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

Foreign ORG
Ministry ORG
spokesman O
Shen PER
Guofang PER
told O
Reuters ORG
:

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 funny 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</td>
<td>PER</td>
<td>B-PER</td>
</tr>
<tr>
<td>showed</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Sue</td>
<td>PER</td>
<td>B-PER</td>
</tr>
<tr>
<td>Mengqiu</td>
<td>PER</td>
<td>B-PER</td>
</tr>
<tr>
<td>Huang</td>
<td>PER</td>
<td>I-PER</td>
</tr>
<tr>
<td>‘s</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>new</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>painting</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>
What is Sequence Learning?

• Most machine learning algorithms are designed for independent, identically distributed (i.i.d.) data
• But many interesting data types are not i.i.d.
  • the successive points in sequential data are strongly correlated
• Sequence learning is the study of machine learning algorithms designed for sequential data. These algorithms should
  • not assume independence
  • make use of context

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

- **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>xXXX</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

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>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>W₀</td>
</tr>
<tr>
<td>W₁</td>
</tr>
<tr>
<td>T₁</td>
</tr>
<tr>
<td>T₁-T₂</td>
</tr>
<tr>
<td>hasDigit?</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
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

Local Context

<table>
<thead>
<tr>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>NNP</td>
<td>VBD</td>
<td>??</td>
<td>??</td>
</tr>
<tr>
<td>The</td>
<td>Dow</td>
<td>fell</td>
<td>22.6</td>
<td>%</td>
</tr>
</tbody>
</table>

Decision Point

<table>
<thead>
<tr>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_0$</td>
</tr>
<tr>
<td>$W_{-1}$</td>
</tr>
<tr>
<td>$W_{-1}$</td>
</tr>
<tr>
<td>$T_{-1}$</td>
</tr>
<tr>
<td>$T_{-2}$</td>
</tr>
</tbody>
</table>

hasDigit? true

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.

Local Context

<table>
<thead>
<tr>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>NNP</td>
<td>VBD</td>
<td>??</td>
<td>??</td>
</tr>
<tr>
<td>The</td>
<td>Dow</td>
<td>fell</td>
<td>22.6</td>
<td>%</td>
</tr>
</tbody>
</table>

Decision Point

<table>
<thead>
<tr>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_0$</td>
</tr>
<tr>
<td>$W_{-1}$</td>
</tr>
<tr>
<td>$W_{-1}$</td>
</tr>
<tr>
<td>$T_{-1}$</td>
</tr>
<tr>
<td>$T_{-2}$</td>
</tr>
</tbody>
</table>

hasDigit? true
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.
Viterbi Inference

- Viterbi inference:
  - Dynamic programming

- Advantage:
  - Exact: the global best sequence is returned.

- Disadvantage:
  - Harder to implement long-distance state-state interactions (but beam inference tends not to allow long-distance resurrection of sequences anyway).

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:
  
  ![Complex Relation](image1)

- But we will focus on the simpler task of extracting relation triples:
  
  ![Simple Relation](image2)

Extracting Relation Triples from Text

Leland Stanford Junior University, commonly referred to as Stanford or Stanford University, is an American private research university located in ... near Palo Alto, California...founded 1891.
Why Relation Extraction?

- Create new structured knowledge bases, useful for any app
- Augment current knowledge bases
  - Adding words to WordNet thesaurus, facts to DBPedia
- Support question answering
  - The granddaughter of which actor starred in the movie “E.T.”?
- But which relations should we extract?

Automated Content Extraction (ACE)

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

- **Physical-Located** \[\text{PER-GPE}\]
  - He was in Tennessee

- **Part-Whole-Subsidiary** \[\text{ORG-ORG}\]
  - XYZ, the parent company of ABC

- **Person-Social-Family** \[\text{PER-PER}\]
  - John’s wife Yoko

- **Org-AFF-Founder** \[\text{PER-ORG}\]
  - Steve Jobs, co-founder of Apple...

