Relation Extraction
from Wikipedia Text

Wikipedia Mining Seminar
Universität des Saarlandes

Miriam Käshammer

January 18, 2010
Outline

Introduction

PORE: Positive-Only Relation Extraction from Wikipedia Text

Unsupervised Relation Extraction from Wikipedia
Relation Extraction

Definition
Automated or human-assisted acquisition of relations between concepts from textual or other data.

Subtask of Information Extraction
- Used for Database/Ontology population, Semantic Web annotations
- Traditional supervised machine learning approaches: annotated training data → substantial human effort
- Need for RE algorithms that operate as unsupervised as possible
Relation Extraction

**PORE: Positive-Only Relation Extraction from Wikipedia Text**
G. Wang, Y. Yu and H. Zhu, ISWC, 2007

**Unsupervised Relation Extraction by Mining Wikipedia Texts Using Information from the Web**
Outline

Introduction

PORE: Positive-Only Relation Extraction from Wikipedia Text

Unsupervised Relation Extraction from Wikipedia
PORE

- Use the structure of Wikipedia articles to semi-automatically extract semantic relations from free Wikipedia text
- Core algorithm: B-POL (Bootstrapping positive-only learning)
PORE

- Use the structure of Wikipedia articles to semi-automatically extract semantic relations from free Wikipedia text
- Core algorithm: B-POL (Bootstrapping positive-only learning)

Steps

1. Extract **entity features** from semi-structured data of Wikipedia
2. Extract **context features** from the co-occurrence of two entities in one sentence in the Wikipedia text
3. For each relation, **filter** out irrelevant pairs
4. Conduct relation **classification** on the filtered set of pairs using B-POL
Entity Feature Extraction

Entity features describe Wikipedia entities (entries).

- **Definition features**: the head word of the first base noun phrase following a *be*-verb
  → film, comedy_film
Entity Feature Extraction

Entity features describe Wikipedia entities (entries).

- **Definition features**: the head word of the first base noun phrase following a *be*-verb
  → film, comedy_film

- **Category features**: the head word of the first base noun phrase in each category phrase
  → film, comedy_film, ...
Annie Hall

From Wikipedia, the free encyclopedia

Annie Hall is a 1977 American romantic comedy film directed by Woody Allen from a script co-written with Marshall brickman. It won numerous awards at the time of its release, including four Academy Awards, including Best Original Screenplay and Best Director. It was also nominated for Best Actress (Diane Keaton) and Best Supporting Actor (Tony Roberts). The film was a critical and commercial success, and is considered one of Woody Allen's most important works.
Entity Feature Extraction

Entity features describe Wikipedia entities (entries).

- **Definition features**: the head word of the first base noun phrase following a *be*-verb
  → film, comedy_film

- **Category features**: the head word of the first base noun phrase in each category phrase
  → film, comedy_film, ...

- **Infobox features**: predicate names from the infoboxes (white spaces replaced by underscores)
  → directed_by, produced_by, ...
Anniversary Hall
From Wikipedia, the free encyclopedia

*Anniversary Hall* is a 1977 American romantic comedy film directed by Woody Allen from a script co-written by Woody Allen. The film won numerous awards at the time of its release, including four Academy Awards, including Best Picture, Best Director, Best Actor for Woody Allen, and Best Actress for Diane Keaton. The film starred Diane Keaton, Woody Allen, and Marshall Thompson.

Directed by Woody Allen
Produced by Charles H. Joffe

Categories: English-language films | 1977 films | 1970s romantic comedy films | American films | Actress Academy Award winning performance | Films set in Brooklyn | Films set in New York City | director won the Best Director Academy Award | Films whose writer won the Best Original Screenplay
Context Feature Extraction

Context features describe co-occurrence contexts of pairs of Wikipedia entities in a sentence.

