Information Extraction: Coreference and Relation Extraction

Lecture #22

Introduction to Natural Language Processing
CMPSCI 585, Spring 2004
University of Massachusetts, Amherst

Andrew McCallum

What is “Information Extraction”

As a family of techniques:

Information Extraction
 segmenetation + classification + association + clustering

Microsoft Corporation
CEO
Bill Gates

Microsoft
Gates

Bill Veghte
Microsoft
VP

Richard Stallman
Founder
Free Software Foundation

Coreference Resolution

AKA “record linkage”, “database record deduplication”, “citation matching”, “object correspondence”, “identity uncertainty”

Input
News article, with named-entity “mentions” tagged

Today Secretary of State Colin Powell met with
Condeleeza Rice...
Mr. Powell...
President Bush...

Output
Number of entities, N = 3

#1 Secretary of State Colin Powell
#2 Condeleeza Rice
#3 President Bush

Coreference Resolution

AKA “record linkage”, “database record deduplication”, “citation matching”, “object correspondence”, “identity uncertainty”

Input
News article, with named-entity “mentions” tagged

Today Secretary of State Colin Powell met with
Condeleeza Rice...
Mr. Powell...
President Bush...

Output
Number of entities, N = 3

#1 Secretary of State Colin Powell
#2 Condeleeza Rice
#3 President Bush

Main Points

Co-reference
- How to cast as classification [Cardie]
- Measures of string similarity [Cohen]
- Scaling up [McCallum et al]

Relation extraction
- With augmented grammar [Miller et al 2000]
- With joint inference [Roth & Yih]
- Semi-supervised [Brin]

Inside the Traditional Solution

Pair-wise Affinity Metric

<table>
<thead>
<tr>
<th>Mention (3)</th>
<th>Y/N?</th>
<th>Mention (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mr Powell</td>
<td></td>
<td>Powell</td>
</tr>
</tbody>
</table>

| N  | Two words in common | 28 |
| Y  | One word in common | 13 |
| Y  | “Normalized” mentions are string identical | 39 |
| Y  | Capitalized word in common | 17 |
| Y  | > 50% character tri-gram overlap | 19 |
| Y  | In same sentence | 9 |
| Y  | Within two sentences | 8 |
| N  | More than three words apart | 1 |
| Y  | “Hobbs Distance” < 3 | 11 |
| N  | Number of entities in between two mentions = 0 | 12 |
| N  | Number of entities in between two mentions > 4 | 1 |
| Y  | Font matches | 1 |
| Y  | Default | -19 |

Overall Score = 98 > threshold=0
Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

SAN SALVADOR, 15 JAN 90 (ACAN-EFE) -- ARMANDO CALDERON SOL, PRESIDENT OF THE NATIONALIST REPUBLICAN ALLIANCE (ARENA), THE RULING SALVADORAN PARTY, TODAY CALLED FOR AN INVESTIGATION INTO ANY POSSIBLE CONNECTION BETWEEN THE MILITARY PERSONNEL IMPLICATED IN THE ASSASSINATION OF JESUIT PRIESTS.

"IT IS SOMETHING SO HORRENDOUS, SO MONSTROUS, THAT WE MUST INVESTIGATE THE POSSIBILITY THAT THE FMLN (FARABUNDO MARTI NATIONAL LIBERATION FRONT) STAGED THESE MURDERS TO DISCREDIT THE GOVERNMENT," CALDERON SOL SAID.

SALVADORAN PRESIDENT ALFREDO CRISTIANI IMPLICATED FOUR OFFICERS, INCLUDING ONE COLONEL, AND FIVE MEMBERS OF THE ARMED FORCES IN THE ASSASSINATION OF SIX JESUIT PRIESTS AND TWO WOMEN ON 16 NOVEMBER AT THE CENTRAL AMERICAN UNIVERSITY.
IE Example: Coreference
SAN SALVADOR, 15 JAN 90 (ACAN-EFE) -- (TEXT) ARMANDO CALDERON SOL, PRESIDENT OF THE NATIONALIST REPUBLICAN ALLIANCE (ARENA), THE RULING SALVADORAN PARTY, TODAY CALLED FOR AN INVESTIGATION INTO ANY POSSIBLE CONNECTION BETWEEN THE MILITARY PERSONNEL IMPLICATED IN THE ASSASSINATION OF JESUIT PRIESTS.

"IT IS SOMETHING SO HORRENDOUS, SO MONSTROUS, THAT WE MUST INVESTIGATE THE POSSIBILITY THAT THE FMLN (FARABUNDO MARTI NATIONAL LIBERATION FRONT) STAGED THESE MURDERS TO DISCREDIT THE GOVERNMENT," CALDERON SOL SAID.

