NYU at Cold Start 2015:
Experiments on KBC with NLP Novices

Yifan He   Ralph Grishman
Computer Science Department
New York University
The KBP Cold Start Task and Common Approaches

• The KBP Cold Start task builds a knowledge base from scratch using a given document collection and a predefined schema for the entities and relations.

• Common approaches
  
  • Hand-written rules (Grishman and Min, 2010)
  
  • Supervised relation classifiers
    
    • Weakly supervised classifiers: distant supervision (Mintz et al., 2009; Surdeanu et al., 2012), active learning / crowd sourcing (Angeli et al., 2014)
Focus this year: NLP Novices

- Current approaches often require NLP expertise
  - NYU rules are tuned every summer for 7 years
  - Supervised systems: annotation and algorithm design
  - Crowdsourcing: secret documents?

- Can a domain expert construct an in-house knowledge base from scratch, by herself, (using tools)?
NYU Cold Start Pipeline

Text Processing
- NP chunking
- Entity tagging
- Coreference

Core Tagger
- NP internal relations (titles, relatives)

Pattern Tagger
- Lexical and dependency paths

Distantly Supervised ME Tagger
- Align Freebase to TAC 2010 document collection

**Single Document**
- tool for domain experts to construct new entity type
- tool for domain expert to acquire relation extraction rules

**Cross Document Coref**
- Based on string matching
Entity Type and Relation Construction with ICE

- ICE [Integrated Customization Environment for Information Extraction]
  - easy tool for non-NLP experts to rapidly build customized IE systems for a new domain
- Entity set construction
- Relation extraction
Constructing Entity Sets

- New entity class (e.g. **DISEASE** in *per:cause_of_death*) by dictionary
  - users are not likely to do a good job assembling such a list
  - users are much better at reviewing a system-generated list
- Entity set expansion: start from 2 seeds, offer more to review
Ranking Entities

• Entities are represented with context vectors

• Contexts are dependency paths from and to the entity

• $V_{\text{heroin}}$: \{dobj\_sell:5, nn\_plant:3, dobj\_seize:4, …\}

• $V_{\text{heart\_attack}}$: \{prep\_from\_suffer:4, prep\_of\_die:3, …\}

• Entities ranked by distance to the cluster centroid (Min and Grishman, 2011)
Constructing Relations: Challenges

- Handle new entity types in relation (solved by entity set expansion: ICE recognizes **DISEASE** after it is built)

- Capture variations in linguistic constructions
  
  - **ORGANIZATION** *leader** **PERSON** vs. **ORGANIZATION** *revived under** **PERSON** (’s leadership)

- User comprehensible rules
Rules: Dependency Path

- Lexicalized dependency paths (LDPs) extractors
  - Simple, transparent approach; no feature engineering
  - Straightforward for bootstrapping
  - Most important component in NYU's slot-filling / cold start submissions (Sun et al. 2011; Min et al. 2012)

LDP
ORGANIZATION — dobj-1:revived:prep_under — PERSON

Can user understand this?
Comprehendible Rules:
Linearized LDPs

- Linearize LDP into English phrases
- User reviews linearized English phrases
- Based on word order in original sentence
- Insert syntactic elements for fluency: indirect objects, possessives etc.
- Lemmatize words except passive verbs
Bootstrapping: Finding Varieties in Rules

• Dependency path acquisition with the classical (active) Snowball bootstrapping (Agichtein and Gravano, 2000)

• Algorithm skeleton

1. User provide seeds
2. Collect arguments from seeds
3. New paths for review
4. Iterate
Experiments

• Entity set expansion and relation bootstrapping on Gigaword AP newswire 2008 data
  • Construct DISEASE entity type
  • Bootstrap all relations, only using seeds from slot descriptions

• **CoreTagger**: only use the core tagger which tags NP internal relations

• **Setting 1**: 5 iterations of bootstrapping, review 20 instances per iteration - 553 dependency path rules

• **Setting 2**: 5 iterations of bootstrapping, review as many phrases as possible, bootstrap with coreference (Gabbard et al., 2011) - 1,559 dependency path rules

