Event Detection and Domain Adaptation with Convolutional Neural Networks

Written by Thien Huu Nguyen and Ralph Grishman for ACL2015
Presented by Xia Cui for NLP@UoL
Overview

• First work on Event Detection using Convolutional Neural Networks
• First Research on Domain Adaptation using CNNs
• Overcome two fundamental limitations of the traditional feature-based approach
  • Complicated feature engineering for rich feature sets
  • Error propagation from the preceding stages which generate these features
Event Detection (ED)

- Identifying instances of specified types of events in text
- Associated with each event mention, identify event triggers and classify them into specific types
  - Event mention: event itself
  - Event trigger: evokes the event
- Example:
  
  A police officer was killed in New Jersey today.
  
  “killed” is the trigger for the event “Die”
Challenges and Problems

• The same event might appear in the form of various trigger expressions and an expression might represent different events in different contexts → event extraction and event argument discovery

• Two problems in previous works
  • *Feature selection is manual*, requires linguistic intuition and domain expertise implying additional studies for new application domains and limiting the capacity to quickly adapt to new domains
  • *Supervised NLP toolkits and resources for feature extraction might involve error* (due to the imperfect nature or the performance loss of the toolkits on new domains)
Solution: Convolutional Neural Network (CNN)

• Automatically learns features from sentences, and minimises the dependence on supervised toolkits and resources for features

• Demonstrates significantly outperform traditional feature-based methods
  • Due to the capacity to mitigate the error propagation from the pre-processing modules for features
  • Due to the use of word embeddings to induce a more general representation for trigger candidates
Token Representations

• Event Detection $\rightarrow$ Multiclass classification problem
• $x = [x_{-w}, x_{-w+1}, ..., x_0, ..., x_{w-1}, x_w]$
  • $x$: event trigger (token)
  • $2w + 1$: fixed window size
  • $m_t \times (2w + 1)$: matrix size
  • $m_t$: dimensionality of the concatenated vectors of the tokens
Embedding Tables

• Word Embedding Table
  • Pretrained, semantic and syntactic properties

• Position Embedding Table
  • Initialized randomly, embed the relative distance $i$ of token $x_i$ to current token $x_0$

• Entity Type Embedding Table
  • Initialized randomly, using known entity type associated with each token in the sentence, BIO annotation scheme to assign entity type labels to each token in the trigger candidate
input sentence with marked trigger candidate

BIO Entity Type Annotation: A police officer was <anchor>killed</anchor> in New Jersey today

{\(f_1, \ldots, f_n\)}, each \(f_i\) window size \(k\),
matrix size \(m_t \times k\)

Gradients by back-propagation

Training: via stochastic gradient descent with shuffled mini-batches and AdaDelta update rule (Kim, 2014)
Experiment Settings

• Multiple window size (in convolution layer): {2, 3, 4, 5}
  • 150 feature maps for each window size
• Window size for the trigger: 31
• Dimensionality of pretrained word embeddings from word2vec: 300
• Dimensionality of position embeddings: 50
• Dimensionality of entity type embeddings: 50
• Parameters taken from Kim (2014):
  • Dropout rate $\rho = 0.5$, mini-batch size = 50, $l_2$ norms = 3
Experiment Datasets

- **ACE2005 corpus**
  - 34-class (33 types + 1 “none”)
  - Test set: 40 newswire articles (672 sentences)
  - Development set: 30 other documents (836 sentences)
  - Training set: remaining 529 documents (14,849 sentences)

<table>
<thead>
<tr>
<th>Systems</th>
<th>-Position</th>
<th>P</th>
<th>R</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Entity Types</td>
<td>-Position</td>
<td>16.8</td>
<td>12.0</td>
<td>14.0</td>
</tr>
<tr>
<td></td>
<td>+Position</td>
<td>75.0</td>
<td>63.0</td>
<td>68.5</td>
</tr>
<tr>
<td>+Entity Types</td>
<td>-Position</td>
<td>17.0</td>
<td>15.0</td>
<td>15.9</td>
</tr>
<tr>
<td></td>
<td>+Position</td>
<td>75.6</td>
<td>66.4</td>
<td>70.7</td>
</tr>
</tbody>
</table>