SALVADORAN PRESIDENT ALFREDO CRISTIANI IMPLICATED FOUR OFFICERS, INCLUDING ONE COLONEL, AND FIVE MEMBERS OF THE ARMED FORCES IN THE ASSASSINATION OF SIX JESUIT PRIESTS AND TWO WOMEN ON 16 NOVEMBER AT THE CENTRAL AMERICAN UNIVERSITY.

Why It’s Hard
• No single source is a completely reliable indicator
  – number agreement
  • the assassination = these murders
• Identifying each of these features automatically, accurately, and in context, is hard
• Coreference resolution subsumes the problem of pronoun resolution…

Many sources of information play a role
– head noun matches
  • IBM executives = the executives
– syntactic constraints
  • John helped himself to…
  • John helped him to…
– number and gender agreement
– discourse focus, recency, syntactic parallelism, semantic class, world knowledge, …

A Machine Learning Approach
• Classification
  – given a description of two noun phrases, $NP_i$ and $NP_j$, classify the pair as coreferent or not coreferent

A Machine Learning Approach
• Clustering
  – coordinates pairwise coreference decisions

Machine Learning Issues
• Training data creation
• Instance representation
• Learning algorithm
• Clustering algorithm
Supervised Inductive Learning

Examples of NP pairs (features + class)

ML Algorithm

(novel) pair of NPs
(features) label
Examples of NP pairs (features + class)
ML Algorithm
Concept description (program) label

Training Data Creation

• Creating training instances
  – texts annotated with coreference information
  – one instance inst(NPi, NPj) for each pair of NPs
    • assumption: NPj precedes NPi
    • feature vector: describes the two NPs and context
    • class value:
      coref pairs on the same coreference chain
      not coref otherwise

Learning Algorithm

• RIPPER (Cohen, 1995)
  C4.5 (Quinlan, 1994)
  – rule learners
    • input: set of training instances
    • output: coreference classifier
  • Learned classifier
    • input: test instance (represents pair of NPs)
    • output: classification
      confidence of classification

Clustering Algorithm

• Best-first single-link clustering
  – Mark each NPj as belonging to its own class:
    NPj \in c_j
  – Proceed through the NPs in left-to-right order.
    • For each NP, NPj, create test instances, inst(NPi, NPj),
      for all of its preceding NPs, NPi.
    • Select as the antecedent for NPj the highest-confidence
      coreferent NP, NPe, according to the coreference
      classifier (or none if all have below .5 confidence);
      Merge c_j and c_e.

Evaluation

• MUC-6 and MUC-7 coreference data sets
• documents annotated w.r.t. coreference
• 30 + 30 training texts (dry run)
• 30 + 20 test texts (formal evaluation)
• scoring program
  – recall
  – precision
  – F-measure: 2PR/(P+R)
Baseline Results

<table>
<thead>
<tr>
<th></th>
<th>MUC-6</th>
<th>MUC-7</th>
<th></th>
<th>MUC-6</th>
<th>MUC-7</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
<td>F</td>
<td>R</td>
<td>P</td>
<td>F</td>
<td>R</td>
<td>P</td>
<td>F</td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Baseline</td>
<td>40.7</td>
<td>73.5</td>
<td>52.4</td>
<td>27.2</td>
<td>86.3</td>
<td>41.3</td>
<td>40.7</td>
<td>73.5</td>
<td>52.4</td>
<td>27.2</td>
<td>86.3</td>
</tr>
<tr>
<td>Worst MUC System</td>
<td>36</td>
<td>44</td>
<td>40</td>
<td>52.5</td>
<td>21.4</td>
<td>30.4</td>
<td>36</td>
<td>44</td>
<td>40</td>
<td>52.5</td>
<td>21.4</td>
</tr>
<tr>
<td>Best MUC System</td>
<td>59</td>
<td>72</td>
<td>65</td>
<td>56.1</td>
<td>68.8</td>
<td>61.8</td>
<td>59</td>
<td>72</td>
<td>65</td>
<td>56.1</td>
<td>68.8</td>
</tr>
</tbody>
</table>

Problem 1

- Coreference is a rare relation
  - skewed class distributions (2% positive instances)
  - remove some negative instances

Problem 2

- Coreference is a discourse-level problem
  - different solutions for different types of NPs
  - proper names: string matching and aliasing
  - inclusion of “hard” positive training instances
  - positive example selection: selects easy positive training instances (cf. Harabagiu et al. (2001))

