Information Extraction

Instructor: Thien Huu Nguyen

Based on slides from: Ralph Grishman
Giuliani, 58, proposed to Nathan, a former nurse, during a business trip to Paris five months after he finalized his divorce from Donna Hanover in July after 20 years of marriage.

In interviews last year, Giuliani said Nathan gave him "tremendous emotional support" through his treatment for prostate cancer and as he led New York City during the Sept. 11, 2001, terror attacks.

IE = automatically extracting structured information from unstructured and/or semi-structured machine-readable documents
Information Extraction Pipeline

Giuliani, 58, proposed to Nathan, a former nurse, during a business trip to Paris _ five months after he finalized his divorce from Donna Hanover in July after 20 years of marriage.

In interviews last year, Giuliani said Nathan gave him "tremendous emotional support" through his treatment for prostate cancer and as he led New York City during the Sept. 11, 2001, terror attacks.
Giuliani, 58, proposed to Nathan, a former nurse, during a business trip to Paris — five months after he finalized his divorce from Donna Hanover in July after 20 years of marriage.

In interviews last year, Giuliani said Nathan gave him "tremendous emotional support" through his treatment for prostate cancer and as he led New York City during the Sept. 11, 2001, terror attacks.
Giuliani, 58, proposed to Nathan, a former nurse, during a business trip to Paris - five months after he finalized his divorce from Donna Hanover in July after 20 years of marriage.

In interviews last year, Giuliani said Nathan gave him "tremendous emotional support" through his treatment for prostate cancer and as he led New York City during the Sept. 11, 2001 terror attacks.
Giuliani, 58, proposed to Nathan, a former nurse, during a business trip to Paris – five months after he finalized his divorce from Donna Hanover in July after 20 years of marriage.

In interviews last year, Giuliani said Nathan gave him “tremendous emotional support through his treatment for prostate cancer and as he led New York City during the Sept. 11, 2001, terror attacks.”
Giuliani, 58, proposed to Nathan, a former nurse, during a business trip to Paris—five months after he finalized his divorce from Donna Hanover in July after 20 years of marriage.

In interviews last year, Giuliani said Nathan gave him "tremendous emotional support" through his treatment for prostate cancer and as he led New York City during the Sept. 11, 2001, terror attacks.
Giuliani, 58, proposed to Nathan, a former nurse, during a business trip to Paris five months after he finalized his divorce from Donna Hanover in July after 20 years of marriage.

In interviews last year, Giuliani said Nathan gave him "tremendous emotional support" through his treatment for prostate cancer and as he led New York City during the Sept. 11, 2001, terror attacks.
Giuliani, 58, proposed to Nathan, a former nurse, during a business trip to Paris five months after he finalized his divorce from Donna Hanover in July after 20 years of marriage. In interviews last year, Giuliani said Nathan gave him "tremendous emotional support" through his treatment for prostate cancer and as he led New York City during the Sept. 11, 2001, terror attacks.
Giuliani, 58, proposed to Nathan, a former nurse, during a business trip to Paris five months after he finalized his divorce from Donna Hanover in July after 20 years of marriage.

In interviews last year, Giuliani said Nathan gave him "tremendous emotional support" through his treatment for prostate cancer and as he led New York City during the Sept. 11, 2001, terror attacks.
Giuliani, 58, proposed to Nathan, a former nurse, during a business trip to Paris five months after he finalized his divorce from Donna Hanover in July after 20 years of marriage.

In interviews last year, Giuliani said Nathan gave him "tremendous emotional support" through his treatment for prostate cancer and as he led New York City during the Sept. 11, 2001, terror attacks.
Giuliani, 58, proposed to Nathan, a former nurse, during a business trip to Paris — five months after he finalized his divorce from Donna Hanover in July after 20 years of marriage.

In interviews last year, Giuliani said Nathan gave him “tremendous emotional support” through his treatment for prostate cancer and as he led New York City during the Sept. 11, 2001, terror attacks.
Information Extraction vs. Information Retrieval

• Information Retrieval returns a set of documents given a query.
• Information Extraction returns facts from documents
• E.g., What you search for in real estate advertisements:
  – Town/suburd. You might think easy, but:
    • Real estate agents: Coldwell Banker, Mosman
    • Phrases: Only 45 minutes from Parramatta
    • Multiple property ads have different suburbs in one ad
  – Money: want a range not a textual match
    • Multiple amounts: was $155K, now $145K
  – Bedrooms
    • Variations: br, bdr, beds, B/R
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
Supervised learning for NER

- Named entities are crucial to different IE and QA tasks
- For Named Entity Recognition (NER) (find and classify names in text), we can use the sequence labeling methods discussed previously (i.e., MEMM, CRF, RNN).

