Chains of Reasoning over Entities, Relations, and Text using RNNs

Authors: Rajarshi Das, Arvind Neelakantan, David Belanger, Andrew McCallum
Presented by: Assma Boughoula
Motivation

- Knowledge Base completion through relation inference
- Infer probability of relation “Lives in” between entities “Melinda” and “Seattle”
- Given the other paths between them in the knowledge graph
Previous work: Compositional VSM for KB Completion (2015)

Previously:
1. No reasoning about entities in path, just relations
2. Reasoning from single path
3. Train a separate model for each relation-type

This work:
1. Jointly reason about relation-types, entities and entity-types
2. Multiple paths
3. Single RNN that can predict all relation types
Per-relation Model: Path-RNN

- Train a separate RNN for each relation type (CountryOfHeadquarters)
- Only relation vectors are taken into account
Single-Model

Target Relation: “country of HQ”
Start: “Microsoft”
Target: “USA”

Path($e_s \rightarrow e_t$): $\pi = \{e_s, r_1, e_1, ..., r_k, e_t\} \in S$
Single-Model

RNN hidden state at step $t$ in the path

Now

$$h_t = f(W_{hh}h_{t-1} + W_{ih}y_{rt}).$$

No dependency on target relation $r$ here!

Previous

$$h_t = f(W_{hh}^r h_{t-1} + W_{ih}^r y_{rt}).$$

$f = \text{sigmoid function}$
Single-Model: incorporating entities

1. Learn Entity Vector Representation

2. Get annotated entity types from FreeBase:
   “Melinda Gates”
   CEO
   Duke University Alumni
   Philanthropist
   American Citizen

\[ h_t = f(W_{hh}h_{t-1} + W_{ih}y_{rt} + W_{eh}y_{et}) \]
So far ...
Single-Model: score pooling

Top K

\[ P(r|e_s, e_t) = \sigma \left( \frac{1}{k} \sum_j s_j \right), \forall j \in \mathcal{K} \]

Assigns 0 weight to some paths

Average

\[ P(r|e_s, e_t) = \sigma \left( \frac{1}{N} \sum_{i=1}^{N} s_i \right) \]

Each path gets same share of gradient regardless of whether it’s more/less important

LogSumExp

\[ P(r|e_1, e_2) = \sigma \left( \log \left( \sum_i \exp(s_i) \right) \right) \]

Each path gets share of gradient proportional to its score since:

\[ \frac{\partial \text{LSE}}{\partial s_i} = \frac{\exp(s_i)}{\sum_i \exp(s_i)} \]
Single-Model: score pooling

Top K

$$P(r|e_s, e_t) = \sigma\left(\frac{1}{k} \sum_{j} s_j\right), \forall j \in \mathcal{K}$$

Assigns 0 weight to some paths

Average

$$P(r|e_s, e_t) = \sigma\left(\frac{1}{N} \sum_{i=1}^{N} s_i\right)$$

Each path gets same share of gradient regardless of whether it’s more/less important

LogSumExp

$$P(r|e_1, e_2) = \sigma\left(\log\left(\sum_{i} \exp(s_i)\right)\right)$$

Each path gets share of gradient proportional to its score since:

$$\frac{\partial \text{LSE}}{\partial s_i} = \frac{\exp(s_i)}{\sum_i \exp(s_i)}$$
Experiments Setup

Dataset:

- Triples: \((e_s, r, e_t)\)

- Set of paths \(S\) connecting \((e_s, e_t)\) in the knowledge graph

Data is from FreeBase (KB) and ClueWeb(Text)

<table>
<thead>
<tr>
<th>Stats</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td># Freebase relation types</td>
<td>27,791</td>
</tr>
<tr>
<td># textual relation types</td>
<td>23,599</td>
</tr>
<tr>
<td># query relation types</td>
<td>46</td>
</tr>
<tr>
<td># entity pairs</td>
<td>3.22M</td>
</tr>
<tr>
<td># unique entity types</td>
<td>2218</td>
</tr>
<tr>
<td>Avg. path length</td>
<td>4.7</td>
</tr>
<tr>
<td>Max path length</td>
<td>7</td>
</tr>
<tr>
<td>Total # paths</td>
<td>191M</td>
</tr>
</tbody>
</table>

Table 2: Statistics of the dataset.
Experiments: Effect of Pooling Techniques

- LogSumExp is best
  - Important to include all paths

- Average is worst
  - Important to weigh path scores according to their values
## Experiments Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Performance (%MAP)</th>
<th>Pooling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Model</td>
<td>68.77</td>
<td>Max</td>
</tr>
<tr>
<td>Single-Model</td>
<td>55.80</td>
<td>Avg.</td>
</tr>
<tr>
<td>Single-Model</td>
<td>68.20</td>
<td>Top(k)</td>
</tr>
<tr>
<td>Single-Model</td>
<td><strong>70.11</strong></td>
<td>LogSumExp</td>
</tr>
<tr>
<td>PRA</td>
<td>64.43</td>
<td>n/a</td>
</tr>
<tr>
<td>PRA + Bigram</td>
<td>64.93</td>
<td>n/a</td>
</tr>
<tr>
<td>Path-RNN</td>
<td>65.23</td>
<td>Max</td>
</tr>
<tr>
<td>Path-RNN</td>
<td>68.43</td>
<td>LogSumExp</td>
</tr>
<tr>
<td>Single-Model</td>
<td><strong>70.11</strong></td>
<td>LogSumExp</td>
</tr>
<tr>
<td>PRA + Types</td>
<td>64.18</td>
<td>n/a</td>
</tr>
<tr>
<td>Single-Model</td>
<td>70.11</td>
<td>LogSumExp</td>
</tr>
<tr>
<td>Single-Model + Entity</td>
<td>71.74</td>
<td>LogSumExp</td>
</tr>
<tr>
<td>Single-Model + Types</td>
<td><strong>73.26</strong></td>
<td>LogSumExp</td>
</tr>
<tr>
<td>Single-Model + Entity + Types</td>
<td>72.22</td>
<td>LogSumExp</td>
</tr>
</tbody>
</table>

- **Effect of Pooling**
- **Other Multi-Hop Algorithms**
- **Effect Including Entities**
Take-aways:

- Complete Knowledge Graph by inferring relations between entities using existing paths
- Single-Model: trains a single RNN to handle multiple relation types
- Incorporating Entity vectors improves results