CS276B
Text Information Retrieval, Mining, and Exploitation

Lecture 6
Information Extraction I
Jan 28, 2003

(includes slides borrowed from Oren Etzioni, Andrew McCallum, Nick Kushmerick, BBN, and Ray Mooney)

Product information

- CNET markets this information
- How do they get most of it?
  - 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[1H] 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

<ADNR>20072061</ADNR>
<Date>March 02, 1998</Date>
<ADTTL>Wadlington $89,000</ADTTL>
<ADTEXT>
OPEN 1.00 - 1.45pm
U 11 / 10 Bertram St<br>
NEW TO MARKET BEAUTIFUL<br>
3 bmr Freestanding villa, close to shops & bus<br>
Owner moved to Melbourne<br>
Ideally suit 1st home buyer, &RR<br>
Investor & SS and over.&RR<br>
BRIAN MAITLAND 0418 958 996 &RR<br>
WHITE LEEMING 9332 3477</ADTEXT>
Why doesn’t text search (IR) work?

What you search for in real estate advertisements:
- Suburbs. You might think easy, but:
  - Real estate agents: Coldwell Banker, Mosman
  - Phrases: Only 45 minutes from Parramatta
  - Multiple property ads have different suburbs
- 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)

Extracting Job Openings from the Web

Knowledge Extraction Vision

Task: Information Extraction

- Goal: being able to answer semantic queries (a.k.a. “database queries”) using “unstructured” natural language sources
- Identify specific pieces of information in an unstructured or semi-structured textual document.
- Transform this unstructured information into structured relations in a database/ontology.
- Suppositions:
  - A lot of information that could be represented in a structured semantically clear format isn’t
  - It may be costly, not desired, or not in one’s control (screen scraping) to change this.
Other applications of IE Systems

- Job resumes: BurningGlass, Mohomine
- Seminar announcements
- Continuing education courses info from the web
- Molecular biology information from MEDLINE, e.g., Extracting gene drug interactions from biomed texts
- Summarizing medical patient records by extracting diagnoses, symptoms, physical findings, test results.
- Gathering earnings, profits, board members, etc. [corporate information] from web, company reports
- Verification of construction industry specifications documents (are the quantities correct/reasonable?)
- Extraction of political/economic/business changes from newspaper articles

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 used 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 (e.g., <NAME> John Smith</NAME> vs. <NAME first="John" last="Smith"/>).
  - Will you annotate every email you send? Every memo you write? Every photograph you scan?

Task: Wrapper Induction

- Wrapper Induction
  - Sometimes, the relations are structural:
    - Web pages generated by a database.
    - Tables, lists, etc.
  - Wrapper indiction is usually regular relations which can be expressed by the structure of the document:
    - the item in bold in the 3rd column of the table is the price
  - Handcoding a wrapper in Perl isn’t very viable
    - sites are numerous, and their surface structure mutates rapidly (around 10% failures each month)
  - 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-vR project]

Amazon Book Description

- Title: The Age of Spiritual Machines: When Computers Exceed Human Intelligence
- Author: Ray Kurzweil
- List-Price: $14.95
- Price: $11.96
- You Save: $2.99 (20%)
Wrappers:
Simple Extraction Patterns

- Specify an item to extract for a slot using a regular expression pattern.
- Price pattern: "\.\d+\$r\(,\\d{2}\)\$/b"
- May require preceding (pre-filler) pattern to identify proper context.
  - Amazon list price:
    - Pre-filler pattern: "<dl-list price="<pre class="price""
    - Filler pattern: ":0"
- May require succeeding (post-filler) pattern to identify the end of the filler.
  - Amazon list price:
    - Pre-filler pattern: "<dl-list price="/b"<pre class="price""
    - Filler pattern: ":0"
    - Post-filler pattern: ":/"

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 tool-kits

- Wrapper toolkits: Specialized programming environments for writing & debugging wrappers by hand
- Examples
  - World Wide Web Wrapper Factory (W4F) [db.cis.upenn.edu/W4F]
  - Java Extraction & Dissemination of Information (JEDI) [www.darmstadt.gmd.de/oasys/projects/jedi]
  - Junglee Corporation

Wrapper induction:
Delimiter-based extraction

- Use `<B>`, `<I>`, `<DT>`, `<I>` for extraction
Learning LR wrappers

\[ \text{labeled pages} \rightarrow \text{wrapper} \]

\((l_1, r_1, l_2, r_2, \ldots, l_K, r_K)\)

**Example:** Find 4 strings

\(<\text{B}>, \text{<B>}, \text{<B>}, \text{<B>}\> \quad (l_1, r_1, l_2, r_2)

LR: Finding \(r_1\)

\(<\text{HTML}>\text{TITLE:Some Country Codes}</\text{TITLE}>\)
\(<\text{B> Congo</B> \text{<B> 242</B> \text{<B> 4</B>}/B><BR>\)
\(<\text{B> Egypt</B> \text{<B> 20</B> \text{<B> 4</B>}/B><BR>\)
\(<\text{B> Belize</B> \text{<B> 501</B> \text{<B> 4</B>}/B><BR>\)
\(<\text{B> Spain</B> \text{<B> 34</B> \text{<B> 4</B>}/B><BR>\)
\(<\text{BODY}</B> </B>/</B>/</B>/</HTML>\)