  
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fred</td>
<td>Smith</td>
<td>works</td>
<td>for</td>
<td>Time</td>
<td>inc.</td>
</tr>
<tr>
<td>B_PER</td>
<td>I_PER</td>
<td>O</td>
<td>O</td>
<td>B_ORG</td>
<td>B_ORG</td>
</tr>
</tbody>
</table>

- **Feature-based models**: the key is to design good feature sets to feed into the sequence labeling models (i.e., feature engineering with MEMM or CRF)
Features for NER

- identity of $w_i$, identity of neighboring words
- embeddings for $w_i$, embeddings for neighboring words
- part of speech of $w_i$, part of speech of neighboring words
- base-phrase syntactic chunk label of $w_i$ and neighboring words
- presence of $w_i$ in a gazetteer
- $w_i$ contains a particular prefix (from all prefixes of length $\leq 4$)
- $w_i$ contains a particular suffix (from all suffixes of length $\leq 4$)
- $w_i$ is all upper case
- word shape of $w_i$, word shape of neighboring words
- short word shape of $w_i$, short word shape of neighboring words
- presence of hyphen

Figure 17.5 Typical features for a feature-based NER system.

\[
\begin{align*}
\text{prefix}(w_i) &= \text{L} & \text{suffix}(w_i) &= \text{tane} \\
\text{prefix}(w_i) &= \text{L'} & \text{suffix}(w_i) &= \text{ane} \\
\text{prefix}(w_i) &= \text{L'O} & \text{suffix}(w_i) &= \text{ne} \\
\text{prefix}(w_i) &= \text{L'Oc} & \text{suffix}(w_i) &= \text{e} \\
\text{word-shape}(w_i) &= \text{X’Xxxxxxxxx} & \text{short-word-shape}(w_i) &= \text{X’Xx}
\end{align*}
\]
Features for NER

• Word shape features: Map words to simplified representation that encodes attributes such as length, capitalization, numerals, Greek letters, internal punctuation, etc.

<table>
<thead>
<tr>
<th>Name</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Varicella-zoster</td>
<td>Xx-xxx</td>
</tr>
<tr>
<td>mRNA</td>
<td>XXX</td>
</tr>
<tr>
<td>CPA1</td>
<td>XXXd</td>
</tr>
</tbody>
</table>

• Shorter word shape features: consecutive character types are removed (i.e., DC10-30 -> Xd-d, I.M.F -> X.X.X)

• Gazetteers: Lists of common names for different types
  – Millions of entries for locations with detailed geographical and political information ([www.geonames.org](http://www.geonames.org))
  – Lists of first names and surnames derived from its decadal census in the U.S ([www.census.gov](http://www.census.gov))
  – Typically implemented as a binary feature for each name list
  – Unfortunately, such lists can be difficult to create and maintain, and their usefulness varies considerably.
Deep learning for NER

**Figure 17.8** Putting it all together: character embeddings and words together a bi-LSTM sequence model. After Lample et al. (2016).
## Evaluation for NER systems

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>tim</td>
<td>cook</td>
<td>is</td>
<td>the</td>
<td>CEO</td>
<td>of</td>
<td>Apple</td>
</tr>
<tr>
<td>gold</td>
<td>B-PER</td>
<td>I-PER</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>B-ORG</td>
</tr>
<tr>
<td>system</td>
<td>B-PER</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>B-PER</td>
<td>O</td>
<td>B-ORG</td>
</tr>
</tbody>
</table>

### <start, end, type>

- **gold**
  - <1,2,PER>
  - <7,7,ORG>

- **system**
  - <1,1,PER>
  - <5,5,PER>
  - <7,7,ORG>
Supervised learning for Relation Extraction

• 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.
History of relations

• Relations were introduced in MUC-7 (1997)
  • 3 relations

• Extensively studied in ACE (2000 – 2007)
  • lots of training data

• Effectively included in KBP
  • Wikipedia infobox model
ACE Relations

• Several revisions of relation definitions
  • With goal of having a set of relations which can be 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>
KBP Slots

• Many KBP slots represent relations between entities:
  • Member_of
  • Employee_of
  • Country_of_birth
  • Countries_of_residence
  • Schools_attended
  • Spouse
  • Parents
  • Children ...

