# Information Extraction

## Extracting limited forms of information from text

- Named entity recognition (NER) seeks to
  - Identify where each named entity is mentioned
  - Identify its type: person, place, organization, . . .
  - Unify distinct names for the same entity
    - United = United Airlines
- Foundational step for virtually any kind of advanced reasoning
  - Extracting relations as to build knowledge graphs
  - Extracting events
  - Answering questions

Suggest a few uses of NER

# Named Entity Recognition

- Entities that can be named
  - For news: Person, location, organization
  - For medicine: drugs, . . .
- Even entities that aren't named, e.g., dates and numbers
- The sentence:

  This Friday United is selling $100 fares to The Big Apple on their new Dreamliner
- Yields this markup:

  This [$_{\text{TIME}}$Friday] [$_{\text{ORG}}$United] is selling [$_{\text{MONEY}}$$100] fares to [$_{\text{LOC}}$The Big Apple] on their new [$_{\text{VEH}}$Dreamliner]
- Challenges
  - Segmentation: what are the boundaries of an entity
  - Ambiguity: JFK can be a person, an airport, . . .
  - Exacerbated by metonymy: Washington (city, government, sports teams)

# Named Entity Types

| **Type** | **Tag** | **Sample Categories** |
|---|---|---|
| People | PER | People, characters |
| Organization | ORG | Companies, teams |
| Location | LOC | Regions, mountains, seas |
| Geopolitical Entity | GPE | Countries, provinces |
| Facility | FAC | Bridges, buildings, airports |
| Vehicle | VEH | Planes, trains, automobiles |

# IOB Tagging for Named Entity Recognition

## Similar to IOB for chunking

- Introduce $2n+1$ tags (given $n$ types—earlier chunk, here NER)
  - $B_k$: Beginning of type $k$
  - $I_k$: Inside of type $k$
  - $O$: Outside of all types
- Example of IOB chunking for NER:

| | | | | | | |
|---|---|---|---|---|---|---|
| Woodson | , | Chancellor | of | NC | State | University |
| $[\mathrm{B}_{\mathrm{PER}}]$ | O | $[\mathrm{B}_{\mathrm{PER}}]$ | O | $[\mathrm{B}_{\mathrm{ORG}}]$ | $[\mathrm{I}_{\mathrm{ORG}}]$ | $[\mathrm{I}_{\mathrm{ORG}}]$ |
| | , | is | a | professor | | |
| | O | O | O | O | | |

# IO Tagging for Named Entity Recognition

## Simpler variant of IOB: Omit the Begin tags

- Requires only $n+1$ tags for $n$ types
- Confuses contiguous names *of the same type* as one name
- Such contiguous names are rare in English, though

| | | | | | | |
|---|---|---|---|---|---|---|
| Woodson | , | Chancellor | of | NC | State | University |
| [I$_{\text{PER}}$] | O | [I$_{\text{PER}}$] | O | [I$_{\text{ORG}}$] | [I$_{\text{ORG}}$] | [I$_{\text{ORG}}$] |
| | , | is | a | professor | | |
| | O | O | O | O | | |

# Feature-Based Named Entity Recognition

- Word-based features

| This word | Neighboring Words |
|---|---|
| Identity | Identity |
| Embedding | Embedding |
| POS | POS |
| Base-phrase label (IOB tag) | Base-phrase label (IOB tag) |
| Presence in a gazetteer (list of place names) | |

- Character-based features, geared toward unknown words

| This word | Neighboring Words |
|---|---|
| Specific prefix up to length 4 | |
| Specific suffix up to length 4 | |
| All upper case | |
| Hyphenated | |
| Word shape | Word shape |
| Short word shape | Short word shape |

# Word Shape and Short Word Shape

- Word shape: a pattern based on the symbols in a word
  - Map upper case letter to X
  - Map lower case letter to x
  - Digit to d
  - Retain hyphens, apostrophes, periods
  - L'Occitane ⇒ X'Xxxxxxxx ($X'Xx^7$)
  - DC10-30 ⇒ XXdd-dd ($X^2d^2$-$d^2$)
  - I.M.F. ⇒ X.X.X.
- Short word shape: reduce consecutive character types to one
  - L'Occitane ⇒ X'Xx
  - DC10-30 ⇒ Xd-d
  - I.M.F. ⇒ X.X.X.