UMLS: Unified Medical Language System

- 134 entity types, 54 relations

<table>
<thead>
<tr>
<th>Entity Type</th>
<th>Relation</th>
<th>Function Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Injury</td>
<td>disrupts</td>
<td>Physiological Function</td>
</tr>
<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>
Extracting UMLS relations from a sentence

Doppler echocardiography can be used to diagnose left anterior descending artery stenosis in patients with type 2 diabetes

Echocardiography, Doppler DIAGNOSES Acquired stenosis

Databases of Wikipedia Relations

Wikipedia Infobox

<table>
<thead>
<tr>
<th>Type</th>
<th>Private</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endowment</td>
<td>US$ 16.5 billion (2011)</td>
</tr>
<tr>
<td>President</td>
<td>John L. Hennessy</td>
</tr>
<tr>
<td>Provost</td>
<td>John Etchemendy</td>
</tr>
<tr>
<td>Academic staff</td>
<td>1,910(4)</td>
</tr>
<tr>
<td>Students</td>
<td>15,319</td>
</tr>
<tr>
<td>Undergraduates</td>
<td>6,878(5)</td>
</tr>
<tr>
<td>Postgraduates</td>
<td>8,441(5)</td>
</tr>
<tr>
<td>Location</td>
<td>Stanford, California, U.S.</td>
</tr>
<tr>
<td>Campus</td>
<td>Suburban, 8.180 acres (3,310 ha)</td>
</tr>
<tr>
<td>Colors</td>
<td>Cardinal red and white</td>
</tr>
</tbody>
</table>
Relation databases that draw from Wikipedia

- Resource Description Framework (RDF) triples
  subject predicate object
  Golden Gate Park location San Francisco
dbpedia:Golden_Gate_Park dbpedia-owl:location dbpedia:San_Francisco
- DBPedia: 1 billion RDF triples, 385 from English Wikipedia
- Frequent Freebase relations:
  people/person/nationality, location/location/contains
  people/person/profession, people/person/place-of-birth
  biology/organism_higher_classification film/film/genre

Ontological relations

Examples from the WordNet Thesaurus

- IS-A (hypernym): 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
   • Bootstrapping (using seeds)
   • Distant supervision
   • Unsupervised learning from the web

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?“
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
(Hearst, 1992): Automatic Acquisition of Hyponyms

<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
What relations hold between 2 entities?

Founder?

Investor?

Member?

Employee?

President?

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

- **American Airlines**, a unit of AMR, immediately matched the move, spokesman **Tim Wagner** said
  
  Mention 1

  - 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
    - \(M2: -1 \text{spokesman}\)
    - \(M2: +1 \text{said}\)
  
  - Bag of words or bigrams between the two entities
    - \{a, AMR, of, immediately, matched, move, spokesman, the, unit\}

Mention 2
Named Entity Type and Mention Level Features for Relation Extraction

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

- 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

- 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></th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity₁ type</td>
<td>ORG</td>
</tr>
<tr>
<td>Entity₁ head</td>
<td>airlines</td>
</tr>
<tr>
<td>Entity₂ type</td>
<td>PERS</td>
</tr>
<tr>
<td>Entity₂ head</td>
<td>Wagner</td>
</tr>
<tr>
<td>Concatenated types</td>
<td>ORGPERS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word-based features</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Between-entity bag of words</td>
<td>{ <em>a, unit, of, AMR, Inc., immediately, matched, the, move, spokesman</em> }</td>
</tr>
<tr>
<td>Word(s) before Entity₁</td>
<td>NONE</td>
</tr>
<tr>
<td>Word(s) after Entity₂</td>
<td>said</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syntactic features</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Constituent path</td>
<td><em>NP → NP → S → S → NP</em></td>
</tr>
<tr>
<td>Base syntactic chunk path</td>
<td><em>NP → NP → PP → NP → VP → NP → NP</em></td>
</tr>
<tr>
<td>Typed-dependency path</td>
<td><em>Airlines ← _subj matched ← _comp said → subj Wagner</em></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

