CMU LTI @ KBP 2015 Event Track

Zhengzhong Liu
Dheeru Dua
Jun Araki
Teruko Mitamura
Eduard Hovy

LTI Carnegie Mellon University
Event Nugget Detection
Nugget Detection

1. Three tasks:
   a. Detect the spans that corresponds to event mentions
   b. Detect the event nugget type
   c. Detect the Realis Status
2. New Challenge:
   a. Double tagging
1. Discriminatively trained CRF.
   a. Test with averaged perceptron
2. Handle double tagging by combining the multiple types into a new label.
3. Each nugget is predicted independently.
### Combining Event Types?

<table>
<thead>
<tr>
<th>Event Type 1</th>
<th>Event Type 2</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Justice_Execute ; Life_Die</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>Transaction_Transfer-Ownership ; Movement_Transport-Artifact</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>Life_Die ; Conflict_Attack</td>
<td></td>
<td>48</td>
</tr>
<tr>
<td>Transaction_Transfer-Ownership ; Transaction_Transfer-Money</td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>Conflict_Attack ; Life_Die</td>
<td></td>
<td>69</td>
</tr>
<tr>
<td>Justice_Extradite ; Movement_Transport-Person</td>
<td></td>
<td>39</td>
</tr>
</tbody>
</table>

**Total Possible Joint Type: 34**
Combining Event Types?

- You can even infer the text by looking at the types.
  - Smuggling (all 3 types all the time)
    - Transaction_Transfer-Money ; Movement_Transport-Artifact ; Transaction_Transfer-Ownership
  - Conflict_Attack ; Transaction_Transfer-Ownership
    - Hijacking, rob, burglary, seize

- Nugget Type detection is similar to WSD with the detailed ontology.

- Joint type should share information with its original types.
  - So the features are extracted on both the joint and splitted version
LTI 1 Features

- **Standard Linguistic Features:**
  - Part-of-Speech, lemma, named entity tag of the following:
    - The 2-word window of the trigger (both side)
    - The trigger word itself
    - Direct dependent words of the trigger
    - Dependent head of the trigger

- **Ontology:**
  - Brown clusters (8, 12, 16 bits)
  - WordNet Synonym and Noun derivative forms of the trigger
  - FrameNet Type

See our system at the end for details
LTI1 Features

- **Selected WordNet senses in the context:**
  - "Leader", "Worker", "Body Part", "Monetary System", "Possession", "Government", "Crime" and "Pathological State" (More on this later)
  - Whether surrounding words match such sense
  - Whether argument of mention match such sense (arguments from semantic roles)

- **Semantic role features:**
  - The frame name (mentioned above)
  - The argument’s role, named entity tag, and headword lemma

See our system at the end for details
1. CRF trained with Passive-Aggressive Perceptron.
2. Multi-tagging handling:
   a. Merging sequence from the top 5 series
   b. Training: Optimize top 5-best sequence
1. Normalized the top scores and take the largest gap.
2. $p=0.4$, $\varepsilon$ is 0.01.

**Input:** 5-Best-Viterbi-Pares $(y_j, \text{score}_j), 1 \leq j \leq 5$

**Input:** Threshold, $\varepsilon$ and Scaling Factor $p$

**Output:** Top-n Predicted Sequence

1. $\textit{prediction} \leftarrow \{y_1\}$
2. $\mu \leftarrow \frac{\sum_{i=1}^{5} \text{score}_i}{5}$
3. $\sigma \leftarrow \sqrt{\frac{\sum_{i=1}^{5} (\text{score}_i - \mu)^2}{5}}$
4. for $j \leftarrow 2...5$ do
5. $N\text{score}_j = [(\text{score}_j - \mu)/\sigma] * p$
6. if $N\text{score}_j - N\text{score}_{j+1} \leq \varepsilon$ then
7. $\textit{prediction} \leftarrow \textit{prediction} \cup y_j$
8. else
9. return $\textit{prediction}$
LTI2: Features