• **“Proteus”**: NYU submission that uses 1,402 dependency patterns, 2,495 lexical patterns, and an add-on distantly supervised relation classifier
Experiments

• Entity set expansion and relation bootstrapping on Gigaword AP newswire 2008 data
  • Construct DISEASE entity type
  • Bootstrap all relations, only using seeds from slot descriptions
• **CoreTagger**: only use the core tagger which tags NP internal relations
• **Setting 1**: 5 iterations of bootstrapping, review 20 instances per iteration dependency path rules
• **Setting 2**: 5 iterations of bootstrapping, review as much as possible, bootstrap with coreference (*Gabbard et al.*, 2011) - 1,559 dependency path rules
• **“Proteus”**: NYU submission that uses 1,402 dependency patterns, 2,495 lexical patterns, and an add-on distantly supervised relation classifier

~20 min per relation
~1 hr per relation
7 summers
# Results: Hop0

<table>
<thead>
<tr>
<th>Model</th>
<th>P</th>
<th>R</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoreTagger</td>
<td>0.71</td>
<td>0.06</td>
<td>0.11</td>
</tr>
<tr>
<td>CoreTagger + Setting1</td>
<td>0.44</td>
<td>0.08</td>
<td>0.13</td>
</tr>
<tr>
<td>CoreTagger + Setting2</td>
<td>0.54</td>
<td>0.13</td>
<td>0.21</td>
</tr>
<tr>
<td>CoreTagger + Proteus</td>
<td>0.46</td>
<td>0.25</td>
<td>0.32</td>
</tr>
</tbody>
</table>

TAC 2014 Evaluation Data; Proteus = Patterns + Fuzzy Match + Distant Supervision
## Results: Hop0+Hop1

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>R</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoreTagger</td>
<td>0.47</td>
<td>0.04</td>
<td>0.07</td>
</tr>
<tr>
<td>CoreTagger + Setting1</td>
<td>0.34</td>
<td>0.05</td>
<td>0.08</td>
</tr>
<tr>
<td>CoreTagger + Setting2</td>
<td>0.37</td>
<td>0.08</td>
<td>0.13</td>
</tr>
<tr>
<td>CoreTagger + Proteus</td>
<td>0.31</td>
<td>0.20</td>
<td>0.24</td>
</tr>
</tbody>
</table>

TAC 2014 Evaluation Data; Proteus = Patterns + Fuzzy Match + Distant Supervision
Summary

• Pilot experiments on bootstrapping a KB constructor from scratch using an open-source tool

• Builds high-precision/modest recall KBs

• Friendly to domain experts who are not familiar with NLP: user only reviews plain English examples

• Builds rule-based interpretable models for both entity and relation recognition
More To Be Done

• Better annotation instance selection
  • So that the casual user can perform similarly to a serious user