In the film "Heavenly Creatures", directed by Peter Jackson, Juliet Hulme had TB, and her fear of being sent ...

on the page Tuberculosis in popular culture

- Six entity pairs (SUBJ, OBJ): e.g.
  (Heavenly Creatures, Peter Jackson)
- For each pair, tokens in the left context, in the right context and tokens between the two entities are encoded as the context features
Filtering

- Very large number of entity pairs
- Use the entity features for filtering the pairs
- Features scoring function:

\[
\text{score}(f) = |P_f| \times \log\left(\frac{|C|}{|C_f|}\right)
\]

- Score the entity features at each argument position (subject or object) and select the top \(k\) features (\(k = 15\)).
- Keep pairs in which entity features of SUBJ intersect with the \textit{Salient Subject Features} and entity features of OBJ intersect with the \textit{Salient Object Features}. \(\Rightarrow C'\)
- \(U = C' - P\)
Positive-only binary classification

Given:

- a collection $C$ of context feature vectors of entity pairs
- a relation type $R$
- a set of positive training data $P \subset C$ (“seeds”)

Task: classify the unlabeled set $U = C - P$ into entity pairs which are of type $R$ (positive set) and entity pairs which are not of type $R$ (negative set)
B-POL

B-POL builds on top of the POL (positive-only learning) approach. POL initially identifies very strong negative examples from the unlabeled data and then iteratively classifies more negative data until no such data can be found.

POL$(P, U)$

1. Use a weak classifier (Rocchio) to classify using $P$ and $U$.
   $P_0 \leftarrow$ the data in $U$ classified as positive; $N_0 \leftarrow U - P_0$
2. $N \leftarrow \emptyset; i \leftarrow 0$
3. Do loop until $N_i = \emptyset$
   $N \leftarrow N \cup N_i$
   Use $v$-SVM to classify $P_i$ with $P$ and $N \Rightarrow N_{i+1}, P_{i+1}$
   $i \leftarrow i + 1$
4. $P_u \leftarrow P_i$; return $P_u$
B-POL cont.

When $P$ is very small $\Rightarrow$ low recall using POL

Extension of POL:
Add the newly generated positive data $P_u$ identified by POL to the set of positive training samples and invoke POL again to generate more positive data. (bootstrapping)

B-POL($P,U$)

1. $P_u \leftarrow \emptyset$; $i \leftarrow 0$

2. Do loop
   
   - $i \leftarrow i + 1$
   - $P_u^{(i)} \leftarrow$ positive examples returned from POL($P \cup P_u, U$)
   - $P_u \leftarrow P_u \cup P_u^{(i)}$; $U \leftarrow U - P_u^{(i)}$
   
   Repeat until $P_u^{(i)} = \emptyset$

3. Return $P_u$
Evaluation

- Definitions of relations and the corresponding training instances are taken from the infoboxes of Wikipedia.
- 10,000 randomly selected Wikipedia pages (no disambiguation or list-of pages) ⇒ ~130,000 pairs of entities
- Construction of a gold standard set for each relation tested

### Table 1. Information about the four relations.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Source</th>
<th>#GS</th>
<th>#U</th>
<th>#(GS ∩ U)</th>
</tr>
</thead>
<tbody>
<tr>
<td>album-artist</td>
<td>album_infobox#artist</td>
<td>274</td>
<td>392</td>
<td>260</td>
</tr>
<tr>
<td>film-director</td>
<td>infobox_movie#director</td>
<td>121</td>
<td>286</td>
<td>115</td>
</tr>
<tr>
<td>university-city</td>
<td>infobox_university#city</td>
<td>74</td>
<td>208</td>
<td>71</td>
</tr>
<tr>
<td>band-member</td>
<td>infobox_band#current_members</td>
<td>117</td>
<td>477</td>
<td>103</td>
</tr>
</tbody>
</table>

Recall $\frac{#(GS \cap U)}{#GS}$ at this stage is relatively high.
Evaluation cont.

