern: "<b>List Price:</b> <span class=listprice>"
    - Filler pattern: "\\$\d+(\.\d{2})?\b"
- May require succeeding (post-filler) pattern to identify the end of the filler.
  - Amazon list price:
    - Pre-filler pattern: "<b>List Price:</b> <span class=listprice>"
    - Filler pattern: "+"
    - Post-filler pattern: "</span>"

## Simple Template Extraction

- Extract slots in order, starting the search for the filler of the  $n+1$  slot where the filler for the  $n$ th slot ended. Assumes slots always in a fixed order.
  - Title
  - Author
  - List price
  - ...
- Make patterns specific enough to identify each filler always starting from the beginning of the document.

## Pre-Specified Filler Extraction

- If a slot has a fixed set of pre-specified possible fillers, text categorization can be used to fill the slot.
  - Job category
  - Company type
- Treat each of the possible values of the slot as a category, and classify the entire document to determine the correct filler.

## Wrapper induction

Highly regular source documents



Relatively simple extraction patterns



Efficient learning algorithm

- Writing accurate patterns for each slot for each domain (e.g. each web site) requires laborious software engineering.
- Alternative is to use machine learning:
  - Build a training set of documents paired with human-produced filled extraction templates.
  - Learn extraction patterns for each slot using an appropriate machine learning algorithm.

## Wrapper induction: Delimiter-based extraction



```
<HTML><TITLE>Some Country Codes</TITLE>
<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>
</BODY></HTML>
```

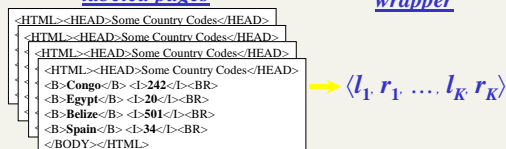


Use **<B>**, **</B>**, **<I>**, **</I>** for extraction

## Learning LR wrappers

labeled pages

wrapper



**Example:** Find 4 strings  
 $\langle \text{<B>, </B>, <I>, </I>} \rangle$   
 $\langle l_1, r_1, l_2, r_2 \rangle$

## LR: Finding $r_1$

```
<HTML><TITLE>Some Country Codes</TITLE>
<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>
</BODY></HTML>
```

$r_1$  can be any prefix  
eg **<B>**

## LR: Finding $l_1$ , $l_2$ and $r_2$

```
<HTML><TITLE>Some Country Codes</TITLE>
<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>
</BODY></HTML>
```

$r_2$  can be any prefix  
eg <I>

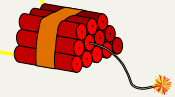
$l_2$  can be any suffix  
eg <I>

$l_1$  can be any suffix  
eg <B>

## A problem with LR wrappers

### Distracting text in head and tail

```
<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>
```



## One (of many) solutions: HLRT

### Ignore page's head and tail

```
<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>
```

start of tail



Head-Left-Right-Tail wrappers

## More sophisticated wrappers

- LR and HLRT wrappers are extremely simple
  - Though applicable to many tabular patterns
- Recent wrapper induction research has explored more expressive wrapper classes [Muslea et al, Agents-98; Hsu et al, JIS-98; Kushmerick, AAAI-1999; Cohen, AAAI-1999; Minton et al, AAAI-2000]
  - Disjunctive delimiters
  - Multiple attribute orderings
  - Missing attributes
  - Multiple-valued attributes
  - Hierarchically nested data
  - Wrapper verification and maintenance

## Boosted wrapper induction

- Wrapper induction is only ideal for rigidly-structured machine-generated HTML...
- ... or is it?!
- Can we use simple documents to extract from natural language documents?

```
... Name: Dr. Jeffrey D. Hermes ...
... Who: Professor Manfred Paul ...
... will be given by Dr. R. J. Pangborn ...

... Ms. Scott will be speaking ...
... Karen Shriver, Dept. of ...
... Maria Klawe, University of ...
```

## BWI: The basic idea

- Learn "wrapper-like" patterns for texts  
pattern = exact token sequence
- Learn many such "weak" patterns
- Combine with *boosting* to build "strong" ensemble pattern
  - Boosting* is a popular recent machine learning method where many weak learners are combined
- Demo: <http://www.smi.ucd.ie/bwi>
- Not all natural text is sufficiently regular for exact string matching to work well!!