- POS tags, in the 5 word window.
- Ontology:
  - Brown clusters with 13 bits
  - Lemmas of the event trigger in the WordNet hierarchy
- History:
  - 2 verbs in past and future
  - 2 events trigger seen in the history
- Event arguments types from SRL followed by NER of the arguments.
- Recall Mode:
  - 8 bit Brown cluster, a gazetteer of event triggers and WordNet synsets
Realis Classification

1. Linear SVM model.
2. Basic features are borrowed from type detection:
   a. All lexicalized features are removed to avoid overfitting
   b. One feature to see if the phrase is “in quote”
3. Done after span and type detection.

See our system at the end for details.
Results (LTI1 on Dev, 5-fold aver.)

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain</td>
<td>74.36</td>
<td>55.722</td>
<td>63.622</td>
</tr>
<tr>
<td>Type</td>
<td>67.08</td>
<td>50.25</td>
<td>57.382</td>
</tr>
<tr>
<td>Realis</td>
<td>51.788</td>
<td>38.754</td>
<td>44.274</td>
</tr>
<tr>
<td>Type+Realis</td>
<td>46.288</td>
<td>34.626</td>
<td>39.562</td>
</tr>
</tbody>
</table>
**Results (Realis On Dev with Gold Mentions)**

- Realis itself is difficult.
- It is more serious with imperfect mention types.

<table>
<thead>
<tr>
<th>Fold</th>
<th>Prec</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fold 1</td>
<td>71.68</td>
<td>71.63</td>
<td>71.66</td>
</tr>
<tr>
<td>Fold 2</td>
<td>64.06</td>
<td>64.06</td>
<td>64.06</td>
</tr>
<tr>
<td>Fold 3</td>
<td>62.07</td>
<td>61.96</td>
<td>62.02</td>
</tr>
<tr>
<td>Fold 4</td>
<td>72.66</td>
<td>72.66</td>
<td>72.66</td>
</tr>
<tr>
<td>Fold 5</td>
<td>62.21</td>
<td>62.21</td>
<td>62.21</td>
</tr>
<tr>
<td><strong>Aver.</strong></td>
<td><strong>66.536</strong></td>
<td><strong>66.504</strong></td>
<td><strong>66.522</strong></td>
</tr>
</tbody>
</table>
### Final Results on Evaluation Set

<table>
<thead>
<tr>
<th></th>
<th>Prec</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LTI1</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plain</td>
<td>82.46</td>
<td>50.3</td>
<td>62.49</td>
</tr>
<tr>
<td>Type</td>
<td>73.68</td>
<td>44.94</td>
<td>55.83</td>
</tr>
<tr>
<td>Realis</td>
<td>62.09</td>
<td>37.87</td>
<td>47.05</td>
</tr>
<tr>
<td>All</td>
<td>55.12</td>
<td>33.62</td>
<td>41.77</td>
</tr>
<tr>
<td><strong>LTI2</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plain</td>
<td>77</td>
<td>39.53</td>
<td>52.24</td>
</tr>
<tr>
<td>Type</td>
<td>68.79</td>
<td>35.31</td>
<td>46.67</td>
</tr>
<tr>
<td>Realis</td>
<td>51.41</td>
<td>26.39</td>
<td>34.88</td>
</tr>
<tr>
<td>All</td>
<td>45.47</td>
<td>23.34</td>
<td>30.85</td>
</tr>
</tbody>
</table>
# Results after fixing LTI2 format error

## LTI2-Prec

<table>
<thead>
<tr>
<th></th>
<th>Prec</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain</td>
<td>81.7</td>
<td>44.36</td>
<td>57.52</td>
</tr>
<tr>
<td>Type</td>
<td>72.91</td>
<td>39.56</td>
<td>51.29</td>
</tr>
<tr>
<td>Realis</td>
<td>61.84</td>
<td>33.55</td>
<td>43.50</td>
</tr>
<tr>
<td>All</td>
<td>55.37</td>
<td>30.04</td>
<td>38.9</td>
</tr>
</tbody>
</table>