Problem 3

- Coreference is an equivalence relation
  - loss of transitivity
  - need to tighten the connection between classification and clustering
  - prune learned rules w.r.t. the clustering-level coreference scoring function

Results

<table>
<thead>
<tr>
<th></th>
<th>MUC-6</th>
<th>MUC-7</th>
<th></th>
<th>MUC-6</th>
<th>MUC-7</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
<td>F</td>
<td>R</td>
<td>P</td>
<td>F</td>
<td>R</td>
<td>P</td>
<td>F</td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Baseline</td>
<td>40.7</td>
<td>73.5</td>
<td>52.4</td>
<td>27.2</td>
<td>86.3</td>
<td>41.3</td>
<td>40.7</td>
<td>73.5</td>
<td>52.4</td>
<td>27.2</td>
<td>86.3</td>
</tr>
<tr>
<td>NEG-SELECT</td>
<td>46.5</td>
<td>67.8</td>
<td>55.2</td>
<td>37.4</td>
<td>58.7</td>
<td>48.0</td>
<td>46.5</td>
<td>67.8</td>
<td>55.2</td>
<td>37.4</td>
<td>58.7</td>
</tr>
<tr>
<td>POS-SELECT</td>
<td>53.1</td>
<td>80.8</td>
<td>64.1</td>
<td>41.1</td>
<td>78.0</td>
<td>53.8</td>
<td>53.1</td>
<td>80.8</td>
<td>64.1</td>
<td>41.1</td>
<td>78.0</td>
</tr>
<tr>
<td>NEG-SELECT + POS-SELECT</td>
<td>63.4</td>
<td>76.3</td>
<td>69.3</td>
<td>59.5</td>
<td>65.1</td>
<td>61.0</td>
<td>63.4</td>
<td>76.3</td>
<td>69.3</td>
<td>59.5</td>
<td>65.1</td>
</tr>
<tr>
<td>NEG-SELECT + POS-SELECT + RULE-SELECT</td>
<td>63.3</td>
<td>76.9</td>
<td>69.5</td>
<td>54.2</td>
<td>76.3</td>
<td>63.5</td>
<td>63.3</td>
<td>76.9</td>
<td>69.5</td>
<td>54.2</td>
<td>76.3</td>
</tr>
</tbody>
</table>

• Ultimately: large increase in F-measure, due to gains in recall

Comparison with Best MUC Systems

<table>
<thead>
<tr>
<th></th>
<th>MUC-6</th>
<th>MUC-7</th>
<th></th>
<th>MUC-6</th>
<th>MUC-7</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
<td>F</td>
<td>R</td>
<td>P</td>
<td>F</td>
<td>R</td>
<td>P</td>
<td>F</td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>NEG-SELECT + POS-SELECT + RULE-SELECT</td>
<td>63.3</td>
<td>76.9</td>
<td>69.5</td>
<td>54.2</td>
<td>76.3</td>
<td>63.5</td>
<td>63.3</td>
<td>76.9</td>
<td>69.5</td>
<td>54.2</td>
<td>76.3</td>
</tr>
<tr>
<td>Best MUC System</td>
<td>59</td>
<td>72</td>
<td>65</td>
<td>56.1</td>
<td>68.8</td>
<td>61.8</td>
<td>59</td>
<td>72</td>
<td>65</td>
<td>56.1</td>
<td>68.8</td>
</tr>
</tbody>
</table>
Supervised ML for NP Coreference

- Good performance compared to other systems, but...lots of room for improvement
  - Common nouns < pronouns < proper nouns
  - Tighter connection between classification and clustering is possible
    - Rich Caruana’s ensemble methods
    - Statistical methods for learning probabilistic relational models (Getoor et al., 2001; Lafferty et al., 2001; Taskar et al., 2003; McCallum and Wellner, 2003).
  - Need additional data sets
    - New release of ACE data from Penn’s LDC
    - General problem: reliance on manually annotated data…

Main Points

Co-reference
- How to cast as classification [Cardie]
- Measures of string similarity [Cohen]
- Scaling up [McCallum et al]

Relation extraction
- With augmented grammar [Miller et al 2000]
- With joint inference [Roth & Yih]
- Semi-supervised [Brin]

Record linkage: definition

- Record linkage: determine if pairs of data records describe the same entity
  - I.e., find record pairs that are co-referent
  - Entities: usually people (or organizations or…)
  - Data records: names, addresses, job titles, birth dates, ...
- Main applications:
  - Joining two heterogeneous relations
  - Removing duplicates from a single relation

Record linkage: terminology

- The term “record linkage” is possibly co-referent with:
  - For DB people: data matching, merge/purge, duplicate detection, data cleansing, ETL (extraction, transfer, and loading), de-duping
  - For AI/ML people: reference matching, database hardening, object consolidation,
  - In NLP: co-reference/anaphora resolution
  - Statistical matching, clustering, language modeling, ...