# Computing NER

- Sequence labeling via
  - Neural models
  - Maximum Entropy Markov Models (logistic regression plus Viterbi)
  - Both rely of inputs such as
    - Features of current, preceding, and following words
    - Labels of preceding words
- Rules: multiple passes each seeking to improve recall
  - High-precision rules for unambiguous names
  - Substrings of identified names
  - Domain-specific name lists
  - Sequence labeling (probabilistic, as above) to complete the list

# Relation Extraction

Identify and classify semantic relations between entities found in the text

- General purpose
  - Child-of: taxonomy
  - Part-whole: meronomy
  - Geospatial
- Domain specific
  - Employee of (domain of human resources)
  - Additive for (domain of chemistry)

# Generic Relations

Read each relation label as a path in a hierarchy

| Relation | Type Pair | Example |
|---|---|---|
| Physical:Located | PER-GPE | IBM, head-quartered in Armonk NY, |
| Part:Whole:Subsidiary | ORG-ORG | XYZ, the parent of ABC, |
| Person:Social:Family | PER-PER | Clinton's daughter, Chelsea |
| Org-Affiliation:Founder | PER-ORG | Microsoft founder, Bill Gates, |

# Relations in Medical Language

Using National Library of Medicine (NLM)'s UMLS, the Unified Medical Language System
https://www.nlm.nih.gov/research/umls/pdf/AMIA_T12_2006_UMLS.pdf

- 135 subject categories (entity types)
- 54 relations between categories

| **Relation** | **Type Pair** | **Example** |
|---|---|---|
| isa | Entity-Entity | Lab Result isa Finding<br>Enzyme isa Biologically Active Substance |
| | Relationship-Relationship | prevents isa affects |
| treats | Pharmacologic Substance – Pathologic Function | Calcium channel blockers treat hypertension |
| diagnoses | Finding – Pathologic Function | Echocardiogram diagnoses stenosis |

- Domain-independent: isa, part of, causes
- Domain-specific (for medicine): treats, diagnoses

# Structured Information on the Web

Usable for NL
Potentially extractable from NL

- Wikipedia Infoboxes
  - Provide structure for facts suited to a given entry
  - Structured facts are relations
- Resource Description Framework (RDF), a W3C recommendation (standard)
  - Expresses statements as triples in the form of
  - Subject, Predicate, Object
- Crowdsourced ontologies such as DBpedia
- WordNet: to be discussed later
- Infoboxes in web search results: provided by a webmaster

# How Can we Extract Instances of a Known Relation?

Assume a large corpus of text

- Given isa, discover
  - Aspirin is a Medication
  - Cardiologist is a Medical Practitioner

# Lexico-Syntactic Patterns

## Manually constructed

- (Hearst patterns) Hyponym relations are often apparent in the syntax
  - Seeing "A, such as B, . . ."
  - We can conclude that B is a hyponym of A
- Coordination applies naturally by forcing type agreement
  - Seeing "A, such as B and C, . . ."
  - We can conclude that B is a hyponym of A
  - We can conclude that C is a hyponym of A
- Key idea: identify lexical markers of hyponym-hypernym relations
  - Including
  - Especially: Z, especially X, . . .
  - And other: X, Y, and other Zs,

# Regular Expressions as Generalized Patterns

## Can tackle broader relations

- per, position of org
  - Relates the instance of person as holder of the specified position in the referenced organization instance
  - $[_{\text{PER}}$George Marshall], $[_{\text{POSITION}}$Secretary of State] of the $[_{\text{ORG}}$United States]
- per (named| appointed| . . . ) per (Prep?) position
  - $[_{\text{PER}}$Truman] appointed $[_{\text{PER}}$Marshall] $[_{\text{POSITION}}$Secretary of State]
- (Xibin Gao) "In case of xxx, the contract is null and . . . "
  - Not about named entities
  - Helps identify exceptions highlighted in a contract—such exceptions are common within a business domain

# Features for Supervised Relation Extraction

- Identify *mentions* $M_1$ and $M_2$
- Important features as word embeddings
  - Headwords of $M_1$ and $M_2$
  - Concatenation of headwords of $M_1$ and $M_2$
  - Adjacent words to $M_1$ and $M_2$
  - N-grams between $M_1$ and $M_2$
- NER features
  - Types of $M_1$ and $M_2$ and their concatenation
  - Entity (constituent) level from Name, Nominal, Pronoun
  - Number of intervening entities between $M_1$ and $M_2$
- Syntactic structure, expressed via *syntactic paths* from $M_1$ and $M_2$ of
  - Base chunks: NP, NP, PP, VP, NP, NP
  - Constituents: NP $\uparrow$ NP $\uparrow$ S $\uparrow$ S $\downarrow$ NP
  - Dependencies: Airlines $\leftarrow_{\text{subj}}$ matched $\leftarrow_{\text{comp}}$ said $\rightarrow_{\text{subj}}$ Wagner