## LTI2-Recall

<table>
<thead>
<tr>
<th></th>
<th>Prec</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain</td>
<td>77.59</td>
<td>49.14</td>
<td>60.17</td>
</tr>
<tr>
<td>Type</td>
<td>69.61</td>
<td>44.08</td>
<td>53.98</td>
</tr>
<tr>
<td>Realis</td>
<td>52.71</td>
<td>38.38</td>
<td>40.87</td>
</tr>
<tr>
<td>All</td>
<td>47.17</td>
<td>29.87</td>
<td>36.58</td>
</tr>
</tbody>
</table>
Future work

1. Hand selected WordNet senses can be replaced by statistical methods
   a. NPMI between WordNet Sense and the type:

<table>
<thead>
<tr>
<th>Sense</th>
<th>Type</th>
<th>NPMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>census</td>
<td>Life_Divorce</td>
<td>0.6645</td>
</tr>
<tr>
<td>harassment</td>
<td>Justice_Sue</td>
<td>0.6641</td>
</tr>
<tr>
<td>declaration</td>
<td>Justice_Charge-Indict</td>
<td>0.6636</td>
</tr>
<tr>
<td>manufacturer</td>
<td>Manufacture_Artifact</td>
<td>0.6611</td>
</tr>
<tr>
<td>destination</td>
<td>Life_Marry</td>
<td>0.6595</td>
</tr>
<tr>
<td>government</td>
<td>Justice_Appeal</td>
<td>0.2502</td>
</tr>
</tbody>
</table>
Future work

1. Model inter-mention dependencies.
2. And of course, continuous representation can be helpful.
Event Hopper Coreference
Hopper Coreference

1. Identify Full Event Coreference links.
2. Given Information:
   a. Event Nuggets given, including the span, Event types and subtypes, and Realis
3. 2 Individual system with 3 submissions.
   a. We focus on our best system in the presentation
The Model

1. Latent Antecedent Tree
2. Represent cluster as a tree.
   a. Note that a coreference can be represented as multiple trees
3. Best First Decoding
   a. Favor “easy” decisions
   b. Ng & Cardie 2002

Input: Training data D, number of iterations T
Output: Weight vector w
1: \( w = \vec{0} \)
2: for \( t \leftarrow 1..T \) do
3:   for \( \langle M_i, A_i, \tilde{A}_i \rangle \in D \) do
4:     \( \hat{y}_i = \arg\max_{y(A)} \text{score}(y) \)
5:     if \( \neg\text{Correct}(\hat{y}_i) \) then
6:       \( \tilde{y}_i = \arg\max_{y(\tilde{A})} \text{score}(y) \)
7:       \( \Delta = \Phi(\hat{y}_i) - \Phi(\tilde{y}_i) \)
8:       \( \tau = \frac{\Delta \cdot w}{\|\Delta\|^2} \)
9:       \( w = w + \tau \Delta \)

return \( w \)

Fernandes et. al. 2012; Björkelund & Kuhn 2014
The LAT model

1. The Gold Tree:
   a. The best tree under current parameters

2. Predicted Tree:
   a. Prediction made with the Best-First algorithm

3. If clusters are difference, then penalize.


---

**Input:** Training data $D$, number of iterations $T$

**Output:** Weight vector $w$

1. $w = \vec{0}$

2. for $t \leftarrow 1..T$ do

3. for $(M_i, A_i, \tilde{A}_i) \in D$ do

4. $\hat{y}_i = \arg \max_{y(A)} \text{score}(y)$

5. if $\neg \text{Correct}(\hat{y}_i)$ then

6. $\tilde{y}_i = \arg \max_{y(\tilde{A})} \text{score}(y)$

7. $\Delta = \Phi(\hat{y}_i) - \Phi(\tilde{y}_i)$

8. $\tau = \frac{\Delta \cdot w}{||\Delta||^2}$

9. return $w = w + \tau \Delta$
The LAT model

- loss = 1.5
- loss = 1
- loss = 1
- loss = 1
Features for coreference

1. Trigger Match - exact and fuzzy match on the trigger word
   a. uses standard linguistic features (pos, lemma, etc.)
   b. resources like Brown Clustering and WordNet.
   c. Information from mention type and realis type are also used

