Information Extraction and Named Entity Recognition
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.k.a.,
    • 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
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 – and I think popular – in applications like Apple or Google mail, and web indexing
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 like IR and text categorization, where there is only one grain size (documents)
• 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 fp and a fn
• Selecting nothing would have been better
• Some other metrics (e.g., MUC scorer) give partial credit (according to complex rules)

\[
\text{wrong} = \text{INCorrect} + \text{Partial}/2 + \text{MISSing} + \text{SPurious}
\]
\[
\text{total} = \text{COR} + \text{PAR} + \text{INC} + \text{MIS} + \text{SPU}
\]
\[
\text{error} = \frac{\text{wrong}}{\text{total}}
\]
Sequence Models for Named Entity Recognition
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>Name</th>
<th>IO encoding</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>
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

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>Varicella-zoster</th>
<th>Xx-xxx</th>
</tr>
</thead>
<tbody>
<tr>
<td>mRNA</td>
<td>xXXX</td>
</tr>
<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

POS tagging

<table>
<thead>
<tr>
<th>VB</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>

Named entity recognition

<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>

Word segmentation

B B I I B I B I B I B B

Text segmentation

Q A Q A A Q A
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.

Local Context

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>+1</td>
</tr>
<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>

Features

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_0$</td>
<td>22.6</td>
</tr>
<tr>
<td>$W_{+1}$</td>
<td>%</td>
</tr>
<tr>
<td>$W_{-1}$</td>
<td>fell</td>
</tr>
<tr>
<td>$T_{-1}$</td>
<td>VBD</td>
</tr>
<tr>
<td>$T_{-1}-T_{-2}$</td>
<td>NNP-VBD</td>
</tr>
<tr>
<td>hasDigit?</td>
<td>true</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

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

Sequence Level

- Sequence Data
- Local Data

Local Level

- Feature Extraction
- Label
- Features
- Maximum Entropy Models
- Conjugate Gradient
- Quadratic Penalties
- Classifier Type
- Optimization
- Smoothing

Sequence Model

Inference
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
Viterbi Inference

- Viterbi inference:
  - Dynamic programming or memoization.
  - Requires small window of state influence (e.g., past two states are relevant).

- 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).
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
  (MEMM- multiplication of local model)

To compute $c$ as sequence
*MEMM: multiplication of $c_i$ position ($\tau$)
*CRF: sum of $c_i$ position in the feature (inside exp)

$$P(c \mid d, \lambda) = \frac{\exp \sum \lambda_i f_i(c, d)}{\sum \exp \sum_{c'} \lambda_i f_i(c', d)}$$

Feature (potential function) can look at future value

- The space of $c'$ s is now the space of sequences
  - But if the features $f_i$ remain local, the conditional sequence likelihood can be calculated exactly using dynamic programming (viterbi search)
- Training is slower, but CRFs avoid causal-competition biases (label biases)
- These (or a variant using a max margin criterion) are seen as the state-of-the-art these days … but in practice usually work much the same as MEMMs.
Relation Extraction

What is relation extraction?
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:**
  
<table>
<thead>
<tr>
<th>Company-Founding</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>IBM</td>
</tr>
<tr>
<td>Location</td>
<td>New York</td>
</tr>
<tr>
<td>Date</td>
<td>June 16, 1911</td>
</tr>
<tr>
<td>Original-Name</td>
<td>Computing-Tabulating-Recording Co.</td>
</tr>
</tbody>
</table>

- **But we will focus on the simpler task of extracting relation** triples

  Founding-year(IBM,1911)
  
  Founding-location(IBM,New York)
Stanford University, commonly referred to as Stanford University or Stanford, is an American private research university located in Stanford, California, near Palo Alto, California. Leland Stanford, a Californian railroad tycoon and politician, founded the university in 1891 in honor of his son, Leland Stanford, Jr., who died of typhoid two months before his 18th birthday. The university was established as a coeducational and nonsectarian institution, but struggled financially after the senior Stanford's 1893 death and after much of the campus was damaged by the 1906 San Francisco earthquake. Following World War II, Provost Frederick Terman supported faculty and graduates' entrepreneurialism to build a self-sufficient local industry in what would become known as Silicon Valley. By 1955, Stanford was home to a linear accelerator, was one of the original four ARPANET nodes, and had transformed itself into a major research university in computer science, mathematics, natural sciences, and social sciences. More than 50 Stanford faculty members and alumni have won the Nobel Prize and Stanford has the largest number of Turing Award winners for a single institution. Stanford faculty and alumni have founded many prominent technology companies including Cisco Systems, Google, Hewlett-Packard, LinkedIn, Rambus, Silicon Graphics, Sun Microsystems, Varian Associates, and Yahoo.