## Natural Language Processing-based Information Extraction

- If extracting from automatically generated web pages, simple regex patterns usually work.
- If extracting from more natural, unstructured, human-written text, some NLP may help.
  - Part-of-speech (POS) tagging
    - Mark each word as a noun, verb, preposition, etc.
  - Syntactic parsing
    - Identify phrases: NP, VP, PP
  - Semantic word categories (e.g. from WordNet)
    - KILL: kill, murder, assassinate, strangle, suffocate
- Extraction patterns can use POS or phrase tags.
  - Crime victim:
    - Prefiller: [POS: V, Hypernym: KILL]
    - Filler: [Phrase: NP]

## MUC: the NLP genesis of IE

- DARPA funded significant efforts in IE in the early to mid 1990's.
- Message Understanding Conference (MUC) was an annual event/competition where results were presented.
- Focused on extracting information from news articles:
  - Terrorist events
  - Industrial joint ventures
  - Company management changes
- Information extraction is of particular interest to the intelligence community

### Example of IE from FASTUS (1993)

Bridgestone Sports Co. said Friday it had set up a joint venture in Taiwan with a local concern and a Japanese trading house to produce golf clubs to be supplied to Japan.

The joint venture, Bridgestone Sports Taiwan Co., capitalized at 20 million new Taiwan dollars, will start production in January 1990 with production of 20,000 iron and "metal wood" clubs a month.

#### TIE-UP-1

Relationship: TIE-UP

Entities: "Bridgestone Sport Co."

"a local concern"

"a Japanese trading house"

Joint Venture Company:

"Bridgestone Sports Taiwan Co."

Activity: ACTIVITY-1

Amount: NT\$200000000

#### ACTIVITY-1

Activity: PRODUCTION

Company:

"Bridgestone Sports Taiwan Co."

Product:

"iron and 'metal wood' clubs"

Start Date:

DURING: January 1990

### Example of IE: FASTUS(1993)

Bridgestone Sports Co. said Friday it had set up a joint venture in Taiwan with a local concern and a Japanese trading house to produce golf clubs to be supplied to Japan.

The joint venture, Bridgestone Sports Taiwan Co., capitalized at 20 million new Taiwan dollars, will start production in January 1990 with production of 20,000 iron and "metal wood" clubs a month.

#### TIE-UP-1

Relationship: TIE-UP

Entities: "Bridgestone Sport Co."

"a local concern"

"a Japanese trading house"

Joint Venture Company:

"Bridgestone Sports Taiwan Co."

Activity: ACTIVITY-1

Amount: NT\$200000000

#### ACTIVITY-1

Activity: PRODUCTION

Company:

"Bridgestone Sports Taiwan Co."

Product:

"iron and 'metal wood' clubs"

Start Date:

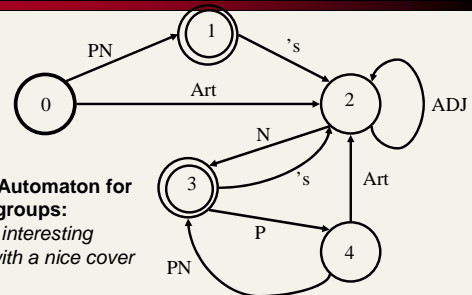
DURING: January 1990

## FASTUS

Based on finite state automata (FSA) transductions

- |  |   |
|--|---|
| <p>set up<br/>new Taiwan dollars</p> <p>a Japanese trading house<br/>had set up</p> <p>production of<br/>20,000 iron and<br/>metal wood clubs</p> <p>[company]<br/>[set up]<br/>[Joint-Venture]<br/>with<br/>[company]</p> | <ol style="list-style-type: none"> <li>1. Complex Words:<br/>Recognition of multi-words and proper names</li> <li>2. Basic Phrases:<br/>Simple noun groups, verb groups and particles</li> <li>3. Complex phrases:<br/>Complex noun groups and verb groups</li> <li>4. Domain Events:<br/>Patterns for events of interest to the application<br/>Basic templates are to be built.</li> <li>5. Merging Structures:<br/>Templates from different parts of the texts are merged if they provide information about the same entity or event.</li> </ol> |
|--|---|