• Entities do not need to appear in the same sentence
• More limited training data
  • Encouraged semi-supervised methods
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
  – Longer-range (‘predicate-linked’) constructs
    • With a predicate disjoint from the arguments
      – Fred lived in New York
      – Fred and Mary got married
Methods for Relation Extraction (RE)

• Rule-based methods
  – Write rules to capture different types of relations

• Feature-based methods
  – Design feature sets for RE and send them to some statistical classifiers (i.e., MaxEnt, SVM)

• Kernel-based methods
  – Design kernels to compute similarities between pairs of entities and use them in kernel-based SVM

• Deep learning methods
  – Let deep learning learn the features for RE from data
Rule-based methods for RE: 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”
Lexicalized Dependency Paths

- using path in dependency tree between two entity mentions as the patterns for RE
- combines dependency types and lexical items
  - type = edge from governor to dependent
  - type-1 = edge from dependent to governor

`PERSON – nsubj-1:shot: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

- subject control
  - Harry planned to bake the cake  \( \rightarrow \) Harry planned (Harry baked the cake)
Supervised learning for RE

• Collect training data
  – Annotate corpus with entities and relations
  – For every pair of entities in a sentence
    • If linked by a relation, treat as positive training instance
    • If not linked, treat as a negative training instance

• Train model
  – For $n$ relation types, either
    • Binary (identification) model + $n$-way classifier model or
    • Unified $n+1$-way classifier
    • Either way, the dataset is very imbalanced toward the negative instances (“Other”)

• On test data
  – Apply entity classifier
  – Apply relation classifier to every pair of entities in same sentence

• Evaluate using Precision, Recall and F1
Supervised learning for RE

- The spokesman, reporting on the meeting, said IBM hired Fred Smith as the president.

Relation instances

- The spokesman, reporting on the meeting, said IBM hired Fred Smith as the president. -> Other
- The spokesman, reporting on the meeting, said IBM hired Fred Smith as the president. -> Other
- The spokesman, reporting on the meeting, said IBM hired Fred Smith as the president. -> Other
- The spokesman, reporting on the meeting, said IBM hired Fred Smith as the president. -> Employment
- The spokesman, reporting on the meeting, said IBM hired Fred Smith as the president. -> Employment
- The spokesman, reporting on the meeting, said IBM hired Fred Smith as the president. -> Other
Feature-based methods for RE

• Design a set of features, compute the values of such features for each instance, and send them statistical classifiers for classification

• Typical features:
  – Heads of entities
  – Types of entities
  – Distance between entities
  – Containment relations
  – Word sequence between entities
  – Individual words between entities
  – Dependency path
  – Individual words on dependency path

Zhou et al., 2005: Exploring Various Knowledge in Relation Extraction (ACL)
Ray Young, the chief financial officer of General Motors, said GM could not bail out Delphi

<table>
<thead>
<tr>
<th>Designed Features</th>
<th>Values</th>
<th>Designed Features</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>head word of M1</td>
<td>Ray_Young</td>
<td>last word in between</td>
<td></td>
</tr>
<tr>
<td>head word of M2</td>
<td>General_Motors</td>
<td>middle token sequence</td>
<td>, the chief financial officer of</td>
</tr>
<tr>
<td>first word before M1</td>
<td>nil</td>
<td>Shortest path connecting M1 and M2 in the dependency parsing tree</td>
<td>PERSON_appos_officer prep_of_ORGANIZATION</td>
</tr>
<tr>
<td>second word before M1</td>
<td>nil</td>
<td>entity type of M1</td>
<td>PERSON</td>
</tr>
<tr>
<td>first word after M2</td>
<td>,</td>
<td>entity type of M2</td>
<td>ORGANIZATION</td>
</tr>
<tr>
<td>second word after M2</td>
<td>said</td>
<td></td>
<td>....</td>
</tr>
<tr>
<td>first word in between</td>
<td></td>
<td></td>
<td>....</td>
</tr>
</tbody>
</table>
Features for RE: Brown Word Clustering

- The Brown algorithm (a hierarchical clustering algorithm):
  - initially assigns each word to its own cluster
  - repeatedly merges the two clusters which cause the least loss in average mutual information between adjacent clusters based on bigram statistics
  - by tracing the pairwise merging steps, one can obtain a word hierarchy which can be represented as a binary tree

- Use prefixes of the bit strings of the heads of the entity mentions as the features (i.e., HM1_WC2, HM2_WC4)

