Chapter 18:
Information Extraction

Review and Pointers

• From logistic regression to neural nets (Chapter 7)
  • Each neural unit multiplies input values by a weight vector, adds a bias, and then applies a non-linear activation function (e.g., sigmoid, etc.)
  • Early layers learn representations that can be utilized by later layers

• More recent developments
  • Unassigned chapters 9, 10
    • Sequence processing with recurrent networks
    • Encoder-decoder models, attention, and contextual embeddings
  • From simple word embeddings to BERT (linked article)
    • GLUE: A MULTI-TASK BENCHMARK AND ANALYSIS PLATFORM FOR NATURAL LANGUAGE UNDERSTANDING
Machines Beat Humans on a Reading Test. But Do They Understand? (article cont.)

• “Before 2018, one of NLP’s main pretraining tools was something like a dictionary. Known as word embeddings, this dictionary encoded associations between words as numbers in a way that deep neural networks could accept as input. But a neural network pretrained with word embeddings is still blind to the meaning of words at the sentence level. ‘It would think that ‘a man bit the dog’ and ‘a dog bit the man’ are exactly the same thing,’”...

• A better method would use pretraining to equip the network with richer rulebooks — not just for vocabulary, but for syntax and context as well — before training it to perform a specific NLP task...

• .. Each of these three ingredients — a deep pretrained language model, attention and bidirectionality — existed independently before BERT. But until Google released its recipe in late 2018, no one had combined them in such a powerful way.”

Administrivia

• Exam and project
• Homework 3
Information Extraction

• A “dumbing down” of more lofty goal of true Natural Language Understanding (i.e., semantics)
  • more technologically manageable
    • often domain and application specific
  • useful practical applications

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
Low-level information extraction

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.

Person
Date
Location
Organization
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

As usual, the problem of ambiguity!

Figure 17.3 Examples of type ambiguities in the use of the name Washington.
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>IO encoding (Stanford)</th>
<th>IOB encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fred</td>
<td>PER</td>
</tr>
<tr>
<td>showed</td>
<td>O</td>
</tr>
<tr>
<td>Sue</td>
<td>PER</td>
</tr>
<tr>
<td>Mengqiu</td>
<td>PER</td>
</tr>
<tr>
<td>Huang</td>
<td>PER</td>
</tr>
<tr>
<td>‘s</td>
<td>O</td>
</tr>
<tr>
<td>new</td>
<td>O</td>
</tr>
<tr>
<td>painting</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**
  - Drug: 18
- **field**
  - Place: 68
  - Person: 6

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>XXXd</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>

<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>

<table>
<thead>
<tr>
<th>B</th>
<th>B</th>
<th>I</th>
<th>I</th>
<th>B</th>
<th>I</th>
<th>B</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>而相对于这些品牌的价</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Text segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q A A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q A Q</td>
</tr>
</tbody>
</table>
MEMM inference in systems

- For a Maximum Entropy Markov Model (MEMM), the classifier makes a single decision at a time, conditioned on evidence from observations and previous decisions
- Maximum entropy is an outdated name for logistic regression

<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>The</td>
<td>Dow</td>
<td>fell</td>
</tr>
</tbody>
</table>

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

MEMMs

- Turn logistic regression onto a discriminative sequence model
  - HMM was generative
    - Easier to add arbitrary features into discriminative models
  - Logistic regression was not a sequence model
  - Optional details in Section 8.5
- Run logistic regression on successive words, using the class assigned to the prior word as a feature in the classification of the next word
HMM

\[ \hat{T} = \arg\max_T P(T|W) \]
\[ = \arg\max_T P(W|T)P(T) \]
\[ = \arg\max_T \prod_i P(\text{word}_i|\text{tag}_i) \prod_i P(\text{tag}_i|\text{tag}_{i-1}) \]

MEMM

\[ \hat{T} = \arg\max_T P(T|W) \]
\[ = \arg\max_T \prod_i P(t_i|w_i, t_{i-1}) \]

Figure 8.12 A schematic view of the HMM (top) and MEMM (bottom) representation of the probability computation for the correct sequence of tags for the *back* sentence. The HMM computes the likelihood of the observation given the hidden state, while the MEMM computes the posterior of each state, conditioned on the previous state and current observation.
Features in a MEMM

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>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3 DT</td>
<td>W₀</td>
</tr>
<tr>
<td>-2 NNP</td>
<td>W₁ᵥ₂</td>
</tr>
<tr>
<td>-1 VBD</td>
<td>W₂ᵥ₁</td>
</tr>
<tr>
<td>0</td>
<td>T₁</td>
</tr>
<tr>
<td>+1</td>
<td>T₁₋₁₋₂</td>
</tr>
<tr>
<td>The Dow fell</td>
<td>hasDigit?</td>
</tr>
<tr>
<td></td>
<td>...</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 DT</td>
<td>-2 NNP</td>
<td>-1 VBD</td>
</tr>
<tr>
<td>The</td>
<td>Dow</td>
<td>fell</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
Recently also Neural Methods

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

Stanford University, commonly referred to as Stanford, 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
- But which relations should we extract?
Automated Content Extraction (ACE)

17 relations from 2008 “Relation Extraction Task”

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

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

- Org-AFF-Founder **PER-ORG**
  
  \textit{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

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 (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

Review

- Admin Questions?
  - Hw3
  - Project (groups, schedule, details)?
- Introduction to Information Extraction
- Named Entity Recognition
  - What?
  - How?
- Relation Extractors
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

<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!
  • (Maps well to logical representations)

Named Entities aren’t quite enough. Which relations hold between 2 entities?

Drug  Cure?  Prevent?  Cause?  Disease
What relations hold between 2 entities?

PERSON ORGANIZATION

- 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
  - 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

- 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
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.

<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>{ a, unit, of, AMR, Inc., immediately, matched, the, move, spokesman }</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>NP → NP → S ↓ S ↓ NP</td>
</tr>
<tr>
<td>Base syntactic chunk path</td>
<td>NP → NP → PP → NP → VP → NP → NP</td>
</tr>
<tr>
<td>Typed-dependency path</td>
<td>Airlines ←subj matched ←comp said →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 (both labeled and unlabeled versions, like with parsing)

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

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

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

**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
    - Often use confidence values to avoid drift
  - 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

  ?x, by ?y,                  ?x, one of ?y's

- Now iterate, finding new seeds that match the pattern

Distant Supervision

- 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
  - Need to also create negative examples via random sampling
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
   Born-In
   <Edwin Hubble, Marshfield>
   <Albert Einstein, Ulm>

2. For each tuple in big database
   Hubble was born in Marshfield
   Einstein, born (1879), Ulm
   Hubble’s birthplace in Marshfield

3. Find sentences in large corpus with both entities

4. Extract frequent features (parse, words, etc)
   PER was born in LOC
   PER, born (XXXX), LOC
   PER’s birthplace in LOC

5. Train supervised classifier using thousands of patterns
   \[ P(\text{born-in} \mid f_1, f_2, f_3, \ldots, f_{70000}) \]
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

Chapter Summary

• IE involves techniques for extracting limited forms of semantic content, e.g.,
  • Named entities
  • Relations among entities
• Methods included various types of machine learning as well as hand crafted methods