Information Extraction

• Information extraction (IE) systems
  • Find and understand limited relevant parts of texts
  • Gather information from many pieces of text
  • Produce a structured representation of relevant information:
    • relations (in the database sense)
    • a knowledge base
  • Goals:
    1. Organize information so that it is useful to people
    2. Put information in a semantically precise form that allows further inferences to be made by computer algorithms

Slides based on Jurafsky and Manning

Information Extraction (IE)

• IE systems extract clear, factual information
  • Roughly: *Who did what to whom when?*
  • E.g.,
    • Gathering earnings, profits, board members, headquarters, etc. from company reports
      • The headquarters of BHP Billiton Limited, and the global headquarters of the combined BHP Billiton Group, are located in Melbourne, Australia.
      • headquarters(“BHP Billiton Limited”, “Melbourne, Australia”)
    • Learn drug-gene product interactions from medical research literature
Low-level information extraction

- Is now available in applications like Apple or Google mail, and web indexing

- Often seems to be based on regular expressions and name lists
Named Entity Recognition (NER)

• A very important sub-task: find and classify names in text, for example:

  • The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.
Named Entity Recognition (NER)

- A very important sub-task: **find** and **classify** names in text, for example:
  - The decision by the independent MP **Andrew Wilkie** to withdraw his support for the minority **Labor** government sounded dramatic but it should not further threaten its stability. When, after the **2010** election, **Wilkie**, **Rob Oakeshott**, **Tony Windsor** and the **Greens** agreed to support **Labor**, they gave just two guarantees: confidence and supply.

Named Entity Recognition (NER)

- The uses:
  - Named entities can be indexed, linked off, etc.
  - Sentiment can be attributed to companies or products
  - A lot of IE relations are associations between named entities
  - For question answering, answers are often named entities.

- Concretely:
  - Many web pages tag various entities, with links to bio or topic pages, etc.
    - Reuters’ OpenCalais, Evri, AlchemyAPI, Yahoo’s Term Extraction, ...
    - Apple/Google/Microsoft/... smart recognizers for document content
The Named Entity Recognition Task

Task: Predict entities in a text

<table>
<thead>
<tr>
<th>Entity</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreign</td>
<td>ORG</td>
</tr>
<tr>
<td>Ministry</td>
<td>ORG</td>
</tr>
<tr>
<td>spokesman</td>
<td>O</td>
</tr>
<tr>
<td>Shen</td>
<td>PER</td>
</tr>
<tr>
<td>Guofang</td>
<td>PER</td>
</tr>
<tr>
<td>told</td>
<td>O</td>
</tr>
<tr>
<td>Reuters</td>
<td>ORG</td>
</tr>
</tbody>
</table>

Standard evaluation is per entity, not per token

Precision/Recall/F1 for IE/NER

- Recall and precision are straightforward for tasks where there is only one grain size
- The measure behaves a bit funnily for IE/NER when there are boundary errors (which are common):
  - First Bank of Chicago announced earnings ...
- This counts as both a false positive and a false negative
- Selecting nothing would have been better
- Some other metrics (e.g., MUC scorer) give partial credit (according to complex rules)
The ML sequence model approach to NER

Training
1. Collect a set of representative training documents
2. Label each token for its entity class or other (O)
3. Design feature extractors appropriate to the text and classes
4. Train a sequence classifier to predict the labels from the data

Testing
1. Receive a set of testing documents
2. Run sequence model inference to label each token
3. Appropriately output the recognized entities

Encoding classes for sequence labeling

<table>
<thead>
<tr>
<th></th>
<th>IO encoding (Stanford)</th>
<th>IOB encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fred showed</td>
<td>PER</td>
<td>B-PER</td>
</tr>
<tr>
<td>Sue</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Mengqiu</td>
<td>PER</td>
<td>B-PER</td>
</tr>
<tr>
<td>Huang ‘s new</td>
<td>PER</td>
<td>I-PER</td>
</tr>
<tr>
<td>painting</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>new painting</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>
Features for sequence labeling

- Words
  - Current word (essentially like a learned dictionary)
  - Previous/next word (context)
- Other kinds of inferred linguistic classification
  - Part-of-speech tags
- Label context
  - Previous (and perhaps next) label

Features: Word substrings

- oxa
- field

Cotrimoxazole
Wethersfield
Alien Fury: Countdown to Invasion
Features: Word shapes

- Word Shapes
  - Map words to simplified representation that encodes attributes such as length, capitalization, numerals, Greek letters, internal punctuation, etc.