Finding a technical paper c. 1995

- Start with citation:
- Find author’s institution (w/ INSPEC)
- Find web host (w/ NETFIND)
- Find author’s home page and (hopefully) the paper by browsing

The data integration problem
String distance metrics: overview

- Term-based (e.g. TF-IDF as in WHIRL)
  - Distance depends on set of words contained in both s and t.
- Edit-distance metrics
  - Distance is shortest sequence of edit commands that transform s to t.
- Pair HMM based metrics
  - Probabilistic extension of edit distance
- Other metrics

String distance metrics: term-based

- Advantages:
  - Exploits frequency information
  - Efficiency: Finding \{ t : \text{sim}(t,s)>k \} is sublinear!
  - Alternative word orderings ignored
- Disadvantages:
  - Sensitive to spelling errors (William Cohen vs. William Cohen)
  - Sensitive to abbreviations (Univ. vs University)
  - Alternative word orderings ignored

Jaccard Distance

\[
\text{Jaccard Score} = \frac{|S \cap T|}{|S \cup T|} = \frac{3}{7}
\]

String distance metrics: Levenshtein

- Edit-distance metrics
  - Distance is shortest sequence of edit commands that transform s to t.
  - Simplest set of operations:
    - Copy character from s over to t
    - Delete a character in s (cost 1)
    - Insert a character in t (cost 1)
    - Substitute one character for another (cost 1)
  - This is "Levenshtein distance"

Levenshtein distance - example
Computing Levenshtein distance - 1

\[ D(i,j) = \text{score of best alignment from } s_1..s_i \text{ to } t_1..t_j \]

\[ = \min \begin{cases} 
D(i-1,j-1), & \text{if } s_i = t_j \\
D(i-1,j) + 1, & \text{if } s_i \neq t_j \\
D(i,j-1) + 1 \\
\end{cases} \]

//copy
//substitute
//insert
//delete

Computing Levenshtein distance - 2

\[ D(i,j) = \text{score of best alignment from } s_1..s_i \text{ to } t_1..t_j \]

\[ = \min \begin{cases} 
D(i-1,j-1) + d(s_i,t_j), & \text{if } s_i = t_j \\
D(i-1,j) + 1, & \text{if } s_i \neq t_j \\
D(i,j-1) + 1 \\
\end{cases} \]

//subst/copy
//insert
//delete

(simplify by letting \(d(c,d)=0\) if \(c=d\), 1 else)

also let \(D(i,0)=i\) (for \(i \) inserts) and \(D(0,j)=j\)

Computing Levenshtein distance - 3

\[ D(i,j) = \min \begin{cases} 
D(i-1,j-1) + d(s_i,t_j), & \text{if } s_i = t_j \\
D(i-1,j) + 1, & \text{if } s_i \neq t_j \\
D(i,j-1) + 1 \\
\end{cases} \]

\[ = D(s,t) \]

Computing Levenshtein distance – 4

\[ D(i,j) = \min \begin{cases} 
D(i-1,j-1) + d(s_i,t_j), & \text{if } s_i = t_j \\
D(i-1,j) + G, & \text{if } s_i \neq t_j \\
D(i,j-1) + G \\
\end{cases} \]

\[ = D(s,t) \]

A trace indicates where the min value came from, and can be used to find edit operations and/or a best alignment (may be more than 1)

Needleman-Wunch distance

\[ D(i,j) = \min \begin{cases} 
D(i-1,j-1) + d(s_i,t_j), & \text{if } s_i = t_j \\
D(i-1,j) + G, & \text{if } s_i \neq t_j \\
D(i,j-1) + G \\
\end{cases} \]

\[ = D(s,t) \]

\[ G = \text{"gap cost"} \]

d\((c,d)\) is an arbitrary distance function on characters (e.g. related to typo frequencies, amino acid substitutibility, etc)

William Cohen
Wukkuan Cigeb

Smith-Waterman distance

• Instead of looking at each sequence in its entirety, this compares segments of all possible lengths and chooses whichever maximise the similarity measure. (Thus it is a generalization of "longest common subsequence.)