\(r_1\) can be any prefix
\(\text{eg } \text{<B>}\)

LR: Finding \(l_1, l_2\) and \(r_2\)

\(<\text{HTML}>\text{TITLE:Some Country Codes}</\text{TITLE}>\)
\(<\text{B> Congo</B> \text{<B> 242</B> \text{<B> 4</B>}/B><BR>\)
\(<\text{B> Egypt</B> \text{<B> 20</B> \text{<B> 4</B>}/B><BR>\)
\(<\text{B> Belize</B> \text{<B> 501</B> \text{<B> 4</B>}/B><BR>\)
\(<\text{B> Spain</B> \text{<B> 34</B> \text{<B> 4</B>}/B><BR>\)
\(<\text{BODY}</B> </B>/</B>/</B>/</HTML>\)

\(r_2\) can be any prefix
\(\text{eg } \text{<B>}\)

\(l_2\) can be any suffix
\(\text{eg } \text{<B>}\)

\(l_1\) can be any suffix
\(\text{eg } \text{<B>}\)

A problem with LR wrappers

**Distracting text in head and tail**

\(<\text{HTML}>\text{TITLE:Some Country Codes}</\text{TITLE}>\)
\(<\text{BODY}</B> </B>/</B>/</B>/</HTML>\)

More sophisticated wrappers

- LR and HLRT wrappers are extremely simple (though useful for ~ 2/3 of real Web sites)
- 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

One (of many) solutions: HLRT

\(<\text{HTML}>\text{TITLE:Some Country Codes}</\text{TITLE}>\)
\(<\text{B> Congo</B> \text{<B> 242</B> \text{<B> 4</B>}/B><BR>\)
\(<\text{B> Egypt</B> \text{<B> 20</B> \text{<B> 4</B>}/B><BR>\)
\(<\text{B> Belize</B> \text{<B> 501</B> \text{<B> 4</B>}/B><BR>\)
\(<\text{B> Spain</B> \text{<B> 34</B> \text{<B> 4</B>}/B><BR>\)
\(<\text{HR}>\text{<B> End</B> </B>/</B>/</B>/</HTML>\)

\text{Head-Left-Right-Tail wrappers}

**end of head**

**start of tail**

\text{Head-Left-Right-Tail wrappers}
Boosted wrapper induction

- Wrapper induction is ideal for rigidly-structured machine-generated HTML...
- ... or is it?
- Can we use simple patterns 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 Klave, 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: www.smi.ucd.ie/bwi
- Not all natural text is sufficiently regular for exact string matching to work well!

Natural Language Processing

- 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, Hyponym: KILL]
      - Filler: [Phrase: NP]

Three generations of IE systems

  - 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)
- Statistical Generative Models [1997 –]
  - One decodes the statistical model to find which bits of the text are relevant, using HMMs or statistical parsers
  - Learning usually supervised; may be partially unsupervised

Trainable IE systems

Pros
- Annotating text is simpler & faster than writing rules.
- Domain independent
- Domain experts don’t need to be linguists or programmers.
- Learning algorithms ensure full coverage of examples.

Cons
- Hand-crafted systems perform better, especially at hard tasks.
- Training data might be expensive to acquire
- May need huge amount of training data
- Hand-writing rules isn’t that hard!!

MUC: the 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 of particular interest to the intelligence community (CIA, NSA).
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.

Example of IE from FASTUS (1993)

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: NTS200000000

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)

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: NTS200000000

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): Resolving anaphora

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.

FASTUS
Based on finite state automata (FSA) transductions

1. Complex Words: Recognition of multi-words and proper names
2. Basic Phrases: Simple noun groups, verb groups and particles
3. Complex phrases: Complex noun groups and verb groups
4. Domain Events: Patterns for events of interest to the application
Basic templates are to be built.
5. Merging Structures: Templates from different parts of the texts are merged if they provide information about the same entity or event.

Grep++ = Casacaded grepping
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 “metal wood” clubs a month.

Example of IE: FASTUS(1993)

1. Complex words
   - Bridgestone Sports Co.: Company name
   - said: Verb Group
   - Friday: Noun Group
   - it: Noun Group
   - had set up: Verb Group
   - a joint venture: Noun Group
   - in: Preposition
   - Taiwan: Location

2. Basic Phrases:
   - Bridgestone Sports Co.: Company name
   - said: Verb Group
   - Friday: Noun Group
   - it: Noun Group
   - had set up: Verb Group
   - a joint venture: Noun Group
   - in: Preposition
   - Taiwan: Location

Attachment
Ambiguities
are not made
explicit

Evaluating IE Accuracy

- Always evaluate performance on independent, manually-annotated test data not used during system development.
- 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


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? (BW 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!
- Applications: What exactly is IE good for?
  - Is there a use for today’s “60%” results?
  - Palmtop devices? – IE is valuable if screen is small
- Next time:
  - Learning methods for information extraction

Good Basic IE References