<table>
<thead>
<tr>
<th>mRNA</th>
<th>XXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPA1</td>
<td>XXXd</td>
</tr>
</tbody>
</table>

Sequence problems

- Many problems in NLP have data which is a sequence of characters, words, phrases, lines, or sentences ...
- We can think of our task as one of labeling each item

<table>
<thead>
<tr>
<th>VBG</th>
<th>NN</th>
<th>IN</th>
<th>DT</th>
<th>NN</th>
<th>IN</th>
<th>NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chasing</td>
<td>opportunity</td>
<td>in</td>
<td>an</td>
<td>age</td>
<td>of</td>
<td>upheaval</td>
</tr>
</tbody>
</table>

POS tagging

<table>
<thead>
<tr>
<th>PERS</th>
<th>O</th>
<th>O</th>
<th>O</th>
<th>ORG</th>
<th>ORG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Murdoch</td>
<td>discusses</td>
<td>future</td>
<td>of</td>
<td>News</td>
<td>Corp.</td>
</tr>
</tbody>
</table>

Named entity recognition

Word segmentation

Text segmentation
MEMM inference in systems

- For a Conditional Markov Model (CMM) a.k.a. a Maximum Entropy Markov Model (MEMM), the classifier makes a single decision at a time, conditioned on evidence from observations and previous decisions.
- A larger space of sequences is usually explored via search.

<table>
<thead>
<tr>
<th>Local Context</th>
<th>Decision Point</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3 DT The</td>
<td>0 Dow</td>
<td>W₀ 22.6</td>
</tr>
<tr>
<td>-2 NNP</td>
<td>1 fell</td>
<td>W₁ %</td>
</tr>
<tr>
<td>-1 VBD</td>
<td>0 22.6</td>
<td>T₁ VBD</td>
</tr>
<tr>
<td>0 ???</td>
<td>+1 ???</td>
<td>T₁*T₂ NNP-VBD</td>
</tr>
<tr>
<td>1 %</td>
<td>2 %</td>
<td>hasDigit? true</td>
</tr>
</tbody>
</table>

(Ratnaparkhi 1996; Toutanova et al. 2003, etc.)

Example: POS Tagging

- Scoring individual labeling decisions is no more complex than standard classification decisions.
  - We have some assumed labels to use for prior positions.
  - We use features of those and the observed data (which can include current, previous, and next words) to predict the current label.

<table>
<thead>
<tr>
<th>Local Context</th>
<th>Decision Point</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3 DT The</td>
<td>0 Dow</td>
<td>W₀ 22.6</td>
</tr>
<tr>
<td>-2 NNP</td>
<td>1 fell</td>
<td>W₁ %</td>
</tr>
<tr>
<td>-1 VBD</td>
<td>0 22.6</td>
<td>T₁ VBD</td>
</tr>
<tr>
<td>0 ???</td>
<td>+1 ???</td>
<td>T₁*T₂ NNP-VBD</td>
</tr>
<tr>
<td>1 %</td>
<td>2 %</td>
<td>hasDigit? true</td>
</tr>
</tbody>
</table>

(Ratnaparkhi 1996; Toutanova et al. 2003, etc.)
Example: POS Tagging

- POS tagging Features can include:
  - Current, previous, next words in isolation or together.
  - Previous one, two, three tags.
  - Word-internal features: word types, suffixes, dashes, etc.