• For every cell the algorithm calculates all possible paths leading to it. These paths can be of any length and can contain insertions and deletions.
Smith-Waterman distance

\[
D(i,j) = \max \begin{cases} 
0 & \text{start over} \\
D(i-1,j-1) - d(si,tj) & \text{substitution/copy} \\
D(i-1,j) - G & \text{insert} \\
D(i,j-1) - G & \text{delete} 
\end{cases}
\]

\[G = 1\]
\[d(c,c) = -2\]
\[d(c,d) = +1\]

\[
\begin{array}{cccccc}
& O & C & H & E & N \\
M & 0 & 0 & 0 & 0 & 0 \\
C & +2 & 0 & 0 & 0 & 0 \\
C & +2 & 0 & 0 & 0 & 0 \\
O & 0 & +4 & +3 & 0 & 0 \\
H & 0 & +3 & +6 & +5 & +3 \\
N & 0 & +2 & +5 & +5 & +7 \\
\end{array}
\]

Smith-Waterman distance in Monge & Elkan’s WEBFIND (1996)

Used a standard version of Smith-Waterman with hand-tuned weights for inserts and character substitutions.

Split large text fields by separators like commas, etc, and found minimal cost over all possible pairings of the subfields (since S-W assigns a large cost to large transpositions)

Result competitive with plausible competitors.

Affine gap distances

- Smith-Waterman fails on some pairs that seem quite similar:

William W. Cohen

William W. ‘Don’t call me Dubya’ Cohen

Intuitively, single long insertions are “cheaper” than a lot of short insertions

Affine gap distances - 2

- Idea:
  - Current cost of a “gap” of \(n\) characters: \(nG\)
  - Make this cost: \(A + (n-1)B\), where \(A\) is cost of “opening” a gap, and \(B\) is cost of “continuing” a gap.
Affine gap distances - 3

\[ D(i,j) = \max \begin{cases} 
D(i-1,j-1) + d(si,tj) \\
D(i-1,j) - 1 \\
D(i,j-1) - 1 
\end{cases} \]

\[ IS(i,j) = \max \begin{cases} 
D(i-1,j) - A \\
IS(i-1,j) - B 
\end{cases} \]

\[ IT(i,j) = \max \begin{cases} 
D(i,j-1) - A \\
IT(i,j-1) - B 
\end{cases} \]

Best score in which \( si \) is aligned with a ‘gap’

Best score in which \( t_j \) is aligned with a ‘gap’

Affine gap distances as automata

Generative version of affine gap automata (Bilenko&Mooney, TechReport 02)

HMM emits pairs: \((c,d)\) in state \( M \), pairs \((-,-)\) in state \( D \), and pairs \((-,-)\) in state \( I \).

For each state there is a multinomial distribution on pairs.

The HMM can trained with EM from a sample of pairs of matched strings \((s,t)\)

E-step is forward-backward; M-step uses some ad hoc smoothing

Affine gap edit-distance learning: experiments results

<table>
<thead>
<tr>
<th>Distance metric</th>
<th>CORA title</th>
<th>RESTAURANT name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Levenshtein</td>
<td>0.870</td>
<td>0.843</td>
</tr>
<tr>
<td>Learned Levenshtein</td>
<td>0.902</td>
<td>0.896</td>
</tr>
<tr>
<td>Affine</td>
<td>0.917</td>
<td>0.883</td>
</tr>
<tr>
<td>Learned Affine</td>
<td>0.971</td>
<td>0.967</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distance metric</th>
<th>RESTAURANT address</th>
<th>MAILING address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Levenshtein</td>
<td>0.950</td>
<td>0.867</td>
</tr>
<tr>
<td>Learned Levenshtein</td>
<td>0.975</td>
<td>0.899</td>
</tr>
<tr>
<td>Affine</td>
<td>0.870</td>
<td>0.923</td>
</tr>
<tr>
<td>Learned Affine</td>
<td>0.929</td>
<td>0.959</td>
</tr>
</tbody>
</table>
Affine gap distances – experiments (from McCallum, Nigam, Ungar KDD2000)

- Goal is to match data like this:


Figure 2: Three sample citations to the same paper.

Affine gap distances – experiments

<table>
<thead>
<tr>
<th>TFIDF</th>
<th>Edit Distance</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cora</td>
<td>0.751</td>
<td>0.839</td>
</tr>
<tr>
<td></td>
<td>0.721</td>
<td></td>
</tr>
<tr>
<td>OrgName1</td>
<td>0.925</td>
<td>0.833</td>
</tr>
<tr>
<td></td>
<td>0.366</td>
<td>0.950</td>
</tr>
<tr>
<td>Orgname2</td>
<td>0.958</td>
<td>0.571</td>
</tr>
<tr>
<td></td>
<td>0.778</td>
<td>0.912</td>
</tr>
<tr>
<td>Restaurant</td>
<td>0.961</td>
<td>0.827</td>
</tr>
<tr>
<td></td>
<td>0.967</td>
<td>0.867</td>
</tr>
<tr>
<td>Parks</td>
<td>0.976</td>
<td>0.967</td>
</tr>
<tr>
<td></td>
<td>0.967</td>
<td>0.967</td>
</tr>
</tbody>
</table>

