Mapping
Open Information Extraction
to a Customer Ontology

Turing Center
University of Washington

Stephen Soderland, Mausam, Oren Etzioni
and the TextRunner team:
Janara Christensen, Tony Fader, Bo Qin, Alan Ritter,
Brendan Roof, Will Xu
Challenge of “Machine Reading”

Reading system is domain-independent
  – General-purpose knowledge of English

Rapid training for a domain:
  – Given an ontology
  – Minimal training for each relation
  – Minimal knowledge engineering

Reading task is domain-specific

Ontology for the domain:
  – Domain-specific classes
  – Domain specific relations
Examples of Customer Ontologies

DARPA Message Understanding Conferences (1990’s)
- Latin American Terrorism: “bank” = PhysicalTarget
- Microchip Manufacture: “cvd” = ManufacturingProcess
- Joint Business Ventures: “bank” = JVParent
- Management Sucession: “bank” = Corporation

DARPA Machine Reading Project (2009 – 2014)
- NFL Scoring: “a 14-13 victory” = NFLGame
- Intelligence Community: “blast” = BombingEvent
Examples of Customer Ontologies

DARPA Message Understanding Conferences (1990’s)
- Latin American Terrorism
- Microchip Manufacture
- Joint Business Ventures
- Management Succession

Training:
- 1,000 annotated documents
- 50% irrelevant documents
- 50 page manual

DARPA Machine Reading Project (2009 – 2014)
- NFL Scoring
- Intelligence Community

Training:
- 10+ examples per relation
- no negative training
Outline of Talk

• The Machine Reading task
• UW reading system
  – Open Information Extraction
  – Map Open IE tuples to domain relations
• Steps in mapping to a domain
  – Recognize domain classes
  – Learn relation mapping rules
  – Learn semantic constraints
• Results for NFL and IC domains
• Future work, conclusions
System Architecture

Documents → TextRunner

Open IE

Class Recognizers

Tuples

Relation Mapping Rules

(arg1, pred, arg2)

Relation Mapping

Domain Relations

Domain independent

Domain specific

Soderland, Roof, Qin, Xu, Mausam, and Etzioni, Adapting Open Information Extraction to Domain-Specific Relations, *AI Magazine*, Autumn 2010
TextRunner Open IE

- Extracts relational tuples (arg1, pred, arg2)
- Operates on Web-scale text corpora
- No pre-specified relations
- No human effort
- But, args and pred are uninterpreted phrases

“Blake threw three touchdown passes as New Orleans cruised to a 31-15 victory over the 49ers.”

arg1 pred arg2
( Blake , threw , three touchdown passes )
( New Orleans , cruised to , a 31-15 victory )
( a 31-15 victory , over , the 49ers )

Banko and Etzioni, The Tradeoffs between Open and Traditional Relation Extraction, ACL, 2008
“Blake threw three touchdown passes as New Orleans cruised to a 31-15 victory over the 49ers.”

FinalScore
- NFLGame: a 31-15 victory
- NFLTeam: New Orleans
- PointsScored: 31

FinalScore
- NFLGame: a 31-15 victory
- NFLTeam: 49ers
- PointsScored: 15

GameWinner
- NFLGame: a 31-15 victory
- NFLTeam: New Orleans

GameLoser
- NFLGame: a 31-15 victory
- NFLTeam: 49ers

Touchdowns
- NFLGame: a 31-15 victory
- NFLTeam: New Orleans
- Count: three touchdown passes (3)
“Tanzanian Al-Qaed blew up a US embassy in Tanzania in 1998, which killed 11 people.”

BombingAgent
- HumanAgent: Tanzanian Al-Qaed

BombingTarget
- Target: a US embassy

BombingLocation
- Location: Tanzania

PersonKilled
- Person: 11 people
Outline of Talk

• The Machine Reading task
• UW reading system
  – Open Information Extraction
  – Map Open IE tuples to domain relations
• Steps in mapping to a domain
  – Recognize domain classes
  – Learn relation mapping rules
  – Learn semantic constraints
• Results for NFL and IC domains
• Future work, conclusions
Steps in Relation Mapping

1. Annotate tuple with domain-specific classes
   (a 14-13 victory, over, the 49ers)

2. Map tuples to domain relations
   Rule:  
   IF arg1 NFLGame & pred “over” & arg2 NFLTeam
   
   THEN gameLoser (NFLGame from arg1, NFLTeam from arg2)
   
   Confidence: 0.88
   
   gameLoser (NFLGame:a 14-13 victory; NFLTeam:49ers)

3. Apply semantic constraints
   Domain-independent classes (Person, Organization, etc.)
Step 1: Learn Class Recognizers

- Recognizer = regular expression with list of key words
- Current implementation
  - Examine training examples
  - Build lists by hand
  - About 3 hours for NFL or IC domains
- Future work
  - Learn key words from training examples
  - Up to 30 minutes verifying by hand
- Challenges
  - Synonyms not found in training
  - Ambiguity of key words (“New Orleans” not always team)
Step 2: Relation Mapping Rules

- **Rule learning** using a covering algorithm
  - Find “base rule” from seed instance, then generalize
  - Learns from a **small amount of training**
  - Domain classes allow **quick generalization**
  - Fully **automatic**