2. Argument match - exact and fuzzy match on the arguments
   a. String matches (head word, substring)
   b. Argument role
   c. Entity coreference information (From stanford)

3. Discourse features
   a. encodes sentence and mention distances

See our system at the end for details
Catch 1: The Importance of PA-algorithm

1. Passive Aggressive algorithm capture the loss term
   a. Penalize more if the tree differs a lot
2. We found that without using the PA-algorithm, it is hard to converge
3. Observations:
   a. Most clusters predictions are wrong -> Update is done almost all the time
   b. Some features differs between Forum dataset and News dataset -> e.g. Distance between mentions
Catch 2: Averaging parameters matters

1. During training, we found different training sequence change the final model a lot.
2. However, the change is small with averaged perceptron.
3. Averaged score is also much better.

- Both problems might be caused by the data (i.e. multi-genre data without considering their differences)
### 5-fold results (Averaged vs. Vanilla)

<table>
<thead>
<tr>
<th></th>
<th>Average Perceptron</th>
<th>Vanilla Perceptron</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV0</td>
<td>83.08</td>
<td>79.16</td>
</tr>
<tr>
<td>CV1</td>
<td>78.53</td>
<td>72.72</td>
</tr>
<tr>
<td>CV2</td>
<td>75.80</td>
<td>75.13</td>
</tr>
<tr>
<td>CV3</td>
<td>77.15</td>
<td>69.63</td>
</tr>
<tr>
<td>CV4</td>
<td>74.20</td>
<td>61.94</td>
</tr>
<tr>
<td>Average</td>
<td><strong>77.75</strong></td>
<td>71.71</td>
</tr>
</tbody>
</table>
## Off-cycle Evaluation (Full Pipeline)

<table>
<thead>
<tr>
<th></th>
<th>BCubed</th>
<th>Ceafe</th>
<th>MUC</th>
<th>BLANC</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUR_PIPELINE</td>
<td>73.01</td>
<td>65.41</td>
<td>59.10</td>
<td>59.33</td>
<td>64.72</td>
</tr>
<tr>
<td>System 1</td>
<td>69.65</td>
<td>64.55</td>
<td>56.86</td>
<td>59.51</td>
<td>63.23</td>
</tr>
<tr>
<td>System 2</td>
<td>67.27</td>
<td>61.35</td>
<td>63.93</td>
<td>58.52</td>
<td>62.95</td>
</tr>
<tr>
<td>System 3</td>
<td>68.28</td>
<td>61.99</td>
<td>61.85</td>
<td>58.05</td>
<td>62.80</td>
</tr>
<tr>
<td>System 4</td>
<td>67.80</td>
<td>61.62</td>
<td>62.30</td>
<td>57.79</td>
<td>62.63</td>
</tr>
</tbody>
</table>
Future Work

1. Consider genre specific features.
   a. We might train each genre independently
   b. Even better, consider only those features that might be affected by the genres (see next slide)
   c. For example, you will find a mention per 13.6 tokens in news but 25.3 tokens in forum.

2. Consider global features.
   a. It is not yet clear what global features can be useful to hopper coreference
Future Work

1. Consider interactions between mention detection.
2. Consider discourse level analysis.
Thank You!
Questions?

Our code here!

Might be hard to set up, but you can still have a look!

We are also working to integrate it into the DEFT project.
References