• More expressive rules beyond dependency paths
  • Event extraction

• Leverage existing KB
Thank you

http://nlp.cs.nyu.edu/ice
http://github.com/rgrishman/ice
1. Preprocessing
2. Key phrase extraction
3. Entity set construction
4. Dependency paths extraction
5. Relation pattern bootstrapping
<table>
<thead>
<tr>
<th>Subject</th>
<th>Predicate</th>
<th>Object/Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>/m/0gg9kfr 2011 Christchurch earthquake</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/0j_2yw_ St Luke’s Church, Christchurch</td>
</tr>
<tr>
<td>/m/0gg9kfr 2011 Christchurch earthquake</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/0gg7hn1 Hotel Grand Chancellor, Christchurch</td>
</tr>
<tr>
<td>/m/0gg9kfr 2011 Christchurch earthquake</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/0by116z Christchurch Hospital</td>
</tr>
<tr>
<td>/m/0qtwtw9 Chelyabinsk Event</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/0r944hl Ice Palace &quot;Ural Lightning&quot;</td>
</tr>
<tr>
<td>/m/0qtwtw9 Chelyabinsk Event</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/0qzqcvy Chelyabinsk Zinc Factory</td>
</tr>
<tr>
<td>/m/0qtwtw9 Chelyabinsk Event</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/0tx4gt Chelyabinsk Drama Theatre</td>
</tr>
<tr>
<td>/m/0j0z2w4 Port Said Stadium disaster</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/064pnfg Traktor Ice Arena</td>
</tr>
<tr>
<td>/m/0gh6mkc 2011 Tōhoku earthquake and tsunami</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/0b7219 Port Said Stadium</td>
</tr>
<tr>
<td>/m/0gh6mkc 2011 Tōhoku earthquake and tsunami</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/02vk_7d Fukushima Daini Nuclear Power Plant</td>
</tr>
<tr>
<td>/m/0b4mlj Katowice Trade Hall roof collapse</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/02vkzy2 Fukushima Daiichi Nuclear Power Plant</td>
</tr>
<tr>
<td>/m/01v8cd Summerland disaster</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/02r05rb Katowice International Fair</td>
</tr>
<tr>
<td>/m/0dc3pc Royal Suspension Chain Pier</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/05bgr4 Summerland Leisure Centre</td>
</tr>
<tr>
<td>/m/05252dm Tay Bridge disaster</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/0dc3pc Royal Suspension Chain Pier</td>
</tr>
<tr>
<td>/m/098sht Buncefield fire</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/04zjqhp The Tay Bridge</td>
</tr>
<tr>
<td>/m/0d0vp3 September 11 attacks</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/098sp5 Buncefield oil depot</td>
</tr>
<tr>
<td>/m/0807k3 1983 United States Senate bombing</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/09w3b The Pentagon</td>
</tr>
<tr>
<td>/m/01y23_ 16th Street Baptist Church bombing</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/07vth United States Capitol</td>
</tr>
<tr>
<td>/m/0244k9 MGM Grand fire</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/0bf9_v 16th Street Baptist Church</td>
</tr>
<tr>
<td>/m/053zwd 1996 Garley Building fire</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/033vpy MGM Grand Las Vegas</td>
</tr>
<tr>
<td>/m/07hxss 1992 Windsor Castle fire</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/05bgkm Garley Building</td>
</tr>
<tr>
<td>/m/0b_94y Whiskey Au Go Go fire</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/0chgsms Windsor Castle</td>
</tr>
<tr>
<td>/m/02vnpxc Uphaar Cinema fire</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/05bgmws Whiskey Au Go Go</td>
</tr>
<tr>
<td>/m/0b27k1 Dee Bridge disaster</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/05bgrjk Uphaar Cinema</td>
</tr>
<tr>
<td>/m/0cfgmk Old Dee Bridge disaster</td>
<td>/event/disaster/structures_damaged</td>
<td>/m/0cfgmk Old Dee Bridge</td>
</tr>
</tbody>
</table>
Entity Set Expansion/ Ranking

• In each iteration, present the user with ranked entity list, ordered by the distance to the “positive centroid” (Min and Grishman, 2011):

\[ c = \frac{\sum_{p \in P} p}{|P|} - \frac{\sum_{n \in N} n}{|N|} \]

• where \( c \) is the positive centroid, \( P \) is the set of positive seeds (initial seeds and entities accepted by user), and \( N \) is the set of negative seeds (entities rejected by user)

• Update centroid for \( k \) iterations
Entity Representation

• Represent each phrase with a context vector, where contexts are dependency paths from and to the phrase

  • DRUGS share $dobj$(sell, X) and $dobj$(seize, X) contexts

  • DISEASE share prep_of(die, X) and prep_from(suffer) contexts

• Examples: count vectors of dependency contexts

  • $V_{\text{heroin}}$: \{dobj\_sell:5, nn\_plant:3, dobj\_seize:4, \ldots\}

  • $V_{\text{heart\_attack}}$: \{prep\_from\_suffer:4, prep\_of\_die:3, \ldots\}

• Features weighted by PMI; word embedding on large data sets for dimension reduction
Entity Representation II

• Using raw vectors cannot provide live response

• Dimension reduction via word embeddings

• Skip-gram model with negative sampling, using dependency context (Levy and Goldberg, 2014a)

• Equivalent of factorization of the original* feature matrix (Levy and Goldberg, 2014b)