- **Active learning** to refine the rules
  - System presents user with **additional instances** to tag
  - System **updates confidence** of rules
  - Even **15-30 minutes** of tagging boosts rule precision
Step 3: Semantic Constraints

- Domain-specific classes are not enough
- Use domain-independent semantic constraints
  - Named Entity Recognizers
  - WordNet classes

“The Abbu Sayyaf killed three people”
KillingAgent = The Abbu Sayyaf

“security guards killed three people”
KillingAgent = security guards

“a car bomb killed three people”
KillingAgent = a car bomb
Learning Semantic Constraints

- Current implementation
  - List of semantic tests for each relation argument
  - Filters the output of relation mapping rules
  - About 10 hours manual effort for IC domain

- Future work
  - Learn semantic constraints from training instances
  - Tightly integrate with rule learning
  - Fully automatic
Outline of Talk

• The Machine Reading task
• UW reading system
  – Open Information Extraction
  – Map Open IE tuples to domain relations
• Steps in mapping to a domain
  – Recognize domain classes
  – Learn relation mapping rules
  – Learn semantic constraints
• Results for NFL and IC domains
• Future work, conclusions
Results for NFL Domain

- Minimal training.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Training</th>
<th>P</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>FinalScore</td>
<td>175</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GameLoser</td>
<td>121</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GameWinner</td>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GameDate</td>
<td>105</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Touchdown</td>
<td>48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FieldGoal</td>
<td>13</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results for NFL Domain

- Minimal training.
- High precision with 15 minutes active learning per relation.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Training</th>
<th>P</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>FinalScore</td>
<td>175</td>
<td>0.92</td>
<td>0.33</td>
</tr>
<tr>
<td>GameLoser</td>
<td>121</td>
<td>0.93</td>
<td>0.59</td>
</tr>
<tr>
<td>GameWinner</td>
<td>120</td>
<td>0.98</td>
<td>0.29</td>
</tr>
<tr>
<td>GameDate</td>
<td>105</td>
<td>0.68</td>
<td>0.79</td>
</tr>
<tr>
<td>Touchdown</td>
<td>48</td>
<td>0.83</td>
<td>0.14</td>
</tr>
<tr>
<td>FieldGoal</td>
<td>13</td>
<td>0.75</td>
<td>0.08</td>
</tr>
</tbody>
</table>

![NFL Domain](chart.png)

**Note:** The chart shows the precision-recall curve for different relations in the NFL domain.
Results for IC Domain

- Limited training, for this domain as well.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Training</th>
<th>P</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>HasMember</td>
<td>143</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PersonKilled</td>
<td>106</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KillingAgent</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KillingDate</td>
<td>53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KillingLocation</td>
<td>42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BombingLocation</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BombingDate</td>
<td>14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results for IC Domain

- Limited training, for this domain as well.
- High precision with up to 30 minutes active learning.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Training</th>
<th>P</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>HasMember</td>
<td>143</td>
<td>0.73</td>
<td>0.30</td>
</tr>
<tr>
<td>PersonKilled</td>
<td>106</td>
<td>0.78</td>
<td>0.32</td>
</tr>
<tr>
<td>KillingAgent</td>
<td>60</td>
<td>1.00</td>
<td>0.08</td>
</tr>
<tr>
<td>KillingDate</td>
<td>53</td>
<td>0.96</td>
<td>0.46</td>
</tr>
<tr>
<td>KillingLocation</td>
<td>42</td>
<td>0.70</td>
<td>0.90</td>
</tr>
<tr>
<td>BombingLocation</td>
<td>25</td>
<td>0.67</td>
<td>0.33</td>
</tr>
<tr>
<td>BombingDate</td>
<td>14</td>
<td>1.00</td>
<td>0.40</td>
</tr>
</tbody>
</table>
Effect of Active Learning

- 15-30 minutes active learning
- System selects instances for user to tag
- Raises confidence of reliable rules
Effect of Semantic Constraints

- **Filters** out incorrect argument values
- **Increases precision**, may reduce recall
Outline of Talk

• The Machine Reading task
• UW reading system
  – Open Information Extraction
  – Map Open IE tuples to domain relations
• Steps in mapping to a domain
  – Recognize domain classes
  – Learn relation mapping rules
  – Learn semantic constraints
• Results for NFL and IC domains
• Future work, conclusions
Future Work

• Minimize human effort
  – Learn class recognizers from training
    3 hours $\rightarrow$ 30 minutes
  – Learn semantic constraints from training
    10 hours $\rightarrow$ fully automatic
  – More effective Active Learning
    Select instances to refine rules for all seed instances

• Evaluate on “Surprise Domains”
  – Given 10-20 examples for each of 5-10 relations
  – Produce high-precision extractions
  – 2 day turn-around
Conclusions

- **Challenge:** rapid retargeting to customer ontology
- **Solution:**
  1. Open Information Extraction, no human effort
  2. Map to domain relations
     - Learn recognizers for domain classes
     - Learn relation mapping rules
     - Filter extractions with domain-independent semantic tests
- **High-precision extractions with minimal knowledge engineering**
  - Limited training, a few dozen examples
  - 15-30 minutes active learning per relation
  - 10-15 hours $\rightarrow$ 30 minutes of knowledge engineering
Thank you.