Exploratory Neural Relation Classification for Domain Knowledge Acquisition

Yan Fan, Chengyu Wang, Xiaofeng He

School of Computer Science and Software Engineering
East China Normal University
Shanghai, China
Outline

• Introduction
• Related Work
• Proposed Approach
• Experiments
• Conclusion
Relation Extraction

• **Relation extraction**
  – Structures the information from the Web by annotating the plain text with entities and their relations
    • E.g., “*Inception* is directed by *Christopher Nolan.*”
      \[
      \text{entity}_1 \quad \text{relation} \quad \text{entity}_2
      \]

• **Relation classification**
  – Formulates relation extraction as a classification problem
    • E.g., *(Inception, Christopher Nolan)* should be classified as the relation “directed by”, instead of “played by”.

Domain Knowledge Acquisition

- **Knowledge graph**
  - Relation extraction is a key technique in constructing knowledge graphs.

- **Challenges for domain knowledge graph**
  - **Long-tail domain entities**: Most domain entities which follow long-tail distribution, leading to the context sparsity problem for pattern-based methods.

  - **Incomplete predefined relations**: Since predefined relations are limited, unlabeled entity pairs may be wrongly forced into existing relation labels.
Dynamic Structured Neural Network for Exploratory Relation Classification

• **Goal**
  1. Classifies entity pairs into a finite pre-defined relations
  2. Discovers new relations and instances from plain texts with high confidence

• **Method**
  – **Context sparsity problem:** A *distributional embedding* layer is introduced to encode corpus-level semantic features of domain entities.
  – **Limited label assignment:** A *clustering method* is proposed to generate new relations from unlabeled data which can not be classified to be any existing relations.
Outline

• Introduction
• Related Work
• Proposed Approach
• Experiments
• Conclusion
Relation Classification Approaches

• **Traditional approaches**
  – Feature-based: applies textual analysis
    • N-grams, POS tagging, NER, dependency parsing
  – Kernel-based: similarity metric in higher dimensional space
    • Kernel functions are applied to strings, word sequences, parsing trees
  – Requires **empirical features** or well-designed kernel functions

• **Deep learning models**
  – Distributional representation: word embeddings
  – Neural network models:
    • CNN: extracts features with local information
    • RNN: captures long-term dependency on the sequence
  – Automatically extracts features
Relation Discovery Approaches

• **Open relation extraction**
  - automatically discovers relations from large-scale corpus with limited seed instances or patterns without predefined types
  - Representative systems: TextRunner, ReVerb, OLLIE
  - Inapplicable to domain knowledge due to data **sparsity problem**

• **Clustering-based approaches**
  - Predefined K: Standard KMeans
  - Automatically learned K: Non-parametric Bayesian models
    - Chinese restaurant process (CRP), distance dependent CRP (ddCRP)
Outline

- Introduction
- Related Work
- Proposed Approach
- Experiments
- Conclusion
Task Definition

• **Notations**
  - Labeled entity pair set $X^l = \{(e_1, e_2)\}$ and their labels $Y^l$
  - Unlabeled entity pair set $X^u = \{(e_1, e_2)\}$

• **Exploratory relation classification (ERC)**
  - Trains a model to predict the relations for entity pairs in $X^u$ with $K + n$ output labels, where $K$ denotes the number of pre-defined relations in $Y^l$, and $n$ is the number of newly discovered relations.
### General Framework

**Algorithm 1 ERC Training Process**

**Input:** Labeled data $X^l$ and $Y^l$, unlabeled data $X^u$

**Output:** Expanded relation set $R_{new}$

1. **while** no new relations can be discovered **do**
2.   // **Base neural network training**
3.   Train base neural network $N_t$ with $X^l$ and $Y^l$
4.   // **Relation discovery**
5.   Generate candidate clusters $\{C_1, \ldots, C_m\}$ for $X^u$
6.   Pick the best cluster $C^*$ from $\{C_1, \ldots, C_m\}$
7.   Update relation set $R_{new} = R_{new} \cup \{C^*\}$
8.   // **Relation prediction**
9.   Predict confident labels for unlabeled data $X^u$ on $R_{new}$
10. **end while**
11. **return** $R_{new}$
Base Neural Network Training

**Syntactic contexts via LSTM**
- Nodes on the root augmented dependency path (RADP)
  - E.g. [Inception, directed, Christopher Nolan]
- Node representation
  - \{word embedding, POS tag, dependency relation, relational direction\}
  - E.g. \{Inception, nnp, nsubjpass, <-\}