# Bootstrapping

- Given instances of a relation as M$_1$–R–M$_2$ (Aspirin–treats-headache)
  - Identify occurrences of M$_1$ and M$_2$ in the corpus
  - Identify patterns that fit those occurrences
  - Apply resulting patterns to identify additional instances
- Semantic drift: Risk of bootstrapping
  - Errors in the initial pattern (e.g., confusing ferry hub for airport hub) propagate
- Pattern confidence, as measure of quality, possibly normalized to [0,1]
- Estimated based on a given set $T$ of relation tuples (instance)

$$\mathrm{conf}(p) = \frac{\mathrm{hits}_p}{\mathrm{finds}_p} \log(\mathrm{finds})_p$$

- Confidence of a tuple $t$ based on at least one pattern that finds $t$

$$\mathrm{confidence}(t) = 1 - \prod_{p \text{ is a pattern for } t} (1 - \mathrm{conf}(p))$$

- Confidence threshold for acceptance

# Extracting Temporal Expressions

- Main varieties
  - Absolute
  - Relative
  - Durational
  - How can we classify holidays, e.g., Memorial Day, Easter, Diwali?
- Often associated with lexical triggers
  - Nouns: Dusk, dawn,
  - Proper Nouns: January, Monday, Ides of March, Rosh Hashana, Ramadan
  - Adjectives: Recent, annual, former
  - Adverbs: hourly, usually
- False hits: temporal expressions used atemporally
  - 1984 (the book or movie)
  - Sunday Bloody Sunday (song by the Irish group U2)

# Temporal Ambiguity

- Where to anchor an expression?
  - Reichenbach's theory, later
- Which polarity to adopt given an anchor (before or after)?
  - Next
  - This

# Event Extraction

How events link to various entities

- Event coreference
  - Which mentions of an event refer to the same event
- Temporal expressions
  - Days, dates, times
  - Relative expressions, such as “next month”
- Normalization with respect to
  - Calendar
  - Discourse, e.g., time of utterance or reference

# Event Extraction

## Identify events or states from text

- Classically, events are occurrences, not states, which are indicated by verbs such as
  - Be, is, are
  - Know, feel, believe
- In the extraction literature, events include states
  - Verbs: increased
  - Nouns: the increase
  - Gerunds: increasing
- Nonevents
  - Verbs indicating transition into an event: took effect
  - Weak or light verbs (make, take, have) that rely on a direct object to bring out an event

# Event Details

- Tense: past, future, present
- Aspect: more complex
  - Progressive: leaving
  - Perfective: left
  - Perfect: has left
- Famous example:

  Einstein has left Princeton

  vs.

  Einstein left Princeton
- Subtypes of events
  - States
  - Actions
  - Reporting events (geared toward news)
  - Perception events

# Temporal Relations and Ordering

## James Allen's thirteen relations between two temporal intervals

Each relation has an inverse

- Before and after
- Overlaps
- Meets
- Equals
- Starts
- Finishes
- During

Draw these relations out

# Template Filling

## How to flesh out set patterns or stereotypical situations

- For an application on business intelligence in the airline industry, we might have an event such as

| | | |
|---|---|---|
| Fare-raising | Leader airline | United Airlines |
| | Amount | $66 |
| | Effective date | 2018-10-07 |
| | Follower | American Airlines |

- As a template, the attributes below are fixed but the values are found in the text

| | | |
|---|---|---|
| Event type | Attribute 1 | Value 1 |
| | Attribute 2 | Value 2 |
| | Attribute 3 | Value 3 |
| | Attribute 4 | Value 4 |

Suggest a short example for the personal fitness industry

# Prototypical Event Structures

## Schank ∼1960s: Scripts and Stories

- Postulated as central representation in cognition
- Relate to Lakoff's conceptual schemas, which additionally signify how events are *framed*
- Scripts highlight a typical structure
  - For having dinner at a restaurant
  - For attending a cocktail party
  - For experiences as a college student
- Facts retrieved from a narrative flesh out a relevant script
  - Provides slots to be filled
  - The slots are interrelated: filler of one constrains another
- A script helps fill in the gaps
  - Between entering a restaurant and receiving food would be the ordering event
  - A waiter would be a normal character in a restaurant script

# Machine Learning for Template Filling

1. Component: Template Recognizer, a text classifier
   - Whether a template occurs in a sentence
   - Learns a template from instances of sentences that fill any slot in the template
   - Collective across all slots in a template
2. Component: Slot Filler (Role Filler), a text classifier
   - One for each slot, e.g., Lead Airline, in a template
   - Needs coreference resolution to reconcile alternatives for the same concept