<table>
<thead>
<tr>
<th>Type</th>
<th>Baseline</th>
<th>PC4 (△)</th>
<th>Baseline</th>
<th>PC4 (△)</th>
<th>Baseline</th>
<th>PC4 (△)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP-ORG</td>
<td>75.4</td>
<td>77.2(+1.8)</td>
<td>79.8</td>
<td>81.5(+1.7)</td>
<td>77.6</td>
<td>79.3(+1.7)</td>
</tr>
<tr>
<td>PHYS</td>
<td>73.2</td>
<td>71.2(-2.0)</td>
<td>61.6</td>
<td>60.2(-1.4)</td>
<td>66.9</td>
<td>65.3(-1.7)</td>
</tr>
<tr>
<td>GPE-AFF</td>
<td>67.1</td>
<td>69.0(+1.9)</td>
<td>60.0</td>
<td>63.2(+3.2)</td>
<td>63.3</td>
<td>65.9(+2.6)</td>
</tr>
<tr>
<td>PER-SOC</td>
<td>88.2</td>
<td>83.9(-4.3)</td>
<td>58.4</td>
<td>61.0(+2.6)</td>
<td>70.3</td>
<td>70.7(+0.4)</td>
</tr>
<tr>
<td>DISC</td>
<td>79.4</td>
<td>80.6(+1.2)</td>
<td>42.9</td>
<td>46.0(+3.2)</td>
<td>55.7</td>
<td>58.6(+2.9)</td>
</tr>
<tr>
<td>ART</td>
<td>87.9</td>
<td>96.9(+9.0)</td>
<td>63.0</td>
<td>67.4(+4.4)</td>
<td>73.4</td>
<td>79.3(+5.9)</td>
</tr>
<tr>
<td>OTHER-AFF</td>
<td>70.6</td>
<td>80.0(+9.4)</td>
<td>41.4</td>
<td>41.4(0.0)</td>
<td>52.2</td>
<td>54.6(+2.4)</td>
</tr>
</tbody>
</table>

Sun at al., 2011: Semi-supervised Relation Extraction with Large-scale Word Clustering (ACL)
Features for RE: Word Embeddings

- Generalizing the head words of the entity mentions seems to be very helpful for RE.
- Use word embeddings to achieve such generalization (i.e., using the dimensions of the word embeddings of the heads as the features).
- Without regularization:

<table>
<thead>
<tr>
<th>System</th>
<th>In-domain</th>
<th>bc</th>
<th>cts</th>
<th>wl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline(B)</td>
<td>51.4</td>
<td>49.7</td>
<td>41.5</td>
<td>36.6</td>
</tr>
<tr>
<td>B+WC10</td>
<td>52.3(+0.9)</td>
<td>50.8(+1.1)</td>
<td>45.7(+4.2)</td>
<td>39.6(+3)</td>
</tr>
<tr>
<td>B+WC</td>
<td>53.7(+2.3)</td>
<td>52.8(+3.1)</td>
<td>46.8(+5.3)</td>
<td>41.7(+5.1)</td>
</tr>
<tr>
<td>B+ED</td>
<td>54.1(+2.7)</td>
<td>52.4(+2.7)</td>
<td>46.2(+4.7)</td>
<td>42.5(+5.9)</td>
</tr>
<tr>
<td>B+WC+ED</td>
<td>55.5(+4.1)</td>
<td>53.8(+4.1)</td>
<td>47.4(+5.9)</td>
<td>44.7(+8.1)</td>
</tr>
</tbody>
</table>

- With regularization:

<table>
<thead>
<tr>
<th>System</th>
<th>In-domain</th>
<th>bc</th>
<th>cts</th>
<th>wl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline(B)</td>
<td>56.2</td>
<td>55.5</td>
<td>48.7</td>
<td>42.2</td>
</tr>
<tr>
<td>B+WC10</td>
<td>57.5(+1.3)</td>
<td>57.3(+1.8)</td>
<td>52.3(+3.6)</td>
<td>45.0(+2.8)</td>
</tr>
<tr>
<td>B+WC</td>
<td>58.9(+2.7)</td>
<td>58.4(+2.9)</td>
<td>52.8(+4.1)</td>
<td>47.3(+5.1)</td>
</tr>
<tr>
<td>B+ED</td>
<td>58.9(+2.7)</td>
<td>59.5(+4.0)</td>
<td>52.6(+3.9)</td>
<td>48.6(+6.4)</td>
</tr>
<tr>
<td>B+WC+ED</td>
<td>59.4(+3.2)</td>
<td>59.8(+4.3)</td>
<td>52.9(+4.2)</td>
<td>49.7(+7.5)</td>
</tr>
</tbody>
</table>
Kernel-based methods for RE

- Goal is to find training examples similar to test case
  - Need similarity metrics between pairs of relation instances
  - Determining similarity through features is awkward
  - Better to define a similarity measure directly: a kernel function

