INFORMATION EXTRACTION

David Kauchak
cs159
Spring 2011


Administrative

- Quiz 4
  - keep up with book reading
  - keep up with paper reading
  - don’t fall asleep during the presentations 😊
  - ask questions
- Final projects
  - 4/15 Status report 1 (Friday)
  - 25% of your final grade
  - Rest of the semester’s papers posted soon
  - Assignment 5 grades out soon

A problem

- Mt. Baker, the school district
- Baker Hostelter, the company
- Baker, a job opening

Timeless...

- Baker, LA Jobs on CareerBuilder.com
- Baker, LA Jobs on Indeed.com
- Baker, LA Jobs on Jobs2Careers.com
- Baker, LA Jobs on Monster.com
- Baker, LA Jobs on ZipRecruiter.com
- Baker, LA Jobs on CareerBuilder.com
- Baker, LA Jobs on Indeed.com
- Baker, LA Jobs on Jobs2Careers.com
- Baker, LA Jobs on Monster.com
- Baker, LA Jobs on ZipRecruiter.com
- Baker, LA Jobs on CareerBuilder.com
- Baker, LA Jobs on Indeed.com
- Baker, LA Jobs on Jobs2Careers.com
- Baker, LA Jobs on Monster.com
- Baker, LA Jobs on ZipRecruiter.com
A solution

Why is this better?  How does it happen?

Extracting Job Openings from the Web

Another Problem

Job Openings:
Category = Food Services
Keyword = Baker
Location = Continental U.S.
Often structured information in text

Another Problem

And One more

Information Extraction

Traditional definition: Recovering structured data from text

What are some of the sub-problems/challenges?
Information Extraction?

- Recovering structured data from text
  - Identifying fields (e.g., named entity recognition)
  - Understanding relations between fields (e.g., record association)
  - Normalization and deduplication

Information extraction

- Input: Text Document
- Various sources: web, e-mail, journals, ...
- Output: Relevant fragments of text and relations possibly to be processed later in some automated way
BACKGROUND: The most challenging aspect of revision hip surgery is the management of bone loss. A reliable and valid measure of bone loss is important since it will aid in future studies of hip revisions and in preoperative planning. We developed a measure of femoral and acetabular bone loss associated with failed total hip arthroplasty. The purpose of the present study was to measure the reliability and the intraoperative validity of this measure and to determine how it may be useful in preoperative planning.

METHODS: From July 1997 to December 1998, forty-five consecutive patients with a failed hip prosthesis in need of revision surgery were prospectively followed. Three general orthopaedic surgeons were taught the radiographic classification system, and two of them classified standardized preoperative anteroposterior and lateral hip radiographs with use of the system. Interobserver testing was carried out in a blinded fashion. These results were then compared with the intraoperative findings of the third surgeon, who was blinded to the preoperative ratings. Kappa statistics (unweighted and weighted) were used to assess correlation. Interobserver reliability was assessed by examining the agreement between the two preoperative raters. Prognostic validity was assessed by examining the agreement between the assessment by either Rater 1 or Rater 2 and the intraoperative assessment (reference standard).

RESULTS: With regard to the assessments of both the femur and the acetabulum, there was significant agreement ($p < 0.0001$) between the preoperative raters (reliability), with weighted kappa values of $>0.75$. There was also significant agreement ($p < 0.0001$) between each rater’s assessment and the intraoperative assessment (validity) of both the femur and the acetabulum, with weighted kappa values of $>0.75$. CONCLUSIONS: With use of the newly developed classification system, preoperative radiographs are reliable and valid for assessment of the severity of bone loss that will be found intraoperatively.
For years, Microsoft Corporation CEO Bill Gates was against open source. But today he appears to have changed his mind. “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…

### Information extraction approaches

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill Gates</td>
<td>CEO</td>
<td>Microsoft</td>
</tr>
<tr>
<td>Bill Veghte</td>
<td>VP</td>
<td>Microsoft</td>
</tr>
<tr>
<td>Richard Stallman</td>
<td>Founder</td>
<td>Free Software</td>
</tr>
</tbody>
</table>

How can we do this? Can we utilize any tools/approaches we’ve seen so far?

### IE Posed as a Machine Learning Task

- Training data: documents marked up with ground truth
- Extract features around words/information
- Pose as a classification problem

### Good Features for Information Extraction

<table>
<thead>
<tr>
<th>Feature</th>
<th>Example word features</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains-question-mark</td>
<td>says string is a person name (80% accurate)</td>
</tr>
<tr>
<td>contains-question-word</td>
<td>In stopword list (e.g., is, in)</td>
</tr>
<tr>
<td>contains-do</td>
<td>In honorific list (e.g., Mr, Mrs, Dr, Sr, etc)</td>
</tr>
<tr>
<td>contains-first-alpha</td>
<td>In person suffix list (e.g., Jr, Sr, PhD, etc)</td>
</tr>
<tr>
<td>contains-italic</td>
<td>In name particle list (e.g., la, van, die, etc)</td>
</tr>
<tr>
<td>contains-latin-latin</td>
<td>In Census last name list, segmented by P(name)</td>
</tr>
<tr>
<td>contains-latin-firstname</td>
<td>In Census first name list, segmented by P(name)</td>
</tr>
<tr>
<td>contains-numeric</td>
<td>In locations lists (e.g., states, cities, countries)</td>
</tr>
<tr>
<td>contains-text</td>
<td>In company name list (e.g., J. C. Penny)</td>
</tr>
<tr>
<td>contains-abbreviation</td>
<td>In list of company suffixes (e.g., &amp; Associates, Foundation)</td>
</tr>
</tbody>
</table>