String distance metrics: outline

- Term-based (e.g. TF/IDF as in WHIRL)
  - Distance depends on set of words contained in both s and t.
- Edit-distance metrics
  - Distance is shortest sequence of edit commands that transform s to t.
- Pair HMM based metrics
  - Probabilistic extension of edit distance
- Other metrics

Jaro metric

- Jaro metric is (apparently) tuned for personal names:
  - Given (s,t) define c to be common in s,t if it sinc, |i-c, and |j-c|<min(|s|,|t|)/2.
  - Define c,d to be a transposition if c,d are common and c,d appear in different orders in s and t.
  - Jaro(s,t) = average of #common/s, #common/t, and 0.5#transpositions/#common
  - Variant: weight errors early in string more heavily
- Easy to compute – note edit distance is O(|s||t|)

NB. This is my interpretation of Winkler’s description
### Soundex metric
- Soundex is a coarse phonetic indexing scheme, widely used in genealogy.
- Every Soundex code consists of a letter and three numbers between 0 and 6, e.g. B-536 for “Bender”. The letter is always the first letter of the surname. The numbers hash together the rest of the name.
  - Vowels are generally ignored: e.g. Lee, Lu => L-000. Later consonants in a name are ignored.
  - Similar-sounding letters (e.g. B, P, F, V) are not differentiated, nor are doubled letters.
  - There are lots of Soundex variants.

### N-gram metric
- Idea: split every string \( s \) into a set of all character \( n \)-grams that appear in \( s \), for \( n \leq k \). Then, use term-based approaches.
  - e.g. "COHEN" => \{C,O,H,E,N,CO,OH,HE,EN,COH,OHE,HEN\}
  - For \( n=4 \) or \( 5 \), this is competitive on retrieval tasks. It doesn’t seem to be competitive with small values of \( n \) on matching tasks (but it’s useful as a fast approximate matching scheme)

### Main Points
**Co-reference**
- How to cast as classification [Cardie]
- Measures of string similarity [Cohen]
- Scaling up [McCallum et al]

**Relation extraction**
- With augmented grammar [Miller et al 2000]
- With joint inference [Roth]
- Semi-supervised [Brin]

### Reference Matching

### The Citation Clustering Data
- Over 1,000,000 citations
- About 100,000 unique papers
- About 100,000 unique vocabulary words
- Over 1 trillion distance calculations

### The Canopies Approach
- Two distance metrics: cheap & expensive
- First Pass
  - very inexpensive distance metric
  - create overlapping canopies
- Second Pass
  - expensive, accurate distance metric
  - canopies determine which distances calculated
Creating canopies with two thresholds
- Put all points in D
- Loop:
  - Pick a point X from D
  - Put points within $K_{loose}$ of X in canopy
  - Remove points within $K_{tight}$ of X from D

Using canopies with Greedy Agglomerative Clustering
- Calculate expensive distances between points in the same canopy
- All other distances default to infinity
- Sort finite distances and iteratively merge closest

Computational Savings
- inexpensive metric $\ll$ expensive metric
- # canopies per data point: $f$ (small, but $> 1$)
- number of canopies: $c$ (large)
- complexity reduction:
  $$O\left(\frac{f^2}{c}\right)$$

The Experimental Dataset
- All citations for authors:
  - Michael Kearns
  - Robert Schapire
  - Yoav Freund
- 1916 citations
- 121 unique papers
- Similar dataset used for parameter tuning
Inexpensive Distance Metric for Text

- Word-level matching (TFIDF)
- Inexpensive using an inverted index

Expensive Distance Metric for Text

- String edit distance
- Compute with Dynamic Programming
- Costs for character:
  - insertion
  - deletion
  - substitution
  - ...