- Kernels can be used directly by
  - SVMs
  - Memory-based learners (k-nearest-neighbor)

- For RE, kernels defined over
  - Strings
  - Parse or Dependency Trees
String kernels

- Two strings are more similar if they share more substrings

\[ k(s_i, s_j) = \sum_n c_n k_n(s_i, s_j) \]

\[ k_n(s_i, s_j) = \sum_{u \in \Sigma^n} \sum_{u=\sigma_i} \sum_{u=\sigma_j} \lambda^{l(o_{s_i})} \lambda^{l(p_{s_j})} \]

- Many variants are possible

Linear combination parameter
Decaying factor \(0 < \lambda \leq 1\)
Sets of strings of length \(n\)
Length of the string
Tree kernels

- Compute the number of common subtrees:
  
  \[
  TK_\sigma(T_1, T_2) = \sum_{n_1 \in N_1, n_2 \in N_2} \Delta(n_1, n_2)
  \]

  where \(\Delta(n_1, n_2)\) is computed by:
  
  i) if \(n_1\) and \(n_2\) have different productions: \(\Delta(n_1, n_2) = 0\); else
  ii) if \(n_1\) and \(n_2\) are pre-terminals: \(\Delta(n_1, n_2) = \lambda\); else
  iii) \(\Delta(n_1, n_2) = \lambda \prod_{j=1}^{nc(n_1)} (1 + \Delta(ch(n_1, j), ch(n_2, j)))\)

- \(T_1, T_2\) can be either constituent or dependency trees. The trees can be pruned to minimally cover the two entity mention of interest.
- Can incorporate with word clusters and word embeddings

\[
\Phi \begin{cases} 
\text{[Diagram 1]} 
\end{cases} \begin{cases} 
\text{[Diagram 2]} 
\end{cases} = 0.95
\]
However, acetaminophen has been demonstrated to produce symptoms of anaphylaxis, including hypotension, in sensitive individuals.

The dependency tree

The constituent tree
Deep learning for RE

• Avoid feature or kernel design for RE

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Features</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxEnt</td>
<td>POS, WordNet, morphological features, noun compound system, thesauri, Google n-grams</td>
<td>77.6</td>
</tr>
<tr>
<td>SVM</td>
<td>POS, WordNet, prefixes and other morphological features, dependency parse, Levin classes, PropBank, FrameNet, NomLex-Plus, Google n-grams, paraphrases, TextRunner</td>
<td>82.2</td>
</tr>
<tr>
<td>CNN (Zeng et al., 2014)</td>
<td>WordNet</td>
<td>82.7</td>
</tr>
<tr>
<td>CNN (Nguyen and Grishman, 2015a)</td>
<td>-</td>
<td>82.8</td>
</tr>
</tbody>
</table>

SemEval 2010 Dataset

Nguyen and Grishman, 2015
To inform the models about the two entity mentions of interest, we introduce (relative) position embeddings (randomly initialized and updated during training)

<table>
<thead>
<tr>
<th>dist from m1</th>
<th>0</th>
<th>1</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>dist from m2</td>
<td>-8</td>
<td>-7</td>
<td>-6</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
</tr>
</tbody>
</table>

[The Big Sleep] is a 1946 film noir directed by [Howard Hawks]
Deep learning for RE

- Can also incorporate syntax into deep learning models for RE: to identify important context words (i.e., via the dependency paths) or to guide the computational flows of the neural network models.

**Recursive neural networks:**
- building the networks based on the constituent trees
- the binarized constituent subtree

Cat et al., Bidirectional Recurrent Convolutional Neural Network for Relation Classification (ACL 2016)

Socher et al., Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank (EMNLP 2013)
Syntactic Structures for Relation Extraction

- Graph Convolutional Neural Network (GCN) over dependency trees for RE (a recent state-of-the-art approach for RE) (Zhang et al., 2018)

He faces up to 30 years in prison
Other types of learning for IE

• **Supervised learning**
  – All training data is labeled

• **Semi-supervised learning**
  – Part of training data is labeled (‘the seed’) (the rest is unlabeled)
  – Make use of redundancies to learn labels of additional data, then train model
  – Co-training
  – Reduces amount of data which must be hand-labeled to achieve a given level of performance

• **Active learning**
  – Start with partially labeled data
  – System selects additional ‘informative’ examples for user to label
Semi-supervised learning

\[ L = \text{labeled data} \]
\[ U = \text{unlabeled data} \]

1. \[ L = \text{seed} \]
   -- repeat 2-4 until stopping condition is reached
2. \[ C = \text{classifier trained on } L \]
3. \[ \text{Apply } C \text{ to } U. \]
   \[ N = \text{most confidently labeled items} \]
4. \[ L += N; \ U -= N \]
Confidence

How to estimate confidence?