**HTML/Formatting Features**

- {begin, end, in} x {<b>, <i>, <a>, <hN>}
- {lengths 1, 2, 3, 4, or longer}
- {begin, end} of line

**Character N-gram Classifier**

- says string is a person name (80% accurate)
- In stopword list (e.g., is, in)
- In honorific list (e.g., Mr, Mrs, Dr, Sr, etc)
- In person suffix list (e.g., Jr, Sr, PhD, etc)
- In name particle list (e.g., la, van, die, etc)
- In Census last name list, segmented by P(name)
- In Census first name list, segmented by P(name)
- In locations lists (e.g., states, cities, countries)
- In company name list (e.g., J. C. Penny)
- In list of company suffixes (e.g., & Associates, Foundation)
How can we pose this as a classification (or learning) problem?

<table>
<thead>
<tr>
<th>Data</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Classifier

train a predictive model

Lots of possible techniques

Classify Candidates

Abraham Lincoln was born in Kentucky.

Sliding Window

Abraham Lincoln was born in Kentucky.

Boundary Models

Abraham Lincoln was born in Kentucky.

Finite State Machines

Abraham Lincoln was born in Kentucky.

Wrapper Induction

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

Information Extraction by Sliding Window
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.
Information Extraction by Sliding Window

- Standard supervised learning setting
  - Positive instances: Windows with real label
  - Negative instances: All other windows
  - Features based on candidate, prefix and suffix

IE by Boundary Detection

- Example: Looking for seminar location

Information Extraction by Sliding Window

- Standard supervised learning setting
  - Positive instances: Windows with real label
  - Negative instances: All other windows
  - Features based on candidate, prefix and suffix

IE by Boundary Detection

- Example: Looking for seminar location
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.

Input: Linear Sequence of Tokens

How can we pose this as a machine learning problem?

<table>
<thead>
<tr>
<th>Data</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Data | Label
---|---
0 | 0
0 | 1
1 | 1
0 | 0

Method: Identify start and end Token Boundaries

Output: Tokens Between Identified Start / End Boundaries
Learning: IE as Classification

Learn **TWO** binary classifiers, one for the beginning and one for the end.

**Begin**

<table>
<thead>
<tr>
<th>Date: Thursday, October 25</th>
<th>Time: 4:15 - 5:30 PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSITIVE (1)</td>
<td></td>
</tr>
<tr>
<td>ALL OTHERS NEGATIVE (0)</td>
<td></td>
</tr>
</tbody>
</table>

\[ \text{Begin}(i) = \begin{cases} 1 & \text{if } i \text{ begins a field} \\ 0 & \text{otherwise} \end{cases} \]

**One approach: Boundary Detectors**

A **“Boundary Detectors”** is a pair of token sequences \( p, s \).

- A detector matches a boundary if \( p \) matches text before boundary and \( s \) matches text after boundary.
- Detectors can contain wildcards, e.g. “capitalized word”, “number”, etc.

\[ <\text{Date: }, [\text{CapitalizedWord}]> \]

Would this boundary detector match anywhere?

**Combining Detectors**

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Suffix</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;a href=&quot; http</code></td>
<td><code>&quot;&gt;</code></td>
</tr>
</tbody>
</table>

\[ \text{match(es)?} \]
Combining Detectors

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Suffix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Begin boundary detector:</td>
<td>http</td>
</tr>
<tr>
<td>End boundary detector:</td>
<td>empty</td>
</tr>
</tbody>
</table>

Learning: IE as Classification

Learn **TWO** binary classifiers, one for the beginning and one for the end

**Begin**

Date: Thursday, October 25
Time: 4:15 - 5:30 PM

**End**

Date: Thursday, October 25
Time: 4:15 - 5:30 PM

Say we learn **Begin** and **End**, will this be enough? Any improvements? Any ambiguities?

Some concerns

1. Learn **three** probabilistic classifiers:
   - Begin(i) = probability position i starts a field
   - End(j) = probability position j ends a field
   - Len(k) = probability an extracted field has length k

2. Score a possible extraction (i,j) by
   
   \[ \text{Score} = \text{Begin}(i) \times \text{End}(j) \times \text{Len}(j-i) \]

3. **Len(k)** is estimated from a histogram data

4. **Begin(i)** and **End(j)** may combine multiple boundary detectors!
Problems with Sliding Windows and Boundary Finders

- Decisions in neighboring parts of the input are made independently from each other.
- 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.

Modeling the sequential nature of data: citation parsing


What patterns do you see here?

Ideas?