<table>
<thead>
<tr>
<th>Local Context</th>
<th>Decision Point</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>-2</td>
<td>-1</td>
</tr>
<tr>
<td>DT</td>
<td>NNP</td>
<td>VBD</td>
</tr>
<tr>
<td>The</td>
<td>Dow</td>
<td>fell</td>
</tr>
<tr>
<td>W₀</td>
<td>22.6</td>
<td></td>
</tr>
<tr>
<td>W₁</td>
<td>%</td>
<td></td>
</tr>
<tr>
<td>W₂</td>
<td>fell</td>
<td></td>
</tr>
<tr>
<td>T₀</td>
<td>VBD</td>
<td></td>
</tr>
<tr>
<td>T₀·T₁</td>
<td>NNP-VBD</td>
<td></td>
</tr>
<tr>
<td>hasDigit?</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td></td>
</tr>
</tbody>
</table>

(Ratnaparkhi 1996; Toutanova et al. 2003, etc.)

Greedy Inference

- Greedy inference:
  - We just start at the left, and use our classifier at each position to assign a label
  - The classifier can depend on previous labeling decisions as well as observed data

- Advantages:
  - Fast, no extra memory requirements
  - Very easy to implement
  - With rich features including observations to the right, it may perform quite well

- Disadvantage:
  - Greedy. We make commit errors we cannot recover from
Beam Inference

- Beam inference:
  - At each position keep the top $k$ complete sequences.
  - Extend each sequence in each local way.
  - The extensions compete for the $k$ slots at the next position.
- Advantages:
  - Fast; beam sizes of 3–5 are almost as good as exact inference in many cases.
  - Easy to implement (no dynamic programming required).
- Disadvantage:
  - Inexact: the globally best sequence can fall off the beam.

CRFs [Lafferty, Pereira, and McCallum 2001]

- Another sequence model: Conditional Random Fields (CRFs)
- A whole-sequence conditional model rather than a chaining of local models
Extracting relations from text

- **Company report:** "International Business Machines Corporation (IBM or the company) was incorporated in the State of New York on June 16, 1911, as the Computing-Tabulating-Recording Co. (C-T-R)...

- **Extracted Complex Relation:**
  - **Company-Founding**
    - **Company:** IBM
    - **Location:** New York
    - **Date:** June 16, 1911
    - **Original-Name:** Computing-Tabulating-Recording Co.

- **But we will focus on the simpler task of extracting relation triples**
  - Founding-year(IBM,1911)
  - Founding-location(IBM,New York)

---

Extracting Relation Triples from Text

Leland Stanford Junior University, commonly referred to as Stanford University, is an American private research university located in ... near Palo Alto, Stanford...founded 1891

- **Stanford EQ Leland Stanford Junior University**
- **Stanford LOC-IN California**
- **Stanford IS-A research university**
- **Stanford LOC-NEAR Palo Alto**
- **Stanford FOUNDED-IN 1891**
- **Stanford FOUNDER Leland Stanford**
Why Relation Extraction?

- Create new structured knowledge bases, useful for any app
- Augment current knowledge bases
  - Adding words to WordNet thesaurus
- But which relations should we extract?

Automated Content Extraction (ACE)

17 relations from 2008 “Relation Extraction Task”
Automated Content Extraction (ACE)

- Part-Whole-Subsidiary **ORG-ORG**
  *XYZ*, the parent company of **ABC**
- Person-Social-Family **PER-PER**
  John’s wife **Yoko**
- Org-AFF-Founder **PER-ORG**
  *Steve Jobs*, co-founder of **Apple**...

UMLS: Unified Medical Language System

- 134 entity types, 54 relations

<table>
<thead>
<tr>
<th>Injury</th>
<th>disrupts</th>
<th>Physiological Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bodily Location</td>
<td>location-of</td>
<td>Biologic Function</td>
</tr>
<tr>
<td>Anatomical Structure</td>
<td>part-of</td>
<td>Organism</td>
</tr>
<tr>
<td>Pharmacologic Substance</td>
<td>causes</td>
<td>Pathological Function</td>
</tr>
<tr>
<td>Pharmacologic Substance</td>
<td>treats</td>
<td>Pathologic Function</td>
</tr>
</tbody>
</table>
Ontological relations

Examples from the WordNet Thesaurus

• IS-A (hyponym): subsumption between classes
  • Giraffe IS-A ruminant IS-A ungulate IS-A mammal IS-A vertebrate IS-A animal...