* shifted; PPMI instead of PMI0
Experiment of Entity Set Expansion

• Finding Drugs in Drug Enforcement Agency news releases

• 10 iterations, review 20 entity candidates per iteration

• Measure recall on a pre-compiled list of 181 drug names from 2,132 key phrases

• DISEASES: ICE 129 diseases; Manual 19 diseases
Constructing Drugs Type

Recall of DRUGS

- DRUGS using PMI matrix
- DRUGS using embeddings
Constructing **Drugs** Type (Weighted Result)

- Recall score weighted by frequency of entities

---

**Recall of DRUGS (Weighted)**

- Iteration 1
- Iteration 2
- Iteration 3
- Iteration 4
- Iteration 5
- Iteration 6
- Iteration 7
- Iteration 8
- Iteration 9
- Iteration 10

- Orange line: DRUGS using PMI matrix
- Blue line: DRUGS using embeddings

- Recall score weighted by frequency of entities
• 84 positive examples from 2,132 candidates
## Results: Hop0 - w/ FM

<table>
<thead>
<tr>
<th>Model</th>
<th>P</th>
<th>R</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoreTagger</td>
<td>0.71</td>
<td>0.06</td>
<td>0.11</td>
</tr>
<tr>
<td>CoreTagger + Setting1</td>
<td>0.44</td>
<td>0.08</td>
<td>0.13</td>
</tr>
<tr>
<td>CoreTagger + Setting2</td>
<td>0.41</td>
<td>0.11</td>
<td>0.18</td>
</tr>
<tr>
<td>CoreTagger + Proteus</td>
<td>0.46</td>
<td>0.25</td>
<td>0.32</td>
</tr>
</tbody>
</table>

TAC 2014 Evaluation Data; Proteus = Patterns + Fuzzy Match + Distant Supervision
Results: Overall - w/ FM

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>R</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoreTagger</td>
<td>0.47</td>
<td>0.04</td>
<td>0.07</td>
</tr>
<tr>
<td>CoreTagger + Setting1</td>
<td>0.34</td>
<td>0.05</td>
<td>0.08</td>
</tr>
<tr>
<td>CoreTagger + Setting2</td>
<td>0.31</td>
<td>0.10</td>
<td>0.15</td>
</tr>
<tr>
<td>CoreTagger + Proteus</td>
<td>0.31</td>
<td>0.20</td>
<td>0.24</td>
</tr>
</tbody>
</table>

TAC 2014 Evaluation Data; Proteus = Patterns + Fuzzy Match + Distant Supervision
Fuzzy dependency path match for small rule set

• Improve recall for small rule sets
  • Also tested in our 2015 KBP Cold Start submission
• Match two LDPs with edit distance on dependency chains
  • Weight of edit operations set by grid search on dev set (substitution: 0.8, insertion: 1.2, deletion: 0.3; feature-based see paper)
  • Substitution cost determined by word similarity based on word embeddings
Fuzzy dependency path match-based extraction: example

<table>
<thead>
<tr>
<th></th>
<th>dsubj:END$</th>
<th>0.3</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>nsubj-1:distribute</td>
<td>0.28*0.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Edit costs
substitution: 0.8
insert: 1.2
delete: 0.3

\[
\text{cost} = \frac{\text{weightedDistance} \cdot |\text{rule}|}{3} = \frac{0.28 \times 0.8 + 0.3}{3} = 0.17
\]
## Official Run Results

<table>
<thead>
<tr>
<th></th>
<th>NestedNames+Pattern+DS+FM</th>
<th></th>
<th>Pattern+DS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
<td>P</td>
</tr>
<tr>
<td>Hop0</td>
<td>0.44</td>
<td>0.20</td>
<td>0.27</td>
<td>0.51</td>
</tr>
<tr>
<td>Hop1</td>
<td>0.06</td>
<td>0.09</td>
<td>0.07</td>
<td>0.15</td>
</tr>
<tr>
<td>MicroAvg</td>
<td>0.17</td>
<td>0.15</td>
<td>0.16</td>
<td>0.30</td>
</tr>
<tr>
<td>MacroAvg</td>
<td></td>
<td></td>
<td>0.18</td>
<td></td>
</tr>
</tbody>
</table>

Main goal: testing the fuzzy match paradigm
False positives on NIL slots from Fuzzy Match in Hop 0 was penalized heavily in Hop 1