Relation Extraction: Rule-based Approaches
CSCI-GA.2590

Ralph Grishman
Information Extraction Evaluations

• CoNLL has sponsored annual evaluations of NLP components for about 15 years
• NIST has organized (annual) US Government evaluations of information extraction for about 25 years
  • covering both components and integrated systems
  • MUC [Message Understanding Conferences] in the 1990’s
  • ACE [Automatic Content Extraction] 2000-2008
  • KBP [Knowledge Base Population] since 2009
ACE Extraction Tasks

Types of information to extract:

• names
• entities [coreferential noun phrases]
• relations
• events
Relations

- A *relation* is a predication about a pair of entities:
  - Rodrigo works for UNED.
  - Alfonso lives in Tarragona.
  - Otto’s father is Ferdinand.

- Typically they represent information which is permanent or of extended duration.
ACE Relations

• Several revisions of relation definitions
  • With goal of having a set of relations which can be more consistently annotated

• 5-7 major types, 19-24 subtypes

• Both entities must be mentioned in the same sentence
  – Do not get a parent-child relation from
    • Ferdinand and Isabella were married in 1481. A son was born in 1485.
  – Or an employee relation for
    • Bank Santander replaced several executives. Alfonso was named an executive vice president.

• Base for extensive research
  – On supervised and semi-supervised methods
# 2004 Ace Relation Types

<table>
<thead>
<tr>
<th>Relation type</th>
<th>Subtypes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical</td>
<td>Located, Near, Part-whole</td>
</tr>
<tr>
<td>Personal-social</td>
<td>Business, Family, Other</td>
</tr>
<tr>
<td>Employment / Membership / Subsidiary</td>
<td>Employ-executive, Employ-staff, Employ-undetermined, Member-of-group, Partner, Subsidiary, Other</td>
</tr>
<tr>
<td>Agent-artifact</td>
<td>User-or-owner, Inventor-or-manufacturer, Other</td>
</tr>
<tr>
<td>Person-org affiliation</td>
<td>Ethnic, Ideology, Other</td>
</tr>
<tr>
<td>GPE affiliation</td>
<td>Citizen-or-resident, Based-in, Other</td>
</tr>
<tr>
<td>Discourse</td>
<td>-</td>
</tr>
</tbody>
</table>
Characteristics of Relations

- Relations appear in a wide range of forms:
  - Embedded constructs (one argument contains the other)
    - within a single noun group
      - John’s wife
    - linked by a preposition
      - the president of Apple
  - Formulaic constructs
    - Tarragona, Spain
    - Walter Cronkite, CBS News, New York
  - Longer-range (‘predicate-linked’) constructs
    - With a predicate disjoint from the arguments
      - Fred lived in New York
      - Fred and Mary got married
Hand-crafted patterns

• Most instances of relations can be identified by the types of the entities and the words between the entities
  • But not all: Fred and Mary got married.

• Word sequence patterns work well enough for short-range relations
  • But problems arise for longer-range patterns ... greater variety, intervening modifiers
Parsing

• progress through corpus-trained parsers
  • probabilistic context-free parsers
  • corpus-trained shift-reduce parsers
  • more accurate, much faster

• how do we take advantage of parsing?
  • arguments of semantic relation generally connected by a limited set of syntactic structures and lexical items
  • need not take into account the wide range of intervening words
Parsing

• “Fred shot Mary.”
• “Fred, 61, shot Mary.”
• “Fred, tired of her endless lectures on parsing, shot Mary.”

• all have the same dependency relations:
  – verb “shot”
  – subject of shot = “Fred”
  – object of shot = “Mary”
Dependency Structures (1)

- Label sets for dependency structures from different parsers are similar but not identical.
- Most widely used set is from Stanford NLP tools.
- We will use a variant set from USC/ISI.
  - Used in Tratz-Hovy dependency parser.
Dependency Structures (2)

• root of tree is generally a (tensed) verb
  – auxiliaries and modals appear as vch [verb chain] dependents of tensed verb
  – principal arguments appear as
    • nsubj [noun subject]
    • dobj [direct object]
    • iobj [indirect object]
  – sentential complements appear as
    • ccomp
    • xcomp
Dependency Structures (3)

• noun modifiers
  – poss [possessive]
  – amod [adjective modifiers]
  – nn [compound noun]
• prepositional phrases: prep and pobj
• conj [conjunction]
Lexicalized Dependency Paths

• path in dependency tree between two entity mentions

• combines dependency types and lexical items
  • type = edge from governor to dependent
  • type-1 = edge from dependent to governor

PERSON – nsubj-1:shoot:dobj -- PERSON
Transformations (1)

- Using dependency paths (rather than linear patterns) greatly increases coverage.
- Can further (modestly) increase coverage through transformations that connect closely related structures:
  - operate to simplify dependency parse
  - reduce sentences to *kernel sentences* + *transformations*
Transformations (2)

• passive:
  – The cake was baked by Harry. \(\rightarrow\) Harry baked the cake.

• relative
  – Harry, who baked the cake \(\rightarrow\) Harry baked the cake

• reduced relative
  – the cake baked by Harry \(\rightarrow\) the cake, which was baked by Harry
Transformations (3)

• subject control:
  – Harry planned to bake the cake $\Rightarrow$ Harry planned
    (Harry baked the cake)