Ontology Extension for Reading the Web

Mohamed Thahir
Outline

- Traditional and Open Relation Extraction
- Read the Web Relation Extraction
- Experimental Results
- Coupled learning of Predicates
- Challenges and ongoing work
A relation is instantiated with a set of manually provided positive and negative examples

city “capital of” Country

Positive Seeds:
{“washington d.c , USA”;”New Delhi , India”..}  
Negative Seeds:
{“USA , Canada”;”London , India”....}
Open Relation Extraction

- Proposed by Banko et. al 2007
- A classifier is built which given the entities and their context, identifies if there a valid relation
- Performs “Unlexicalized” extraction
- E1 Context E2

Some Features:
- Part of Speech (POS) tags in ‘Context’
- Number of tokens and stop words in ‘Context’
- POS tag to left of E1 and to right of E2
Comparison

- Banko et.al 2008 – “TradeOff between Open and Traditional RE”
- Comparison between Traditional (R1–CRF) and Open RE (O–CRF)

Averaged results for 4 common relations

<table>
<thead>
<tr>
<th>O–CRF (P)</th>
<th>O–CRF (R)</th>
<th>R1–CRF (P)</th>
<th>R1–CRF (R)</th>
<th>Train Ex</th>
</tr>
</thead>
<tbody>
<tr>
<td>75.0</td>
<td>18.4</td>
<td>73.9</td>
<td>58.4</td>
<td>5930</td>
</tr>
</tbody>
</table>
Open RE vs. Traditional RE

Pros:

✓ Open RE can scale to the size of the web (hundreds of thousands of relation predicates)
✓ Does not require human input unlike traditional RE
✓ Pretty reasonable level of precision
Open RE vs. Traditional RE

Cons:
- Open RE has much lower recall
- 30% of extracted tuples are not well-formed (does not imply a relation)
  - (demands, securing of, border)
  - (29, dropped, instruments)
- 87% of well-formed tuples are abstract/underspecified
  - (Einstein, derived, theory) – abstract tuple
  - (Washington dc, capital of, USA) – concrete tuple
RTW Relation Extraction

Combine beneficial aspects of Traditional and Open Relation Extraction with RTW

- Find new Relation Predicates automatically
- Also extract positive seed examples and negative seed examples automatically
- Leverage the constrained & coupled learning offered by RTW
- Improve learning of the existing category and relation predicates as well
Learning new Relations

Actor
De Caprio
Johnny Depp
Arnold

Movie
Titanic
Pirates of Carr..
Terminator

Actor “stars in” Movie
Actor “starring in” Movie
Movie “movie” Actor
Actor “praised“ Movie
Actor “sang in” Movie
Patterns which are rare are removed
Patterns which have either a very small Domain or very small Range are removed
  ◦ Removes many irrelevant patterns (caused due to ambiguity)
  ◦ Removes very specific patterns

NP “was engulfed in” flames
Vehicle
Sportsteam
## Learning new Relations

<table>
<thead>
<tr>
<th></th>
<th>starring</th>
<th>stars in</th>
<th>movie</th>
<th>sang in</th>
<th>praised</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeCaprio:Titanic</td>
<td>10</td>
<td>22</td>
<td>15</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Depp:Pirates of..</td>
<td>22</td>
<td>10</td>
<td>19</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Arnold:Terminat.</td>
<td>12</td>
<td>15</td>
<td>20</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Arnold:Titanic</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>X:Y</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>XX:YY</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
## Learning new Relations

<table>
<thead>
<tr>
<th></th>
<th>starring</th>
<th>stars in</th>
<th>movie</th>
<th>sang in</th>
<th>praised</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeCaprio: Titanic</td>
<td>10</td>
<td>22</td>
<td>15</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Depp: Pirates of..</td>
<td>22</td>
<td>10</td>
<td>19</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Arnold: Terminator</td>
<td>12</td>
<td>15</td>
<td>20</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Arnold: Titanic</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>X:Y</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>XX:YY</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- TF/IDF Normalization
- K–means clustering
Learning new Relations

- Each cluster with sufficient instances is taken as a new relation predicate ($NR$)
- Instances near the centroid of the cluster are taken as seed instances
- Relations whose domain and range are mutually exclusive to the domain and range of $NR$ are considered as mutually exclusive for $NR$
- $NR$ is introduced to RTW system as a new predicate
RTW Category Instance Promotion

- *Movie category* predicate classifier

- Titanic
  - Promoted
  - Co-occurrence with positive patterns

- Terminator
  - Not Promoted
  - Co-occurrence with negative patterns
RTW Relation Instance Promotion