## Grep++ = Cascaded grepping



Finite Automaton for Noun groups:  
John's interesting book with a nice cover

## Rule-based Extraction Examples

Determining which person holds what office in what organization

- [person] , [office] of [org]
  - Vuk Draskovic, leader of the Serbian Renewal Movement
- [org] (named, appointed, etc.) [person] P [office]
  - NATO appointed Wesley Clark as Commander in Chief

Determining where an organization is located

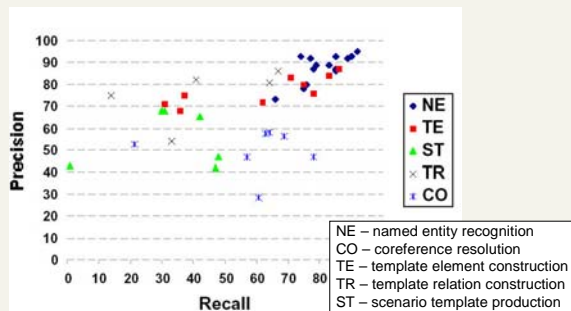
- [org] in [loc]
  - NATO headquarters in Brussels
- [org] [loc] (*division, branch, headquarters, etc.*)
  - KFOR Kosovo headquarters

## Evaluating IE Accuracy

- Always evaluate performance on independent, manually-annotated test data not used during system development.
- Template Measure for each test document:
  - Total number of correct extractions in the solution template:  $N$
  - Total number of slot/value pairs extracted by the system:  $E$
  - Number of extracted slot/value pairs that are correct (i.e. in the solution template):  $C$
- Compute average value of metrics adapted from IR:
  - Recall =  $C/N$
  - Precision =  $C/E$
  - F-Measure = Harmonic mean of recall and precision

Note subtle difference

## MUC Information Extraction: State of the Art c. 1997



## Summary and prelude

- We've looked at the "fragment extraction" task. Future?
  - Top-down semantic constraints (as well as syntax)?
  - Unified framework for extraction from regular & natural text? (BWI is one tiny step; Webfoot [Soderland 1999] is another.)
- Beyond fragment extraction:
  - Anaphora resolution, discourse processing, ...
  - Fragment extraction is good enough for many Web information services!
- Next time:
  - Learning methods for information extraction

## Three generations of IE systems

- Hand-Built Systems – Knowledge Engineering [1980s– ]
  - Rules written by hand
  - Require experts who understand both the systems and the domain
  - Iterative guess-test-tweak-repeat cycle
- Automatic, Trainable Rule-Extraction Systems [1990s– ]
  - Rules discovered automatically using predefined templates, using methods like ILP
  - Require huge, labeled corpora (effort is just moved!)
- Machine Learning (Sequence) Models [1997 – ]
  - One decodes a statistical model that classifies the words of the text, using HMMs, random fields or statistical parsers
  - Learning usually supervised; may be partially unsupervised

## Basic IE References

- Douglas E. Appelt and David Israel. 1999. Introduction to Information Extraction Technology. IJCAI 1999 Tutorial. <http://www.ai.sri.com/~appelt/ie-tutorial/>
- Kushmerick, Weld, Doorenbos: **Wrapper Induction for Information Extraction**, IJCAI 1997. <http://www.cs.ucd.ie/staff/nick/>
- Stephen Soderland: Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning* 34(1-3): 233-272 (1999)