

# Relation Extraction

What is relation  
extraction?

Many slides adapted from Dan Jurafsky

# 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)

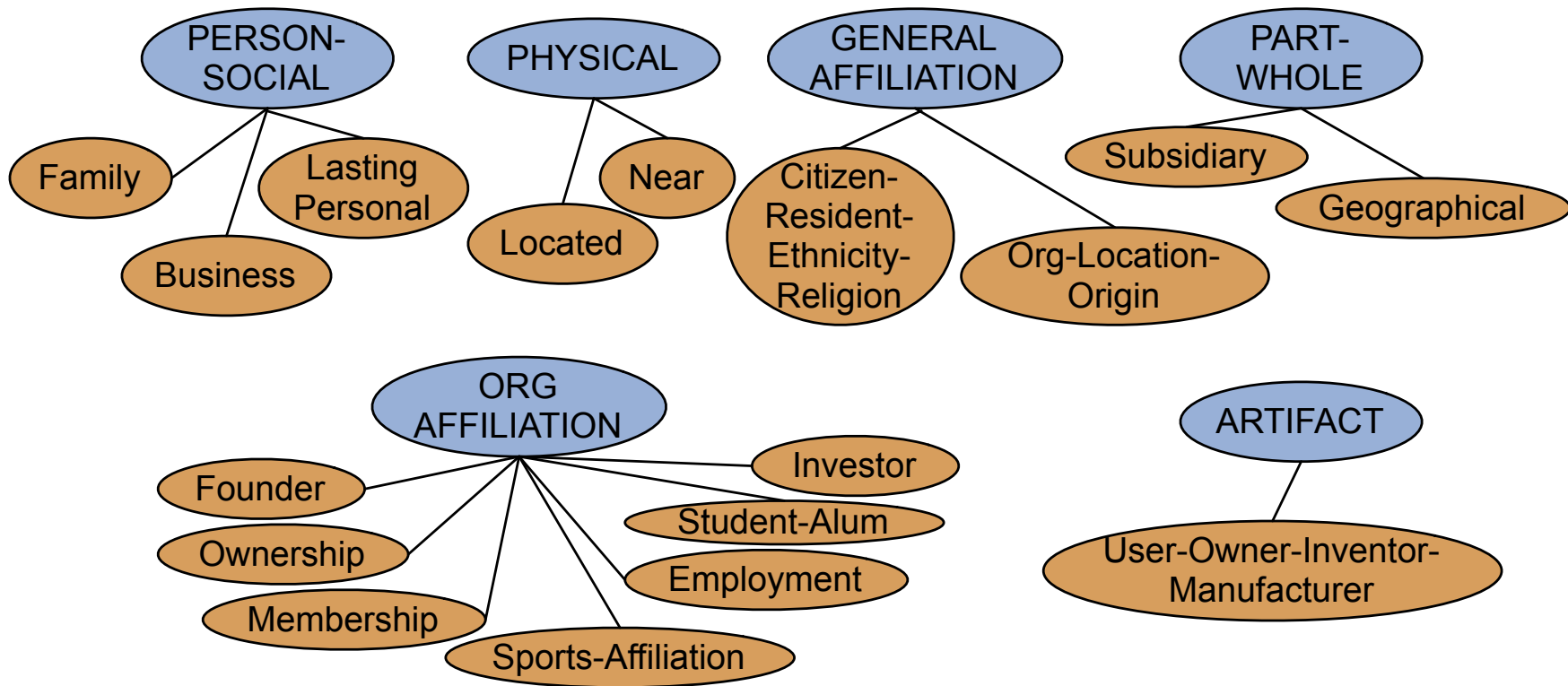
Founding-location(IBM,New York)

# 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?

# Automatic Content Extraction (ACE)

17 relations from 2008 “Relation Extraction Task”



# Automatic Content Extraction (ACE)

- Physical-Located      PER-GPE  
    He was in Tennessee
- 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

Injury	disrupts	Physiological Function
Bodily Location	location-of	Biologic Function
Anatomical Structure	part-of	Organism
Pharmacologic Substance	causes	Pathological Function
Pharmacologic Substance	treats	Pathologic Function