• Instance-of: relation between individual and class
  • San Francisco instance-of city

How to build relation extractors

1. Hand-written patterns
2. Supervised machine learning
3. Semi-supervised and unsupervised
Rules for extracting IS-A relation

Early intuition from Hearst (1992)

• “Agar is a substance prepared from a mixture of red algae, such as Gelidium, for laboratory or industrial use”
• What does Gelidium mean?
• How do you know?"
## Hearst’s Patterns for extracting IS-A relations

<table>
<thead>
<tr>
<th>Hearst pattern</th>
<th>Example occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>X and other Y</td>
<td>...temples, treasuries, and other important civic buildings.</td>
</tr>
<tr>
<td>X or other Y</td>
<td>Bruises, wounds, broken bones or other injuries...</td>
</tr>
<tr>
<td>Y such as X</td>
<td>The bow lute, such as the Bambara ndang...</td>
</tr>
<tr>
<td>Such Y as X</td>
<td>...such authors as Herrick, Goldsmith, and Shakespeare.</td>
</tr>
<tr>
<td>Y including X</td>
<td>...common-law countries, including Canada and England...</td>
</tr>
<tr>
<td>Y, especially X</td>
<td>European countries, especially France, England, and Spain...</td>
</tr>
</tbody>
</table>

## Extracting Richer Relations Using Rules

- Intuition: relations often hold between specific entities
  - located-in (ORGANIZATION, LOCATION)
  - founded (PERSON, ORGANIZATION)
  - cures (DRUG, DISEASE)
- Start with Named Entity tags to help extract relation!
Named Entities aren’t quite enough. Which relations hold between 2 entities?

Drug

Cure?

Prevent?

Cause?

Disease

Extracting Richer Relations Using Rules and Named Entities

Who holds what office in what organization?

PERSON, POSITION of ORG

- George Marshall, Secretary of State of the United States

PERSON (named | appointed | chose | etc.) PERSON Prep? POSITION

- Truman appointed Marshall Secretary of State

PERSON [be]? (named | appointed | etc.) Prep? ORG POSITION

- George Marshall was named US Secretary of State
Hand-built patterns for relations

- **Plus:**
  - Human patterns tend to be high-precision
  - Can be tailored to specific domains
- **Minus**
  - Human patterns are often low-recall
  - A lot of work to think of all possible patterns!
  - Don’t want to have to do this for every relation!
  - We’d like better accuracy

Supervised machine learning for relations

- Choose a set of relations we’d like to extract
- Choose a set of relevant named entities
- Find and label data
  - Choose a representative corpus
  - Label the named entities in the corpus
  - Hand-label the relations between these entities
  - Break into training, development, and test
- Train a classifier on the training set
How to do classification in supervised relation extraction

1. Find all pairs of named entities (usually in same sentence)
2. Decide if 2 entities are related
3. If yes, classify the relation
   • Why the extra step?
     • Faster classification training by eliminating most pairs
     • Can use distinct feature-sets appropriate for each task.

Automated Content Extraction (ACE)

17 sub-relations of 6 relations from 2008 “Relation Extraction Task”
Relation Extraction

Classify the relation between two entities in a sentence

American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said.

Word Features for Relation Extraction

- Headwords of M1 and M2, and combination
  - Airlines, Wagner, Airlines-Wagner
- Bag of words and bigrams in M1 and M2
  - {American, Airlines, Tim, Wagner, American Airlines, Tim Wagner}
- Words or bigrams in particular positions left and right of M1/M2
  - M2: -1 spokesman
  - M2: +1 said
- Bag of words or bigrams between the two entities
  - {a, AMR, of, immediately, matched, move, spokesman, the, unit}
Named Entity Type and Mention Level Features for Relation Extraction

*American Airlines*, a unit of AMR, immediately matched the move, spokesman *Tim Wagner* said

Mention 1

• Named-entity types
  • M1: ORG
  • M2: PERSON

• Concatenation of the two named-entity types
  • ORG-PERSON

• Entity Level of M1 and M2 (NAME, NOMINAL, PRONOUN)
  • M1: NAME [it or he would be PRONOUN]
  • M2: NAME [the company would be NOMINAL]