Some sequential patterns

- Authors come first
- Title comes before journal
- Page numbers come near the end
- All types of things generally contain multiple words

Predict a sequence of tags


Ideas?
Hidden Markov Models (HMMs)

HMM: Model
- States: $x_i$
- State transitions: $P(x_i | x_j) = a[x_i | x_j]$
- Output probabilities: $P(o_i | x_j) = b[o_i | x_j]$
- Markov independence assumption

HMMs: Performing Extraction
- Given output words:
  - fahlman 1991 the recurrent cascade correlation learning architecture nips 3 190 205
- Find state sequence that maximizes:
  $$\prod \left( a[x_i | x_{i-1}] b[o_i | x_j] \right)$$
- State transition Output probabilities
- Lots of possible state sequences to test ($5^{14}$)

IE Evaluation
- Precision
  - of those we identified, how many were correct?
- Recall
  - what fraction of the correct ones did we identify?
- F1
  - blend of precision and recall
### IE Evaluation

<table>
<thead>
<tr>
<th>Ground truth</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>author</td>
<td>author</td>
</tr>
<tr>
<td>year</td>
<td>year</td>
</tr>
<tr>
<td>title</td>
<td>title</td>
</tr>
<tr>
<td>title</td>
<td>title</td>
</tr>
</tbody>
</table>

**How should we calculate precision?**

5/6? 2/3? something else?

---

### Data regularity is important!

- **Highly structured**
- **Partially structured**
- **Natural text**

- As the regularity decreases, so does the performance

---

### Improving task regularity

- Instead of altering methods, alter text
- Idea: Add limited grammatical information
  - Run shallow parser over text
  - Flatten parse tree and insert as tags

**Example of Tagged Sentence:**

```plaintext
Uba2p is located largely in the nucleus.
```

- NP_SEQ
- VP_SEQ
- PP_SEQ
- NP_SEQ
Tagging Results on Natural Domain

Bootstrapping

Problem: Extract (author, title) pairs from the web

- Abraham Lincoln by James Russell Lowell
- Anna Karenina by Leo Tolstoy
- Adventures of Waverly Place, The by Arthur Conan Doyle
- Adventures of the Board, The by Arthur Conan Doyle
- Adventures of the Devil's Foot, The by Arthur Conan Doyle
- Adventures of the Devil's Den, The by Arthur Conan Doyle
- Adventures of the Red Carb, The by Arthur Conan Doyle
- Adventures of the Diamond, The by John Fiske

Approach 1: Old school style
Download the web:

Grab a sample and label:
Approach 1: Old school style

Download the web:
Grab a sample and label:
train model:
classifier

run model on web and get titles/authors

Problems? Better ideas?

Bootstrapping

Seed set
author/title pairs

Google
author/title occurrences in context
Bootstrapping

Seed set

author/title pairs

patterns

Bootstrapping

Seed set

author/title pairs

patterns

Brin, 1998
(Extracting patterns and relations from the world wide web)

Seed books

Patterns

New books

Experiments

<table>
<thead>
<tr>
<th>Iteration</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique (author, title) pairs</td>
<td>5</td>
<td>4647</td>
<td>9127</td>
</tr>
<tr>
<td>Occurrences</td>
<td>199</td>
<td>3972</td>
<td>9938</td>
</tr>
<tr>
<td>patterns</td>
<td>3</td>
<td>103</td>
<td>346</td>
</tr>
</tbody>
</table>

Result: unique pairs

4647  9127  15257
Final list

NELL

- NELL: Never-Ending Language Learning
  - http://rtw.ml.cmu.edu/rtw/
  - continuously crawls the web to grab new data
  - learns entities and relationships from this data
    - started with a seed set
    - uses learning techniques based on current data to learn new information

4 different approaches to learning relationships
- Combine these in the knowledge integrator
  - idea: using different approaches will avoid overfitting
- Initially was wholly unsupervised, now some human supervision
- cookies are food => internet cookies are food => files are food

An example learner: coupled pattern learner (CPL)

Cities:
- Los Angeles
- San Francisco
- New York
- Seattle

... city of X ...
... the official guide to X ...
... only in X ...
... what to do in X ...
... mayor of X ...

extract occurrences of group

statistical co-occurrence test

NELL continuously crawls the web to grab new data, learns entities and relationships from this data, started with a seed set, uses learning techniques based on current data to learn new information.
**CPL**

... mayor of <CITY> ...

- extract other cities from the data
  - Albuquerque
  - Springfield

- Can also learn patterns with multiple groups
  - ... X is the mayor of Y ...
  - ... X plays for Y ...
  - ... X is a player of Y ...
  - can extract other groups, but also relationships
  - Antonio Villaraigosa
  - mayor of Los Angeles

**NELL performance**

- The good:
  - Continuously learns
  - Uses the web (a huge data source)
  - Learns generic relationships
  - Combines multiple approaches for noise reduction

- The bad:
  - makes mistakes (overall accuracy still may be problematic for real world use)
  - does require some human intervention
  - still many general phenomena won’t be captured

For more details: http://rtw.ml.cmu.edu/papers/carlson-aaai10.pdf