- **Actor–Movie relation** predicate classifier

  - Arnold : Terminator
  - Terminator

  - Promoted
  - Promoted

- New Relation helps learning new Category instances
Experimental Results

- Improved learning for existing category predicates
- Validation without running the RTW
- **Actor : Movie** predicate and its high confidence relation pattern set $R$
- Obtained all instances of “NP1 Context NP2”
  Where,
  - Context is in $R$
  - Either NP1 or NP2 is a promoted Actor instance
  - List the other NP that is not the Actor
Experimental Results

- 200+ new movie instances
- Constrained by the number of promoted Actor instances (~800 in CBL)
- Future iterations should cause further increase in Actor and Movie instances.
- > 80% precision
  - Negatives: comedy film
- RTW system category predicate classifiers would ideally not promote these negatives
RTW Relation Instance Promotion

- **Actor-Movie relation** predicate classifier

- Promoted only when category classifier is reasonably confident about the instance
Experimental Results

Repeated same experiment for Food–Food relation predicates

Two relations were extracted

<table>
<thead>
<tr>
<th>Relation</th>
<th>Patterns</th>
<th>Instances</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>“contain”, “is rich in”, “are rich in”</td>
<td>&gt;700</td>
<td>~60%</td>
</tr>
<tr>
<td>typeOf</td>
<td>“Such as”, “and other”, “including”</td>
<td>&gt;3000</td>
<td>~70%</td>
</tr>
</tbody>
</table>

Negatives: apple “contains” few calories
Learning more Relation Instances

- Learning of Horn Clause rules
  - `foodTreatsDisease(food,disease)` – existing predicate
  - `isTypeOf(food1,food2)` – learnt predicate
  - `isTypeOf(food1,food2) & foodTreatsDisease(food2,disease)`
    \[\Rightarrow \text{foodTreatsDisease(food1,disease)}\]

- Relation instances could be learnt even without direct contextual patterns connecting them (not possible in Open RE)
We saw that new relation predicates leads to learning more category & relation instances.
Learning more category & relation instances would also lead to learning new predicates.

- Actor
  - Tom Hanks
  - Arnold
  - Depp
  - ...

- Award
  - Oscar
  - Golden Globe
  - ...
  - ...

Coupled Learning of Predicates
Coupled Learning of Predicates

Actor
Tom Hanks
Arnold
Depp

Award
Oscar
Golden Globe
Many invalid relations are retrieved
Un-lexicalized approaches to tackle them
Banko & Etzioni 2008, suggest that 95% of relation patterns are classified into 8 categories

<table>
<thead>
<tr>
<th>Rel. Frequency</th>
<th>Category</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>37.8</td>
<td>E1 Verb E2</td>
<td>X established Y</td>
</tr>
<tr>
<td>22.8</td>
<td>E1 Noun+Prep E2</td>
<td>X settlement with Y</td>
</tr>
<tr>
<td>16.0</td>
<td>E1 Verb+Prep E2</td>
<td>X moved to Y</td>
</tr>
<tr>
<td>9.4</td>
<td>E1 Infinitive E2</td>
<td>X plans to acquire Y</td>
</tr>
<tr>
<td>5.2</td>
<td>E1 Modifier E2</td>
<td>X is Y winner</td>
</tr>
</tbody>
</table>
Challenges & Ongoing work

- Build a model which would estimate the validity of an extracted relation predicate

- Possible Features
  - Un–lexicalized features
  - One–One relations are mostly valid
  - Relations with Hearst’s patterns (isA /part of relation – “such as”) have high chance of being valid. (Hearst 1992)
Invalid Relations and causes

- Error in the promoted instances
  - CBL promotes Months of the year as countries
  - Organization ‘meeting in’ Country
    - US Senate ‘meeting in’ November
  - Cluster all country instances using the category patterns. Months might form a unique sub-cluster.
  - If the Organization instances link only to a particular sub-cluster then it indicates a weak relation
  - Above metric could be used as another feature
Invalid Relations and causes

- Ambiguity
  - Animal names match with sports team names
  - Animal ‘won’ trophy
    - Compare with other predicates which are mutex to it (Sportsteam *won* Trophy) and check if there have exactly matching patterns.
    - If the ‘animal’ instances associated with the *animal ‘won’ trophy* relation also have evidence that it is a ‘Sportsteam’ then this is a feature indicating the weakness of Animal ‘*won’ trophy’ relation
Challenges & Ongoing work

Invalid Relations and causes

- Underspecified Relations
  - These relations require more entities to be useful
  - `SportsTeam ‘defeated ‘ SportsTeam`
  - X defeated Y, Y defeated X etc.
  - There should be temporal and location information for this relation to make sense