Extraction performance of B-POL

<table>
<thead>
<tr>
<th>#P</th>
<th>method</th>
<th>album-artist P/R/F1</th>
<th>film-director P/R/F1</th>
<th>university-city P/R/F1</th>
<th>band-member P/R/F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>T-POL’</td>
<td>96.7/36.5/47.8</td>
<td>82.8/50.6/60.6</td>
<td>65.4/74.4/68.6</td>
<td>70.2/25.0/35.7</td>
</tr>
<tr>
<td></td>
<td>T-POL</td>
<td>89.6/49.8/59.2</td>
<td>82.2/58.2/66.4</td>
<td>62.0/76.8/68.1</td>
<td>67.6/25.0/34.8</td>
</tr>
<tr>
<td></td>
<td>B-POL</td>
<td>86.6/77.5/79.9</td>
<td>69.4/81.2/73.2</td>
<td>47.2/84.8/58.5</td>
<td>46.8/57.6/47.1</td>
</tr>
<tr>
<td></td>
<td>M-SVM</td>
<td>93.6/40.4/54.5</td>
<td>71.2/32.8/41.4</td>
<td>17.4/36.9/19.5</td>
<td>35.4/29.7/27.5</td>
</tr>
<tr>
<td>30</td>
<td>T-POL’</td>
<td>97.4/45.8/58.8</td>
<td>85.5/51.1/62.2</td>
<td>75.1/67.7/70.5</td>
<td>74.3/24.5/35.9</td>
</tr>
<tr>
<td></td>
<td>T-POL</td>
<td>93.2/56.7/68.2</td>
<td>83.7/51.0/61.8</td>
<td>70.7/72.6/70.6</td>
<td>67.6/22.0/32.4</td>
</tr>
<tr>
<td></td>
<td>B-POL</td>
<td>90.6/70.2/76.5</td>
<td>73.4/69.6/68.6</td>
<td>62.7/79.0/68.5</td>
<td>58.5/46.6/49.3</td>
</tr>
<tr>
<td></td>
<td>M-SVM</td>
<td>93.4/46.2/58.0</td>
<td>72.1/37.9/44.8</td>
<td>20.9/33.7/21.9</td>
<td>36.1/32.5/30.0</td>
</tr>
<tr>
<td>20</td>
<td>T-POL’</td>
<td>97.1/34.6/48.0</td>
<td>84.6/37.7/49.9</td>
<td>80.3/63.6/70.5</td>
<td>77.7/21.7/33.5</td>
</tr>
<tr>
<td></td>
<td>T-POL</td>
<td>93.5/52.8/63.7</td>
<td>81.3/47.0/56.5</td>
<td>79.8/64.0/70.2</td>
<td>72.3/21.0/31.5</td>
</tr>
<tr>
<td></td>
<td>B-POL</td>
<td>90.0/69.2/76.4</td>
<td>74.7/64.1/66.6</td>
<td>75.3/70.1/71.6</td>
<td>67.9/32.3/41.9</td>
</tr>
<tr>
<td></td>
<td>M-SVM</td>
<td>93.8/42.4/55.9</td>
<td>73.1/40.5/46.9</td>
<td>27.0/31.6/26.0</td>
<td>39.4/39.2/29.8</td>
</tr>
<tr>
<td>10</td>
<td>T-POL’</td>
<td>99.1/35.3/50.7</td>
<td>89.1/32.1/45.7</td>
<td>82.5/57.7/66.7</td>
<td>81.4/12.5/21.2</td>
</tr>
<tr>
<td></td>
<td>T-POL</td>
<td>96.7/40.5/53.8</td>
<td>86.2/30.5/42.5</td>
<td>84.1/54.1/64.8</td>
<td>76.7/15.2/24.6</td>
</tr>
<tr>
<td></td>
<td>B-POL</td>
<td>95.0/48.6/61.3</td>
<td>83.2/41.3/51.0</td>
<td>82.7/58.1/67.5</td>
<td>74.0/19.9/30.1</td>
</tr>
<tr>
<td></td>
<td>M-SVM</td>
<td>93.4/46.3/58.9</td>
<td>78.3/31.4/42.7</td>
<td>32.1/28.1/29.1</td>
<td>40.6/32.8/26.4</td>
</tr>
</tbody>
</table>
Outline

Introduction

PORE: Positive-Only Relation Extraction from Wikipedia Text

Unsupervised Relation Extraction from Wikipedia
An integrated Approach

Dependency patterns from dependency analysis
- Linguistic technologies to abstract away from different surface realizations of semantic relations
- Expected to be more accurate $\Rightarrow$ good precision
- Dependency parsing requires text of good quality
- Wikipedia as a local corpus

Surface patterns generated from redundant Web information
- Contribute greatly to the coverage
- The Web as a global corpus

$\Rightarrow$ Bridge the gap separating “deep” linguistic technology and redundant Web information for IE tasks
The framework

Assumption: It is likely that a salient semantic relation $r$ exists between a page $p$ and a related page $p'$ which is linked on page $p$.

$\Rightarrow$ Relation Extraction between the entitled concept (ec) and a related concept (rc), which appear as links in the text of the article.
The framework

Assumption: It is likely that a salient semantic relation $r$ exists between a page $p$ and a related page $p'$ which is linked on page $p$.

$\Rightarrow$ Relation Extraction between the entitled concept (ec) and a related concept (rc), which appear as links in the text of the article.