• Binary probabilistic classifier
  – Confidence = | \( P - 0.5 \) | * 2

• N-ary probabilistic classifier
  – Confidence = \( P_1 - P_2 \)
  where
    \( P_1 = \) probability of most probable label
    \( P_2 = \) probability of second most probable label

• SVM
  – Distance from the separating hyperplane
Co-training

- Two ‘views’ of data (subsets of features)
  - Producing two classifiers $C_1(x)$ and $C_2(x)$
- Ideally
  - Independent
  - Each sufficient to classify data
- Apply classifiers in alternation (or in parallel)
  1. $L =$ seed
     -- repeat 2-7 until stopping condition is reached
  2. $C_1 =$ classifier trained on $L$
  3. Apply $C_1$ to $U$.
     $N =$ most confidently labeled items
  4. $L += N; U -= N$
  5. $C_2 =$ classifier trained on $L$
  6. Apply $C_2$ to $U$.
     $N =$ most confidently labeled items
  7. $L += N; U -= N$
Problems with semi-supervised learning

• When to stop?
  • $U$ is exhausted
  • Reach performance goal using held-out labeled sample
  • After fixed number of iterations based on similar tasks

• Poor confidence estimates
  • Errors from poorly-chosen data rapidly magnified
Learning Names

• We have discussed hand-coded rules and supervised models (HMM, MEMM, CRF, RNN) for NER [named entity recognition]

• We will now consider
  • Semi-supervised models
  • Active learning
Semi-supervised NER

• Annotating a large corpus to train a high-performance NER is fairly expensive

• We can use the idea of name consistency across documents to train an NER using
  – A smaller annotated corpus
  – A large unannotated corpus
Co-training for NER

- We can split the features for NER into two sets:
  - Spelling features
    (the entire name + tokens in the name)
  - Context features
    (left and right contexts + syntactic context)

- Start with a seed
  - E.g., some common unambiguous full names

- Iteratively grow seed, alternatively applying spelling and context models and adding most -confidently-labeled instances to seed
Co-training for NER

- Seed
- Build context model
  - Apply context model
  - Add most confident exs to labeled set
- Apply spelling model
  - Build spelling model
  - Add most confident exs to labeled set
Name co-training: results

- 3 classes: person, organization, location (and ‘other’)
- Data: 1M sentences of news
- Seed:
  - New York, California, U.S. → location
  - contains(Mr.) → person
  - Microsoft, IBM → organization
  - contains(Incorporated) → organization
- Took names appearing with appositive modifier or as complement of preposition (88K name instances)
- Accuracy: 83%
- Clean accuracy (ignoring names not in one of the 3 categories): 91%
- (Collins and Singer 1999)
Semi-supervised NER: when to stop

• Semi-supervised NER labels a few more examples at every iteration
  – It stops when it runs out of examples to label

• This is fine if
  – Names are easily identified (e.g., by capitalization in English)
  – Most names fall into one of the categories being trained (e.g., people, organizations, and locations for news stories)
Semi-supervised NER: semantic drift

- Semi-supervised NER doesn’t work so well if
  - The set of names is hard to identify
    - Monocase languages
    - Extended name sets including lower-case terms
  - The categories being trained cover only a small portion of the set of names

- The result is *semantic drift* and *semantic spread*
  - The name categories gradually grow to include related terms
Fighting Semantic Drift

• We can fight drift by training a larger, more inclusive set of categories
  – Including ‘negative’ categories
    • Categories we don’t really care about but include to compete with the original categories
  – These negative categories can be built
    • By hand (Yangarber et al. 2003)
    • Or automatically (McIntosh 2010)
Active Learning

• For supervised learning, we typically annotate text data sequentially

• Not necessarily the most efficient approach
  • Most natural language phenomena have a Zipfian distribution ... a few very common constructs and lots of infrequent constructs
  • After you have annotated “Spain” 50 times as a location, the NER model is little improved by annotating it one more time

• We want to select the most informative examples and present them to the annotator
  • The data which, if labeled, is most likely to reduce NER error
How to select informative examples?