# Databases of Wikipedia Relations

## Wikipedia Infobox

Relations extracted from Infobox

Stanford [state](#) California

Stanford [motto](#) “Die Luft der Freiheit weht”

```
{{Infobox university
|image_name= Stanford University seal.svg
|image_size= 210px
|caption = Seal of Stanford University
|name =Stanford University
|native_name =Leland Stanford Junior Uni
|motto = {{lang|de|"Die Luft der Freiheit v
name="casper">{{cite speech|title=Die Lu
Casper|first=Gerhard|last=Casper|author
05|url=http://www.stanford.edu/dept/pr
|mottoeng = The wind of freedom blows<
|established = 1891<ref>{{cite web |
url=http://www.stanford.edu/home/stan
publisher = Stanford University | accessd:
|type = [[private university|Private]]
|calendar= Quarter
|president = [[John L. Hennessy]]
|provost = [[John Etchemendy]]
|city = [[Stanford, California|Stanford]]
|state = California
|country = U.S.
```

<b>Type</b>	Private
<b>Endowment</b>	US\$ 16.5 billion (2011) <sup>[3]</sup>
<b>President</b>	John L. Hennessy
<b>Provost</b>	John Etchemendy
<b>Academic staff</b>	1,910 <sup>[4]</sup>
<b>Students</b>	15,319
<b>Undergraduates</b>	6,878 <sup>[5]</sup>
<b>Postgraduates</b>	8,441 <sup>[5]</sup>
<b>Location</b>	Stanford, California, U.S.
<b>Campus</b>	Suburban, 8,180 acres (3,310 ha) <sup>[6]</sup>
<b>Colors</b>	Cardinal red and white



```

|
tml}}</ref>
```

```
ty History |
```

# 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



# Relation Extraction

What is relation  
extraction?

# Relation Extraction

Using patterns to  
extract relations

# 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?



PERSON

Founder?

Investor?

Member?

Employee?

President?



ORGANIZATION

# 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

# Relation Extraction

Using patterns to  
extract relations



# Relation Extraction

Supervised relation  
extraction

# 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.

FAMILY

CITIZEN

SUBSIDIARY

FOUNDER



NIL

EMPLOYMENT

INVENTOR

...

# Word Features for Relation Extraction

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

Mention 1

Mention 2

- 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 Mention 2

- 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

# Gazeteer 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.

### Entity-based features

Entity <sub>1</sub> type	ORG
Entity <sub>1</sub> head	<i>airlines</i>
Entity <sub>2</sub> type	PERS
Entity <sub>2</sub> head	<i>Wagner</i>
Concatenated types	ORGPERS

### Word-based features

Between-entity bag of words	{ <i>a, unit, of, AMR, Inc., immediately, matched, the, move, spokesman</i> }
Word(s) before Entity <sub>1</sub>	NONE
Word(s) after Entity <sub>2</sub>	<i>said</i>

### Syntactic features

Constituent path	$NP \uparrow NP \uparrow S \uparrow S \downarrow NP$
Base syntactic chunk path	$NP \rightarrow NP \rightarrow PP \rightarrow NP \rightarrow VP \rightarrow NP \rightarrow NP$
Typed-dependency path	$Airlines \leftarrow_{subj} matched \leftarrow_{comp} said \rightarrow_{subj} Wagner$

# 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

Supervised relation  
extraction

# 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

Brin, Sergei. 1998. Extracting Patterns and Relations from the World Wide Web.

- Start with 5 seeds:

Author	Book
Isaac Asimov	The Robots of Dawn
David Brin	Startide Rising
James Gleick	Chaos: Making a New Science
Charles Dickens	Great Expectations
William Shakespeare	The Comedy of Errors

- 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

# 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

# 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 features

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

$P(\text{born-in} \mid f_1, f_2, f_3, \dots, f_{70000})$

# Distant supervision paradigm

- Like unsupervised classification:
  - Uses very large amounts of unlabeled data
  - Not sensitive to genre issues in training corpus
- Like supervised classification:
  - Uses a classifier with lots of features
  - Supervised by detailed hand-created knowledge
  - Doesn't require iteratively expanding patterns

# 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

(Google, acquired, Android) *to compete with Apple.*

(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

39 But no way to evaluate recall

# Relation Extraction

Semi-supervised  
and unsupervised  
relation extraction