The university is organized into seven schools including academic schools of Humanities, Science, Engineering, Law, Business, Education, and Medicine.
Why Relation Extraction?

- Create new structured knowledge bases, useful for any app
- Augment current knowledge bases
  - Adding words to WordNet thesaurus, facts to FreeBase or DBPedia
- Support question answering
  - The granddaughter of which actor starred in the movie “E.T.”?
    (acted-in ?x “E.T.”) (is-a ?y actor) (granddaughter-of ?x ?y)
- 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}
  \text{He was in Tennessee}

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

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

- Org-AFF-Founder \text{PER-ORG}
  \text{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>
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

Relations extracted from Infobox
Stanford state California
Stanford motto “Die Luft der Freiheit weht”

<table>
<thead>
<tr>
<th>Type</th>
<th>Private</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endowment</td>
<td>US$ 16.5 billion (2011)[3]</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)[6]</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, 385M 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

* hierarchical relation name
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
   • 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?"
### 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, <strong>and other</strong> important civic buildings.</td>
</tr>
<tr>
<td>X or other Y</td>
<td>Bruises, wounds, broken bones <strong>or other</strong> injuries...</td>
</tr>
<tr>
<td>Y such as X</td>
<td>The bow lute, <strong>such as</strong> the Bambara ndang...</td>
</tr>
<tr>
<td>Such Y as X</td>
<td>...<strong>such</strong> authors <strong>as</strong> Herrick, Goldsmith, and Shakespeare.</td>
</tr>
<tr>
<td>Y including X</td>
<td>...common-law countries, <strong>including</strong> Canada and England...</td>
</tr>
<tr>
<td>Y, especially X</td>
<td>European countries, <strong>especially</strong> 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.
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 *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.

• 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>Entity$_1$ type</td>
<td>ORG</td>
</tr>
<tr>
<td>Entity$_1$ head</td>
<td>airlines</td>
</tr>
<tr>
<td>Entity$_2$ type</td>
<td>PERS</td>
</tr>
<tr>
<td>Entity$_2$ 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>{ a, unit, of, AMR, Inc., immediately, matched, the, move, spokesman }</td>
</tr>
<tr>
<td>Word(s) before Entity$_1$</td>
<td>NONE</td>
</tr>
<tr>
<td>Word(s) after Entity$_2$</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>$NP \uparrow NP \uparrow S \uparrow S \downarrow NP$</td>
</tr>
<tr>
<td>Base syntactic chunk path</td>
<td>$NP \rightarrow NP \rightarrow PP \rightarrow NP \rightarrow VP \rightarrow NP \rightarrow NP$</td>
</tr>
<tr>
<td>Typed-dependency path</td>
<td>Airlines $\leftarrow_{subj} matched \leftarrow_{comp} said \rightarrow_{subj} Wagner$</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 $P/R/F_1$ for each relation

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

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

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

Semi-supervised and unsupervised relation extraction
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
  - 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

<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>

\[
\text{Conf (P)} = \frac{P.\text{positive}}{P.\text{positive} + P.\text{negative}}
\]

.69 \text{ ORGANIZATION} \{s, in, headquarters\} \text{ LOCATION}

.75 \text{ LOCATION} \{in, based\} \text{ 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
   - Find sentences in large corpus with both entities
3. Extract frequent features (parse, words, etc)
4. Train supervised classifier using thousands of patterns
5. 

Born-In

<Edwin Hubble, Marshfield>
<Albert Einstein, Ulm>

Hubble was born in Marshfield
Einstein, born (1879), Ulm
Hubble’s birthplace in Marshfield

PER was born in LOC
PER, born (XXXX), LOC
PER’s birthplace in LOC

P(born-in | f1,f2,f3,…,f70000)
Unsupervised relation extraction

M. Banko, M. Cararella, S. Soderland, M. Broadhead, and O. Etzioni.
2007. Open information extraction from the web. IJCAI

- **Open Information Extraction:**
  - extract relations from the web with no training data, **no list of relations**

1. Use parsed data to train a “trustworthy tuple” classifier
2. Single-pass extract all relations between NPs, keep if trustworthy
3. Assessor ranks relations based on text redundancy

(FCI, specializes in, software development)
(Tesla, invented, coil transformer)
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