• Uncertainty-based sampling
  – For binary classifier
    • For MaxEnt, probability near 50%
    • For SVM, data near separating hyperplane
  – For n-ary classifier, data with small margin

• Committee-based sampling
  – Data on which committee members disagree
  – (co-testing ... use two classifiers based on independent views)
Representativeness

• It’s more helpful to annotate examples involving less common features
  • Weighting these features correctly will have a larger impact on error rate

• So we rank examples by frequency of features in the entire corpus
Batching and Diversity

- Each iteration of active learning involves running classifier on (a large) unlabeled corpus
  - This can be quite slow
  - Meanwhile annotator is waiting for something to annotate

- So we run active learning in batches
  - Select best $n$ examples to annotate each time
  - But all items in a batch are selected using the same criteria and same system state, and so are likely to be similar

- To avoid example overlap, we impose a diversity requirement with a batch: limit maximum similarity of examples within a batch
  - Compute similarity based on example feature vectors
Simulated Active Learning

• True active learning experiments are
  – Hard to reproduce
  – Very time consuming

• So most experiments involve simulated active learning:
  – “unlabeled” data has really been labeled, but the labels have been hidden
  – When data is selected, labels are revealed
  – Disadvantage: “unlabeled” data can’t be so bit

• This leads us to ignore lots of issues of true active learning:
  – An annotation unit of one sentence or even one token may not be efficient for manual annotation
  – So reported speed-ups may be optimistic (typical reports reduce by half the amount of data to achieve a given NER accuracy)
Limitations

• Cited performance is for well matched training and test
  • Same domain
  • Same source
  • Same epoch
  – Performance deteriorates rapidly if less matched
  • NER trained on Reuters (F=91), tested on Wall Street Journal (F=64) [Ciaramita and Altun 2003]
  – Work on NER adaptation is vital

• Adding rarer classes to NER is difficult
  – Supervised learning inefficient
  – Semi-supervised learning is subject to semantic drift
Semi-supervised methods for RE

- Preparing training data for relations is more costly than for names
  - Must annotate entities and relations

- So there is a strong motivation to minimize training data through semi-supervised methods

- As for names, we will adopt a co-training approach:
  - Feature set 1: the two entities
  - Feature set 2: the contexts between the entities

- We will limit the bootstrapping
  - to a specific pair of entity types
  - and to instances where both entities are named
Semi-supervised learning for RE

- **Seed:**
  - *Moby Dick, Herman Melville*

- **Contexts for seed:**
  - ... wrote ...
  - ... is the author of ...

- **Other pairs appearing in these contexts**
  - *Animal Farm, George Orwell*
  - *Don Quixote, Miguel de Cervantes*

- **Additional contexts ...**
Co-training for relations

- seed
- Find occurrences of seed tuples
- Tag entities
- Generate new seed tuples
- Generate extraction patterns
Ranking contexts

- If relation $R$ is functional, and $[X, Y]$ is a seed, then $[X, Y'], Y' \neq Y$, is a negative example.

- Confidence of pattern $P$

  $$Conf(P) = \frac{P.positive}{P.positive + P.negative}$$

- Where

  $P.positive = \text{number of positive matches to pattern } P$

  $P.negative = \text{number of negative matches to pattern } P$
Ranking pairs

• Once a confidence has been assigned to each pattern, we can assign a confidence to each new pair based on the patterns in which it appears
  
  – Confidence of best pattern
  – Combination assuming patterns are independent

\[
Conf(X,Y) = 1 - \prod_{P \in \text{contexts of } (X,Y)} (1 - Conf(P))
\]
Semantic drift

- Ranking / filtering quite effective for functional relations (book ➔ author, company ➔ headquarters)
  - But expansion may occur into other relations generally implied by seed ('semantic drift')
    - Ex: from governor ➔ state governed_to person ➔ state born_in

- Precision poor without functional property
Distant supervision

• Sometimes a large data base is available involving the type of relation to be extracted
  • A number of such public data bases are now available, such as FreeBase and YAGO
• Text instances corresponding to some of the data base instances can be found in a large corpus or from the Web
• Together these can be used to train a relation classifier

https://developers.google.com/freebase/
Distant supervision

Ronaldinho

From Wikipedia, the free encyclopedia

"Ronaldinho Gaucho" redirects here. For the comic strip based on him, see Ronaldinho Gaucho (comic strip). For other uses, see Ronaldinho (disambiguation).

This name uses Portuguese naming customs: the first or maternal family name is Assis and the second or paternal family name is Moreira.

Ronaldo de Assis Moreira (born 21 March 1980), commonly known as Ronaldinho Gaucho (Brazilian Portuguese: [ʁonawdĩjˈguʃu]) or simply Ronaldinho,[note 1] is a Brazilian former professional footballer and ambassador for Barcelona,[4] He played mostly as an attacking midfielder, but was also deployed as a forward or a winger. He played the bulk of his career at European clubs Paris Saint-Germain, Barcelona and A.C. Milan as well as playing for the Brazilian national team. Often considered one of the best players of his generation and regarded by many as one of the greatest of all time,[note 2] Ronaldinho won two FIFA World Player of the Year awards and a Ballon d'Or. He was renowned for his technical skills and creativity; due to his agility, pace and dribbling ability, as well as his use of tricks, feints, overhead kicks, no-look passes and accuracy from free-kicks.