**Lexical contexts via CNN**
- Word embeddings of sliding window of n-grams around entities

**Semantic contexts**
- Word embeddings of two tagged entities
Base Neural Network Architecture

- **Word embedding**
- **Part-of-speech tag**
- **Dependency relation**
- **Relational direction**

Diagram:
- **Syntactic context**
- **Lexical context**
- **Semantic context**
- **Softmax**
- **LSTM network**
- **Lookup table**
- **Input**

Example Sentence:
Inception is directed by Christopher Nolan
Chinese Restaurant Process (CRP)

- **Goal**
  - Groups customers into random tables where they sit

- **Distribution over table assignment**

\[
\Pr(z_i = p \mid \bar{z}_{-i}, \alpha) \propto \begin{cases} 
N_p & \text{if } p \leq K \\
\alpha & \text{if } p = K + 1 
\end{cases}
\]

- \(N_p\): number of customers sitting at table \(p\)
- \(z_i\): index of the table where the \(i\)-th customer sits
- \(\bar{z}_{-i}\): indices of tables for customers except for the \(i\)-th customer
- \(\alpha\): scaling parameter for a new table
- \(K\): number of occupied tables
Similarity Sensitive Chinese Restaurant Process (ssCRP)

- **Idea**
  - Exploits similarities between customers
  - Turns the problem to customer assignment

- **Distribution over customer assignment**

  \[
  \Pr(c_i = j \mid \eta) \propto \begin{cases} 
  \alpha & \text{if } j \text{ is customer } i \text{ itself} \\
  g(s_{ij}) & \text{if } j \text{ is an upcoming customer} \\
  g(s_{ij})(1 + \beta \lg N_p) & \text{if } j \text{ is averaged from table } p
  \end{cases}
  \]

  - \( s_{ij} \): similarity score between the \( i \)-th and \( j \)-th customer
  - \( g(x) \): similarity function to magnify input differences
  - \( \beta \): the parameter balancing the weight of table size
  - \( \eta = \{S, N_p, \alpha, \beta\} \): set of hyperparameters
Illustration of ssCRP

Step 1: set fixed tables
(result of the base neural network)

Step 2: draw customer assignments
for multiple times

Step 3: generate tables

Step 4: pick the best table

Step 5: map the table to a relation
Relation Prediction

• **Idea**
  – Populates small clusters generated via ssCRP
  – Enriches existing relations with more instances

• **Prediction criteria**
  – Distribution over $K + l$ relations for entity pair $(e_1, e_2)$:
    $$[\Pr(r_1|e_1, e_2), \ldots, \Pr(r_{K+l}|e_1, e_2)]$$
  – “Max-secondMax” value for “near uniform” criteria:
    $$\text{conf}(e_1, e_2) = \frac{\max([\Pr(r_1|e_1, e_2), \ldots, \Pr(r_{K+l}|e_1, e_2)])}{\text{secondMax}([\Pr(r_1|e_1, e_2), \ldots, \Pr(r_{K+l}|e_1, e_2)])}$$
Outline

• Introduction
• Related Work
• Proposed Approach
• Experiments
• Conclusion
Experimental Data

- **Text corpus**
  - Text contents from 37,746 pages of entertainment domain in Chinese Wikipedia

- **Statistics**
  - Training & Validation & Testing:
    - 3480 instances on 4 predefined relations from (Fan et al., 2017)
  - Unlabeled:
    - 3161 entity pairs which share joint occurrence in the sentences

<table>
<thead>
<tr>
<th>Predefined relations</th>
<th>Directing</th>
<th>Singing</th>
<th>Starring</th>
<th>Spouse</th>
</tr>
</thead>
<tbody>
<tr>
<td># Instances</td>
<td>633</td>
<td>648</td>
<td>1609</td>
<td>590</td>
</tr>
</tbody>
</table>
Evaluation of Relation Classification