Table 1: Performance on the Development Set.
Performance Comparison

<table>
<thead>
<tr>
<th>Methods</th>
<th>P</th>
<th>R</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentence-level in Hong et al. (2011)</td>
<td>67.6</td>
<td>53.5</td>
<td>59.7</td>
</tr>
<tr>
<td>MaxEnt with local features in Li et al. (2013b)</td>
<td>74.5</td>
<td>59.1</td>
<td>65.9</td>
</tr>
<tr>
<td>Joint beam search with local features in Li et al. (2013b)</td>
<td>73.7</td>
<td>59.3</td>
<td>65.7</td>
</tr>
<tr>
<td>Joint beam search with local and global features in Li et al. (2013b)</td>
<td>73.7</td>
<td>62.3</td>
<td>67.5</td>
</tr>
<tr>
<td>Cross-entity in Hong et al. (2011) †</td>
<td>72.9</td>
<td>64.3</td>
<td>68.3</td>
</tr>
<tr>
<td>CNN1: CNN without any external features</td>
<td>71.9</td>
<td>63.8</td>
<td>67.6</td>
</tr>
<tr>
<td>CNN2: CNN augmented with entity types</td>
<td>71.8</td>
<td>66.4</td>
<td>69.0</td>
</tr>
</tbody>
</table>

Table 2: Performance with Gold-Standard Entity Mentions and Types. † beyond sentence level.

- External features are the features generated from the supervised NLP modules and manual resources such as parsers, name taggers, entity mention extractors, gazetteer etc.

<table>
<thead>
<tr>
<th>Methods</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentence level in Ji and Grishman (2008)</td>
<td>59.7</td>
</tr>
<tr>
<td>MaxEnt with local features in Li et al. (2013b)</td>
<td>64.7</td>
</tr>
<tr>
<td>Joint beam search with local features in Li et al. (2013b)</td>
<td>63.7</td>
</tr>
<tr>
<td>Joint beam search with local and global features in Li et al. (2013b)</td>
<td>65.6</td>
</tr>
<tr>
<td>CNN1: CNN without any external features</td>
<td>67.6</td>
</tr>
</tbody>
</table>

Table 3: Performance with Predicted Entity Mentions and Types.
Domain Adaptation

- Goal: develop techniques taking training data in some *source domain* and learning models that can work well on *target domains*
- Datasets: ACE2005 corpus
  - **6 different domains:** broadcast conversation (bc), broadcast news (bn), telephone conversation (cts), newswire (nw), usenet (un) and weblogs (wl)
  - **Source domain:** news (bn ∪ nw)
  - **Target domains:** bc, cts, wl (3 different domains)
  - **Development set:** half of bc
  - **Test set:** remaining half of bc
In-domain & out-of-domain Performance

<table>
<thead>
<tr>
<th>System</th>
<th>In-domain(bn+nw)</th>
<th>bc</th>
<th>cts</th>
<th>wl</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
<td>P</td>
</tr>
<tr>
<td>MaxEnt</td>
<td>74.5</td>
<td>59.4</td>
<td>66.0</td>
<td>70.1</td>
</tr>
<tr>
<td>Joint beam search in Li et al. (2013b)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joint+Local</td>
<td>73.5</td>
<td>62.7</td>
<td>67.7</td>
<td>70.3</td>
</tr>
<tr>
<td>Joint+Local+Global</td>
<td>72.9</td>
<td>63.2</td>
<td>67.7</td>
<td>68.8</td>
</tr>
<tr>
<td>CNN1</td>
<td>70.9</td>
<td>64.0</td>
<td>67.3</td>
<td>71.0</td>
</tr>
<tr>
<td>CNN2</td>
<td>69.2</td>
<td>67.0</td>
<td><strong>68.0</strong></td>
<td>70.2</td>
</tr>
</tbody>
</table>

Table 4: In-domain (first column) and Out-of-domain Performance (columns two to four). Cells marked with † designate CNN models that significantly outperform ($p < 0.05$) all the reported feature-based methods on the specified domain.

The performance of feature-based systems MaxEnt, Joint+Local and Joint+Local+Global are obtained from the actual systems in Li et al. (2013b). CNN1 and CNN2 via 5-fold cross validation.