Given: a set of Wikipedia articles

Output: a list of concept pairs for each article with a relation label assigned to each concept pair
Relation Extraction from Wikipedia Text

Unsupervised Relation Extraction from Wikipedia

Preprocessor

- Split the text of an article into sentences.
- Select sentences containing one reference of the entitled concept and one of a linked concept
  ⇒ a set of concept pairs, each associated with a sentence
Dependency Pattern Extractor

1. Parse the sentence for a concept pair; induce the shortest dependency path with the entitled concept and the related concept.

2. Generate sub-paths of the shortest path as dependency patterns (frequent tree-mining algorithm, Zaki, 2002).
Web Context Collector

- Query a concept pair using a search engine (Google).
- Extract two kinds of relational information from the retrieved snippets:
  1. Ranked relational terms (as keywords)
  2. Surface patterns
Web Context Collector cont.

Relational term ranking

- Relational terms are defined as verbs and nouns in the snippet.
- For each concept pair, collect a list of relational terms.
- Rank the relational terms of all concept pairs (entropy-based algorithm, Chen et al., 2005) \( \Rightarrow T_{all} \)
- For each concept pair, sort the list \( T_{cp} \) according to the terms’ order in \( T_{all} \)
- For each concept pair, select the top term as a keyword \( t_{cp} \)
Web Context Collector cont.

Surface pattern generation

- Consider a snippet sentence
- Content Words (CW): entitled concept, related concept and the keyword $t_{cp}$
- Functional Words (FW): verbs, nouns, prepositions and coordinating conjunctions
- General form of surface patterns:
  \[CW_1 \text{ Infix } CW_2 \text{ Infix } CW_3\]
  with Infix = FW*
  e.g. ec assign rc as ceo
  ceo of ec rc
Clustering

k-Means Algorithm

Initial Centroid Selection based on the keyword $t_{cp}$ of each concept pair:

- Group all concept pairs by their keyword $t_{cp}$
- $G = \{ G_1, G_2, ... G_n \}$ with each $G_i = \{ cp_{i1}, cp_{i2}, ... \}$ being a group of concept pairs that share the same keyword
- Rank the groups by their number of concept pairs and choose the top $k$ groups (stability-based criteria, Chen et al., 2005)
- For each group $G_i$, select a centroid $c_i$:

$$c_i = \arg \max_{cp \in G_i} |\{ cp_{ij} | (dis_1(cp_{ij}, cp) + \\
\lambda \times dis_2(cp_{ij}, cp)) \leq D_z, 1 \leq j \leq |G_i|)\}$$
Clustering cont.

Dependency pattern distance

\[
dis_1(cp_i, cp_j) = 1 - \frac{|DP_i \cap DP_j|}{\sqrt{|DP_i| * |DP_j|}}
\]

\(DP_x\): dependency pattern set of the concept pair \(cp_x\)
Clustering cont.

Surface pattern distance

\( dis_2(cp_i, cp_j) \)

Input: \( SP_1 = \{ sp_{11}, ..., sp_{1m} \}, \ SP_2 = \{ sp_{21}, ..., sp_{2n} \} \)

1. Define a \( m \times n \) distance matrix \( A \):

\[
\{ A_{ij} = \frac{LD(sp_{1i}, sp_{2j})}{\max(|sp_{1i}|, |sp_{2j}|)} ; 1 \leq i \leq m; 1 \leq j \leq n \}
\]

2. \( dis \gets 0 \)

3. for \( \min(m, n) \) times do

\((x, y) \gets \arg \min_{0 < i < m; 0 < j < n} A_{ij}\)

\( dis \gets dis + A_{xy} / \min(m, n) \)

\( A_{x*} \gets 1; A_{*y} \gets 1 \)

4. return \( dis \)
Depend Clustering

- Given the initial $k$ centroids, merge the concept pairs into $k$ clusters according to their dependency patterns.
- Each concept pair $cp_i$ has a set of dependency patterns $DP_i$.
- Distance between two concept pairs $cp_i$ and $cp_j$: $dis_1(cp_i, cp_j)$

Steps:

1. Assign each concept pair to the cluster with the closest centroid if the distance is smaller than $D_i$.
2. Then recompute each centroid based on the current members of the cluster.
3. Repeat steps 1. and 2. until the centroids do not change anymore.
CEO-relation