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>e</th>
<th>c</th>
<th>a</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>0.0</td>
<td>0.7</td>
<td>1.4</td>
<td>2.1</td>
<td>2.8</td>
</tr>
<tr>
<td>c</td>
<td>1.4</td>
<td>0.7</td>
<td>1.0</td>
<td>0.7</td>
<td>1.4</td>
</tr>
<tr>
<td>o</td>
<td>0.0</td>
<td>0.7</td>
<td>0.7</td>
<td>1.1</td>
<td>1.4</td>
</tr>
<tr>
<td>t</td>
<td>2.8</td>
<td>1.4</td>
<td>2.1</td>
<td>1.8</td>
<td>2.4</td>
</tr>
<tr>
<td>t</td>
<td>3.5</td>
<td>1.8</td>
<td>2.4</td>
<td>2.1</td>
<td>2.8</td>
</tr>
</tbody>
</table>

do Fahlman vs Falman

Extracting Fields using HMMs


Author: Fahlman, S.E. and Lebiere, C.
Title: The Cascade Correlation Learning Architecture
Venue: NIPS
Year: 1990

Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canopies GAC</td>
<td>0.838</td>
<td>7.65</td>
</tr>
<tr>
<td>Complete GAC</td>
<td>0.835</td>
<td>134.09</td>
</tr>
<tr>
<td>Existing Cora</td>
<td>0.784</td>
<td>0.03</td>
</tr>
<tr>
<td>Author/Year</td>
<td>0.697</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Add precision, recall along side F1

Main Points

Co-reference
- How to cast as classification [Cardie]
- Measures of string similarity [Cohen]
- Scaling up [McCallum et al]

Relation extraction
- With augmented grammar [Miller et al 2000]
- With joint inference [Roth & Yih]
- Semi-supervised [Brin]

Information Extraction

Named Entity Recognition

INPUT: Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.

OUTPUT: Profits soared at Company Boeing Co., easily topping forecasts on Location Wall Street, as their CEO Person Alan Mulally announced first quarter results.

Relationships between Entities

INPUT: Boeing is located in Seattle. Alan Mulally is the CEO.

OUTPUT:

{Relationship = Company-Location} = Boeing Co.
{Relationship = Employer-Employee} = Boeing Co.
{Person = Alan Mulally} = Boeing Co.
**Relationship Extraction**

**[Miller et. al, 2000]**

**An example:**

Donald M. Goldstein, a historian at the University of Pittsburgh ...

- **Entity information to be extracted:**
  - Named entity boundaries:
    - Organizations, people, and locations
  - Person descriptors: “a historian at the University of Pittsburgh” refers to “Donald M. Goldstein”

- **Entity relationships to be extracted:**
  - Employer/employee relations
    - (e.g., Goldstein is employed at University of Pittsburgh)
  - Company/product relations
  - Organization/headquarters-location relation

---

**The Basic Approach**

- **Build a statistical parsing model which simultaneously recovers syntactic relation and the information extraction information**
  - **To do this:**
    - **Step 1:** annotate training sentences for entities, descriptors, coreference links, and relation links
    - **Step 2:** train a parser on the Penn treebank, and apply it to the new training sentences. Force the parser to produce parses that are consistent with the entity/descriptor etc. boundaries
    - **Step 3:** enhance the parse trees to include the information extraction information (we’ll come to this soon)
    - **Step 4:** re-train the parser on the new training data, and with the new annotations

---

**Extraction From Entire Documents**

Hi [PERSON Ted] and [PERSON Hill],

Just a reminder that the game move will need to be entered [TIME tonight]. We will need data on operations, raw materials ordering, and details of the bond to be sold.

[PERSON Hill]: I will be in the [LOCATION lobby] after the class at [TIME 9 pm]. How about we meet in the [LOCATION lobby] around that time (i.e when both our classes are over).

[PERSON Ted]: Let me know how you are going to provide the bond related input information. We can either meet in the [LOCATION lobby] around [TIME 5:30 pm] or you can e-mail me the info.

Thanks, [PERSON Ajay]

<table>
<thead>
<tr>
<th>TIME</th>
<th>9 pm, 18th September</th>
<th>5:30 pm, 18th September</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCATION</td>
<td>Lobby, Building NE43</td>
<td>Lobby, Building NE43</td>
</tr>
<tr>
<td>PERSON</td>
<td>David Hill, Ajay Sinclaire</td>
<td>Ted Jones, Ajay Sinclaire</td>
</tr>
<tr>
<td>TOPIC</td>
<td>data on operations...</td>
<td>bond related input information</td>
</tr>
</tbody>
</table>

---

**Relationship Extraction: Annotation**

**Another example:**

Nance, who is a paid consultant to ABC News, said ...