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

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

\[
F₁ = \frac{2PR}{P + R}
\]
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
Dipre: Extract <author,book> pairs

- Start with 5 seeds:
<table>
<thead>
<tr>
<th>Author</th>
<th>Book</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isaac Asimov</td>
<td>The Robots of Dawn</td>
</tr>
<tr>
<td>David Brin</td>
<td>Startide Rising</td>
</tr>
<tr>
<td>James Gleick</td>
<td>Chaos: Making a New Science</td>
</tr>
<tr>
<td>Charles Dickens</td>
<td>Great Expectations</td>
</tr>
<tr>
<td>William Shakespeare</td>
<td>The Comedy of Errors</td>
</tr>
</tbody>
</table>

- Find Instances:
  - The Comedy of Errors, by William Shakespeare, was
  - The Comedy of Errors, by William Shakespeare, is
  - The Comedy of Errors, one of William Shakespeare’s earliest attempts
  - The Comedy of Errors, one of William Shakespeare’s most

- Extract patterns (group by middle, take longest common prefix/suffix)
  \(?x\), by \(?y\), \(?x\), one of \(?y\)’s

- Now iterate, finding new seeds that match the pattern

Snowball
E. Agichtein and L. Gravano 2000. Snowball: Extracting Relations from Large Plain-Text Collections. ICDL

- Similar iterative algorithm

<table>
<thead>
<tr>
<th>Organization</th>
<th>Location of Headquarters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft</td>
<td>Redmond</td>
</tr>
<tr>
<td>Exxon</td>
<td>Irving</td>
</tr>
<tr>
<td>IBM</td>
<td>Armonk</td>
</tr>
</tbody>
</table>

- Group instances w/similar prefix, middle, suffix, extract patterns
  - But require that X and Y be named entities
  - And compute a confidence for each pattern

  .69  ORGANIZATION  {‘s, in, headquarters}  LOCATION
  .75  LOCATION     {in, based}             ORGANIZATION
Distant Supervision
Snow, Jurafsky, Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. NIPS 17
Fei Wu and Daniel S. Weld. 2007. Autonomously Semantifying Wikipeida. CIKM 2007
Mintz, Bills, Snow, Jurafsky. 2009. Distant supervision for relation extraction without labeled data. ACL09

• Combine bootstrapping with supervised learning
  • Instead of 5 seeds,
    • Use a large database to get huge # of seed examples
  • Create lots of features from all these examples
  • Combine in a supervised classifier

Distant supervision paradigm

• Like supervised classification:
  • Uses a classifier with lots of features
  • Supervised by detailed hand-created knowledge
  • Doesn’t require iteratively expanding patterns

• Like unsupervised classification:
  • Uses very large amounts of unlabeled data
  • Not sensitive to genre issues in training corpus
### Distantly supervised learning of relation extraction patterns

1. For each relation
2. For each tuple in big database
3. Find sentences in large corpus with both entities
4. Extract frequent features (parse, words, etc)
5. Train supervised classifier using thousands of patterns

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Born-In</td>
<td>&lt;Edwin Hubble, Marshfield&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;Albert Einstein, Ulm&gt;</td>
</tr>
<tr>
<td></td>
<td>Hubble was born in Marshfield</td>
</tr>
<tr>
<td></td>
<td>Einstein, born (1879), Ulm</td>
</tr>
<tr>
<td></td>
<td>Hubble’s birthplace in Marshfield</td>
</tr>
</tbody>
</table>

\[
P(\text{born-in} \mid f_1, f_2, f_3, \ldots, f_{70000})
\]

### Unsupervised relation extraction

- Open Information Extraction:
  - extract relations from the web with no training data, no list of relations
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
    \[ \hat{p} = \frac{\text{# of correctly extracted relations in the sample}}{\text{Total # of extracted relations in the sample}} \]
- Can also compute precision at different levels of recall.
  - Precision for top 1000 new relations, top 10,000 new relations, top 100,000
  - In each case taking a random sample of that set
- But no way to evaluate recall

Temporal IE for Temporal Reasoning

- Another area, optional