Ronaldinho made his career debut for Grêmio, in 1998. At age 20, he moved to Paris Saint-Germain in France before signing for Barcelona in 2003. In his second season with Barcelona, he won his first FIFA World Player of the Year award, as Barcelona won La Liga. The season that followed is considered one of the best in his career as he was instrumental in Barcelona winning the UEFA Champions League, their first in fourteen years, as well as another La Liga title, giving Ronaldinho his first career double. After scoring two spectacular solo goals in El Clásico, Ronaldinho became the second Barcelona player, after Diego Maradona in 1983, to receive a standing ovation from Real Madrid fans at the Santiago Bernabéu. Ronaldinho also received his second FIFA World Player of the Year award, as well as the Ballon d'Or.

Ronaldinho in 2019

<table>
<thead>
<tr>
<th>Personal information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Full name</strong></td>
</tr>
<tr>
<td><strong>Date of birth</strong></td>
</tr>
<tr>
<td><strong>Place of birth</strong></td>
</tr>
<tr>
<td><strong>Height</strong></td>
</tr>
<tr>
<td><strong>Playing position</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Youth career</th>
</tr>
</thead>
<tbody>
<tr>
<td>1987–1998</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Senior career*</th>
<th></th>
<th>Apps (Gls)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Years</strong></td>
<td><strong>Team</strong></td>
<td></td>
</tr>
<tr>
<td>1998–2001</td>
<td>Grêmio</td>
<td>52 (21)</td>
</tr>
<tr>
<td>2003–2008</td>
<td>Barcelona</td>
<td>145 (70)</td>
</tr>
<tr>
<td>2008–2011</td>
<td>A.C. Milan</td>
<td>76 (20)</td>
</tr>
<tr>
<td>2011–2012</td>
<td>Flamengo</td>
<td>33 (15)</td>
</tr>
<tr>
<td>2012–2014</td>
<td>Atlético Mineiro</td>
<td>48 (16)</td>
</tr>
<tr>
<td>2014–2015</td>
<td>Querétaro</td>
<td>25 (8)</td>
</tr>
<tr>
<td>2015</td>
<td>Fluminense</td>
<td>7 (0)</td>
</tr>
</tbody>
</table>
Distant supervision: approach

- **Given:**
  - Data base for relation $R$
  - Corpus containing information about relation $R$
- Collect $<X, Y>$ pairs from data base relation $R$
- Collect sentences in corpus containing both $X$ and $Y$
  - These are positive training examples
- Collect sentences in corpus containing $X$ and some $Y'$ with the same entity type as $Y$ such that $<X, Y'>$ is not in the data base
  - These are negative training examples
- Use examples to train classifier which operates on pairs of entities
Distant supervision: limitations

• The training data produced through distant supervision may be quite noisy:

  – If a pair \(< X, Y >\) is involved in multiple relations, \(R < X, Y >\) and \(R' < X, Y >\) and the data base represents relation \(R\), the text instance may represent relation \(R'\), yielding a false positive training instance.

  – If many \(< X, Y >\) pairs are involved, the classifier may learn the wrong relation.

  – If a relation is incomplete in the data base ... for example, if \(\text{resides\_in} < X, Y >\) contains only a few of the locations where a person has resided ... then we will generate many false negatives, possibly leading the classifier to learn no relation at all.
Multi-label multi-instance learning for distant supervision (MIML)

To reduce noise in distant supervision:

- Group instances (sentences) corresponding to the same entity pair \(<X, Y>\) in the knowledge base into a group (a bag of instances).
- Each bag can be assigned to multiple relations to capture the possible relations between X and Y in the knowledge base.
- People might just do multi-instance learning (i.e., a single label for a bag).

Surdeanu et al., Multi-instance Multi-label Learning for Relation Extraction (EMNLP 2012)
Multiple-instance learning for distant supervision

\[ s = \sum_{i} \alpha_i x_i \]
\[ \alpha_i = \frac{\exp(e_i)}{\sum_{k} \exp(e_k)} \]
\[ e_i = x_iA \]
\[ o = Ms + d \]

\[ p(r | S, \theta) = \frac{\exp(o_r)}{\sum_{k=1}^{n_r} \exp(o_k)} \]