- **The following information was annotated:**
  - **Nance** as a person; **ABC News** as an organization; a paid consultant to ABC News as a descriptor
  - A coreference link between Nance and a paid consultant to ABC News
  - An employer-relation link from a paid consultant to ABC News to ABC News

Next question: how can we build a model which recovers this information?
Add semantic tags showing named entities
org = organization, per = person, org-r = organization “reportable” (complete),
per-r = person “reportable” (complete)

Add semantic tags showing descriptors
per-desc = person descriptor,
per-desc-r = person descriptor “reportable” (complete)

Add link showing employee-employer relation
emp-of = employee-of link, emp.ptr = employee-of pointer

Building a Parser
- We now have context-free rules where each non-terminal in
  the grammar has
  - A syntactic category
  - A semantic label
  - A head-word/head-tag

NP/per-desc-r(consultant/NN)  PP-lnk/emp-of(to/TO)

It’s possible to modify syntactic parsers to estimate rule
probabilities in this case
**Summary**

- **Goal**: build a parser that recovers syntactic structure, named entities, descriptors, and relations
- **Annotation**: mark entity boundaries, descriptor boundaries, links between entities and descriptors
- **Enriched parse trees**: given annotation, and a parse tree, form a new enriched parse tree
- **The statistical model**: non-terminals now include syntactic category, semantic label, head word, head tag. Rule probabilities are estimated using similar methods to syntactic parsers
- **Results**: precision = 81%, recall = 64% in recovering relations (employer/employee, company/product, company/headquarters-location)

**Main Points**

**Co-reference**
- How to cast as classification [Cardie]
- Measures of string similarity [Cohen]
- Scaling up [McCallum et al]

**Relation extraction**
- With augmented grammar [Miller et al 2000]
- **With joint inference** [Roth & Yih]
- Semi-supervised [Brin]
Partially Supervised Approaches to Relation Extraction

- Last lecture: introduced a partially supervised method for named entity classification
- Basic observation: “redundancy” in that either spelling or context of an entity is often sufficient to determine its type
- Lead to *cotraining* approaches, where two classifiers bootstrap each other from a small number of seed rules
- Can we apply these kind of methods to relation extraction?

---

**From [Brin, 1998]**

For data we used a repository of 24 million web pages totalling 147 gigabytes. This data is part of the Stanford WebBase and is used for the Google search engine [BP], and other research projects. As a part of the search engine, we have built an inverted index of the entire repository.

The repository spans many disks and several machines. It takes a considerable amount of time to make just one pass over the data even without doing any substantial processing. Therefore, in these [sic] we only made passes over subsets of the repository on any given iteration.

http://google.stanford.edu

---

**DIPRE [Brin, 1998]**

A pattern is a 5 tuple:

- *Order*: author preceding title, or visa versa
- *URL-prefix*: a prefix of the URL of the page of the pattern
- *prefix*: up to 10 characters preceding the author/title pair
- *middle*: the characters between the author and title
- *suffix*: up to 10 characters following the author/title pair

---

**From [Brin, 1998]**

authors/book-titles, data = web data, seeds are

<table>
<thead>
<tr>
<th>Author</th>
<th>Book Title</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 Gleik</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>

---

**DIPRE: Inducing Patterns from Data**

- Find all instances of seeds on web pages.
  Basic question: how do we induce patterns from these examples?
Answer = Following procedure:
1. Group all occurrences together which have the same values for order, middle
2. For any group: Set url-prefix to be longest common prefix of the group’s URLs, prefix to be the longest common prefix of the group, suffix to be the longest common suffix
3. For each group’s pattern, calculate its specificity as
   \[ spe(x) = \frac{|\text{prefix}|}{|\text{suffix}|} \]
   where \( \gamma \) is the number of examples in the group, \( |x| \) is the length of \( x \) in characters
4. If specificity exceeds some threshold, include the pattern
5. Else If all patterns occur on same webpage, reject the pattern
6. Else create new sub-groups grouped by characters in the urls which is one past url-prefix, and repeat the procedure in step 2 for these new sub-groups.

The Overall Algorithm
1. Use the seed examples to label some data
2. Induce patterns from the labeled examples, using method described on the previous slide
3. Apply the patterns to data, to get a new set of author/title pairs
4. Return to step 2, and iterate

DIPRE: Inducing Patterns from Data
The patterns found in the first iteration:
- The 5 seeds produced 199 labeled instances, giving the 3 patterns above
- Applying the three patterns gave 4047 new book instances
- Searching 5 million web pages gave 3972 occurrences of these books
- This gave 105 patterns, 24 applied to more than one URL.
- Applied to 2 million URLs produced 9369 unique (author,title) pairs
- Manual intervention: removed 242 “bogus” items where the author was “Conclusion”
- Final iteration: ran over 156,000 documents which contained the word “books”; induced 346 patterns, 15,257 (author,title) pairs

(1) Association using Parse Tree
Simultaneously POS tag, parse, extract & associate! [Miller at al 2000]