- Comparative study
  - We compare our method to CNN-based and RNN-based models, and experiment with different feature sets to verify their significance.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Feature set</th>
<th>F1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>logistic regression/SVM</td>
<td>entity pairs (add)</td>
<td>77.3/ 77.4</td>
</tr>
<tr>
<td></td>
<td>entity pairs (sub)</td>
<td>75.9/ 80.8</td>
</tr>
<tr>
<td></td>
<td>entity pairs (concat)</td>
<td>89.0/ 87.5</td>
</tr>
<tr>
<td></td>
<td>syntactic units, entity pairs (concat)</td>
<td>84.9/ 82.5</td>
</tr>
<tr>
<td></td>
<td>context words, entity pairs (concat)</td>
<td>87.6/ 86.6</td>
</tr>
<tr>
<td></td>
<td>syntactic units, context words</td>
<td>89.2/ 87.8</td>
</tr>
<tr>
<td></td>
<td>syntactic units, context words, entity pairs (concat)</td>
<td>89.9/ 88.0</td>
</tr>
<tr>
<td>Shwartz et al. (Shwartz et al., 2016)</td>
<td>shortest dependency path, entity pairs</td>
<td>65.3</td>
</tr>
<tr>
<td>Zeng et al. (Zeng et al., 2014)</td>
<td>context words, entity pairs</td>
<td>81.5</td>
</tr>
<tr>
<td>RNN+E</td>
<td>syntactic units, entity pairs (concat)</td>
<td>66.8</td>
</tr>
<tr>
<td>CNN+E</td>
<td>context words, entity pairs (concat)</td>
<td>91.4</td>
</tr>
<tr>
<td>Full implementation</td>
<td>syntactic units, context words, entity pairs (concat)</td>
<td><strong>92.2</strong></td>
</tr>
</tbody>
</table>
Evaluation of Relation Discovery

• **Pairwise experiment**
  
  - We manually construct a testing set by sampling pairs of instances $(x_i, x_j)$ from unlabeled data where $x = (e_1, e_2)$.

  \[
  \text{Precision} = \frac{|\{(x_i, x_j) \in D | v_{i,j} = 1 \land v_{i,j} = 1\}|}{|\{(x_i, x_j) \in D | v_{i,j} = 1\}|} \\
  \text{Recall} = \frac{|\{(x_i, x_j) \in D | v_{i,j} = 1 \land v_{i,j} = 1\}|}{|\{(x_i, x_j) \in D | v_{i,j} = 1\}|}
  \]

  - $v_{i,j} \in \{1,0\}$ for the ground truth, $v_{i,j} \in \{1,0\}$ for the clustering result

<table>
<thead>
<tr>
<th>Algorithm</th>
<th># Instances</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
<th>F1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit ssCRP</td>
<td>3161</td>
<td>31.0</td>
<td>35.7</td>
<td>33.2</td>
</tr>
<tr>
<td>Exploratory EM-based Naive Bayes</td>
<td>3161</td>
<td>70.7</td>
<td>40.2</td>
<td>52.8</td>
</tr>
<tr>
<td>Exploratory seeded KMeans</td>
<td>3161</td>
<td>80.5</td>
<td>53.0</td>
<td>63.9</td>
</tr>
<tr>
<td>ssCRP w/o tables</td>
<td>593</td>
<td>66.6</td>
<td>60.4</td>
<td>63.3</td>
</tr>
<tr>
<td>ssCRP w/o prediction</td>
<td>903</td>
<td>83.7</td>
<td>61.0</td>
<td>70.6</td>
</tr>
<tr>
<td>Exp ssCRP</td>
<td>3161</td>
<td>77.9</td>
<td>66.7</td>
<td>71.9</td>
</tr>
<tr>
<td>Logistic ssCRP</td>
<td>3161</td>
<td>81.4</td>
<td>66.9</td>
<td>73.0</td>
</tr>
<tr>
<td>Full implementation of ssCRP</td>
<td>3048</td>
<td><strong>83.1</strong></td>
<td><strong>68.4</strong></td>
<td><strong>75.0</strong></td>
</tr>
</tbody>
</table>
Evaluation of Relation Discovery

• **Newly discovered relations**
  - 6 new relations are generated, covering 96.4% unlabeled data

<table>
<thead>
<tr>
<th>Relation name</th>
<th># Instances</th>
<th>Relation name</th>
<th># Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group members</td>
<td>1328</td>
<td>Belong to the country</td>
<td>956</td>
</tr>
<tr>
<td>Family members</td>
<td>355</td>
<td>Series works</td>
<td>247</td>
</tr>
<tr>
<td>Employed by</td>
<td>144</td>
<td>Produced by</td>
<td>18</td>
</tr>
</tbody>
</table>

• **Top-k precision**
  - We heuristically choose $k = 0.4$ because the precision drops relatively faster when $k$ is larger than this setting.
Outline

• Introduction
• Related Work
• Proposed Approach
• Experiments
• Conclusion
Conclusion

• Exploratory relation classification
  – Problem: assign labels for unlabeled entity pairs to both pre-defined and unknown relations
  – Iterative process:
    • an integrated base neural network for relation classification
    • a similarity-based clustering algorithm ssCRP to generate new relations
    • constrained relation prediction process to populate new relations
  – Experiments: on Chinese Wikipedia entertainment domain, with base neural network achieving 0.92 F1-score, and 6 new relations generated with 0.75 F1-score.
Thanks!