Parse Features for Relation Extraction

*American Airlines*, a unit of AMR, immediately matched the move, spokesman *Tim Wagner* said

Mention 1

Mention 2

• Base syntactic chunk sequence from one to the other
  NP NP PP VP NP NP

• Constituent path through the tree from one to the other
  NP ↑ NP ↑ S ↑ S ↓ NP

• Dependency path
  Airlines matched Wagner said
Gazetteer and trigger word features for relation extraction

- Trigger list for family: kinship terms
  - parent, wife, husband, grandparent, etc. [from WordNet]
- Gazeteer:
  - Lists of useful geo or geopolitical words
    - Country name list
    - Other sub-entities

---

**American Airlines**, a unit of AMR, immediately matched the move, spokesman **Tim Wagner** said.

<table>
<thead>
<tr>
<th>Entity-based features</th>
<th>Word-based features</th>
<th>Syntactic features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity\textsubscript{1} type</td>
<td>ORG</td>
<td>(NP \uparrow NP \uparrow S \uparrow S \downarrow NP)</td>
</tr>
<tr>
<td>Entity\textsubscript{1} head</td>
<td>airlines</td>
<td>(NP \rightarrow NP \rightarrow PP \rightarrow NP \rightarrow VP \rightarrow NP \rightarrow NP)</td>
</tr>
<tr>
<td>Entity\textsubscript{2} type</td>
<td>PERS</td>
<td>(\textit{ Airlines} \leftarrow_{\text{subj matched}} \text{ -- comp } \text{ said } \rightarrow_{\text{subj}} \text{ Wagner})</td>
</tr>
<tr>
<td>Entity\textsubscript{2} head</td>
<td>Wagner</td>
<td></td>
</tr>
<tr>
<td>Concatenated types</td>
<td>ORGPERS</td>
<td></td>
</tr>
</tbody>
</table>
Classifiers for supervised methods

- Now you can use any classifier you like
  - MaxEnt
  - Naïve Bayes
  - SVM
  - ...
- Train it on the training set, tune on the dev set, test on the test set

Evaluation of Supervised Relation Extraction

- Compute P/R/F₁ for each relation

\[
P = \frac{\text{# of correctly extracted relations}}{\text{Total # of extracted relations}}
\]

\[
F_1 = \frac{2PR}{P + R}
\]

\[
R = \frac{\text{# of correctly extracted relations}}{\text{Total # of gold relations}}
\]
Summary: Supervised Relation Extraction

+ Can get high accuracies with enough hand-labeled training data, if test similar enough to training
- Labeling a large training set is expensive
- Supervised models are brittle, don’t generalize well to different genres

Seed-based or bootstrapping approaches to relation extraction

• No training set? Maybe you have:
  • A few seed tuples or
  • A few high-precision patterns
• Can you use those seeds to do something useful?
  • Bootstrapping: use the seeds to directly learn to populate a relation
Relation Bootstrapping (Hearst 1992)

- Gather a set of seed pairs that have relation R
- Iterate:
  1. Find sentences with these pairs
  2. Look at the context between or around the pair and generalize the context to create patterns
  3. Use the patterns for grep for more pairs

Bootstrapping

- <Mark Twain, Elmira> Seed tuple
  - Grep (google) for the environments of the seed tuple
    "Mark Twain is buried in Elmira, NY."
      X is buried in Y
    "The grave of Mark Twain is in Elmira"
      The grave of X is in Y
    "Elmira is Mark Twain’s final resting place"
      Y is X’s final resting place.
- Use those patterns to grep for new tuples
- Iterate
Evaluation of Semi-supervised and Unsupervised Relation Extraction

- Since it extracts totally new relations from the web
  - There is no gold set of correct instances of relations!
  - Can’t compute precision (don’t know which ones are correct)
  - Can’t compute recall (don’t know which ones were missed)
- Instead, we can approximate precision (only)
  - Draw a random sample of relations from output, check precision manually
    \[
    \hat{p} = \frac{\text{# of correctly extracted relations in the sample}}{\text{Total # of extracted relations in the sample}}
    \]