Text 1: the CEO of EC is RC  
Text 2: RC is the CEO of EC

Text 3: RC was hired as EC's CEO  
Text 4: EC assign RC as CEO
Surface Clustering

- Merge more concept pairs into the existing clusters using the surface patterns to improve the coverage.
- Each concept pair \( cp_i \) has a set of surface patterns \( SP_i \).
- Distance between two concept pairs \( cp_i \) and \( cp_j \): \( dis_2(cp_i, cp_j) \)

Steps:

1. Assign each concept pair (which has not yet been assigned to a cluster by depend clustering) to the cluster with the closest centroid if the distance is smaller than \( D_g \).
2. Then recompute each centroid based on the current members of the cluster.
3. Repeat steps 1. and 2. until the centroids do not change anymore.
Clustering cont.

Result

1. \( k \) clusters of concept pairs
2. Use the centroid pair to assign a label to the corresponding relation
Evaluation

Comparison with another method

- 526 articles from the Wikipedia category “American chief executives”
- 7310 concept pairs
- $k = 18$
- 15 clearly identifiable relations
- $\#\text{Ins.}: \text{number of concept pairs in the cluster}$
- $\text{pre}: \text{precision of the cluster}$
- $\text{coverage} = \frac{\#\text{correct Ins.}}{\#\text{all concept pairs}}$

<table>
<thead>
<tr>
<th>Relation (sample)</th>
<th># Ins.</th>
<th>pre</th>
<th># Ins.</th>
<th>pre</th>
</tr>
</thead>
<tbody>
<tr>
<td>chairman (x be chairman of y)</td>
<td>434</td>
<td>63.52</td>
<td>547</td>
<td>68.37</td>
</tr>
<tr>
<td>ceo (x be ceo of y)</td>
<td>396</td>
<td>73.74</td>
<td>423</td>
<td>77.54</td>
</tr>
<tr>
<td>bear (x be bear in y)</td>
<td>138</td>
<td>83.33</td>
<td>276</td>
<td>86.96</td>
</tr>
<tr>
<td>attend (x attend y)</td>
<td>225</td>
<td>67.11</td>
<td>313</td>
<td>70.28</td>
</tr>
<tr>
<td>member (x be member of y)</td>
<td>14</td>
<td>85.71</td>
<td>175</td>
<td>91.43</td>
</tr>
<tr>
<td>receive (x receive y)</td>
<td>97</td>
<td>67.97</td>
<td>117</td>
<td>73.53</td>
</tr>
<tr>
<td>graduate (x graduate from y)</td>
<td>18</td>
<td>83.33</td>
<td>92</td>
<td>88.04</td>
</tr>
<tr>
<td>degree (x obtain y degree)</td>
<td>5</td>
<td>80.00</td>
<td>78</td>
<td>82.05</td>
</tr>
<tr>
<td>marry (x marry y)</td>
<td>55</td>
<td>41.67</td>
<td>74</td>
<td>61.25</td>
</tr>
<tr>
<td>earn (x earn y)</td>
<td>23</td>
<td>86.96</td>
<td>51</td>
<td>88.24</td>
</tr>
<tr>
<td>award (x won y award)</td>
<td>23</td>
<td>43.47</td>
<td>46</td>
<td>84.78</td>
</tr>
</tbody>
</table>
Evaluation

Contribution of the different patterns

<table>
<thead>
<tr>
<th>Pattern type</th>
<th>#Instance</th>
<th>Precision</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>dependency</td>
<td>1127</td>
<td>84.29</td>
<td>13.00%</td>
</tr>
<tr>
<td>surface</td>
<td>1510</td>
<td>68.27</td>
<td>14.10%</td>
</tr>
<tr>
<td>Combined</td>
<td>2314</td>
<td>75.63</td>
<td>23.94%</td>
</tr>
</tbody>
</table>
Conclusion

Two Approaches to Relation Extraction

1. **Semi-supervised**: seed-based bootstrapping approach
   - Uses the structure of Wikipedia to extract entity features and entity pairs with context features
   - Positive-only learning
   - Finds instances of one specified relation in Wikipedia

2. **Unsupervised**: clustering approach
   - Uses the link structure of Wikipedia to extract entity pairs in a sentence
   - Combines linguistic analysis with redundancy information from the Web
   - k-means clustering
   - Finds the major relations and the corresponding instances in the given corpus (e.g. Wikipedia articles)
References

