Traversing Knowledge Graphs in Vector Space

Kelvin Guu, John Miller, Percy Liang (2015)
Presented by Ben Striner 10/17/2017
Contents

• What is the dataset?
• What is the task?
• What is their model?
• How does it perform?
Dataset
Knowledge Graph

• Entities are nodes and relationships are labeled edges
• Tuples (s,r,t): (tad_lincoln, parent, abe_lincoln)
Dataset

- WordNet and Freebase
  - Dataset of entities and relationships
  - Some edges withheld for testing

<table>
<thead>
<tr>
<th></th>
<th>WordNet</th>
<th>Freebase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relations</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>Entities</td>
<td>38,696</td>
<td>75,043</td>
</tr>
<tr>
<td>Base</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Train</td>
<td>112,581</td>
<td>316,232</td>
</tr>
<tr>
<td>Test</td>
<td>10,544</td>
<td>23,733</td>
</tr>
<tr>
<td>Paths</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Train</td>
<td>2,129,539</td>
<td>6,266,058</td>
</tr>
<tr>
<td>Test</td>
<td>46,577</td>
<td>109,557</td>
</tr>
</tbody>
</table>

Table 1: WordNet and Freebase statistics for base and path query datasets.
Tasks
Tasks

• Path Query
  • Given start point and series of relationships predict target
  • Tad_lincoln/parent/location = DC
  • Using multi-step paths generated by walking base dataset

• Knowledge Base Completion
  • Predict whether an edge exists or not
  • Formulated as single-edge path query
  • Using base dataset
Path Query

• A query (q) consists of an “anchor entity” (s) and a path (p)
• A path is a series of relationships \( p = (r_1, \ldots, r_k) \)
• The answer is the “denotation” \([q]\)
• Defined recursively

\[
[s] \overset{\text{def}}{=} \{s\}, \quad \quad (1)
\]
\[
[q/r] \overset{\text{def}}{=} \{t : \exists s \in [q], (s, r, t) \in \mathcal{G}\}. \quad (2)
\]
Path Query Evaluation

• C: Candidate answers “type match”
  • Participate in final relationship at least once
  • For example, all entities that are the target of a “located at” relationship would identify most valid locations

• N(q): Incorrect candidate answers

\[
C(s/r_1/\cdots/r_k) \overset{\text{def}}{=} \{ t \mid \exists e, (e, r_k, t) \in G \} \quad (3)
\]

\[
N(q) \overset{\text{def}}{=} C(q) \setminus [q]. \quad (4)
\]
Mean Quantile

• Evaluate fraction of incorrect answers are ranked after correct answer

\[
\frac{|\{t' \in \mathcal{N}(q) : \text{score}(q, t') < \text{score}(q, t)\}|}{|\mathcal{N}(q)|}
\]  

(13)
Knowledge Base Completion Evaluation

• Evaluate accuracy versus negative samples
• For comparison to previous work (Socher)
Models
Modelling Traversal and Membership

• Traversal operator determines the set that can be reached from \( xs \)
• Membership operator determines if \( xt \) is in the set reached from \( xs \)
• Defined recursively

\[
T_{r_i}(v) = v^\top W_{r_i} \\
M(v, x_t) = v^\top x_t
\]

\[
score(s/r, t) = M(T_r(x_s), x_t)
\]

\[
[s]_V \overset{\text{def}}{=} x_s, \quad [q/r]_V \overset{\text{def}}{=} T_r([q]_V).
\]

\[
score(q, t) = M([q]_V, [t]_V).
\]
Objective Function

• Use Max-Margin loss against incorrect answers that “type match”
• Use paths of different lengths

\[ J(\Theta) = \sum_{i=1}^{N} \sum_{t' \in N(q_i)} \left[ 1 - \text{margin}(q_i, t_i, t') \right]_+, \]

\[ \text{margin}(q, t, t') = \text{score}(q, t) - \text{score}(q, t'), \]

\[ \Theta = \{M\} \cup \{T_r : r \in R\} \cup \{x_e \in \mathbb{R}^d : e \in E\}. \]
Experimental models

• Model Traversal and Membership functions three ways
  • Bilinear
  • TransE
  • Bilinear-Diag

• Also use NTN (for some experiments)
Bilinear Model (Nickel et al., 2011)

• Traditional queries can be answered by chaining matrix multiplication
  • Entities are one-hot indicator vectors
  • Relationships are adjacency matrices
  • $xA =$ all nodes connected to node $x$ by adjacency $A$ (vector)
  • $y^Tx = 1$ if $x$ is in set $y$ else 0 (scalar)

• Build a similar model with continuous learned representations

\[
\text{score}(s/r, t) = x_s^\top W_r x_t. \tag{5}
\]

\[
\text{score}(q, t) = x_s^\top W_{r_1} \cdots W_{r_k} x_t. \tag{6}
\]
TransE (Bordes et al., 2013)

• Every entity and relationship embedded as a vector
• Traversal is addition, Membership is L2 distance

\[
\text{score}(s/r, t) = -\|x_s + w_r - x_t\|^2 \tag{11}
\]

\[
\text{M}(v, x_t) = -\|v - x_t\|^2 \tag{12}
\]

\[
\text{T}_r(x_s) = x_s + w_r
\]

\[
\text{score}(q, t) = -\|x_s + w_{r_1} + \cdots + w_{r_k} - x_t\|^2.
\]
Bilinear-Diag Model (Yang et al., 2015)

• Same as Bilinear model but matrices are diagonal
  • Can no longer interpret weight matrices as adjacency matrices
  • Same number of parameters as TransE

Experiments
Results

• Models trained on single edges perform poorly even when all edges have been seen during training

<table>
<thead>
<tr>
<th>Path query task</th>
<th>Bilinear</th>
<th>Bilinear-Diag</th>
<th>TransE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SINGLE</td>
<td>COMP (%)red</td>
<td>SINGLE</td>
</tr>
<tr>
<td>WordNet MQ</td>
<td>84.7</td>
<td>89.4 30.7</td>
<td>59.7</td>
</tr>
<tr>
<td>WordNet H@10</td>
<td>43.6</td>
<td>54.3 19.0</td>
<td>7.9</td>
</tr>
<tr>
<td>Freebase MQ</td>
<td>58.0</td>
<td>83.5 60.7</td>
<td>57.9</td>
</tr>
<tr>
<td>Freebase H@10</td>
<td>25.9</td>
<td>42.1 21.9</td>
<td>23.1</td>
</tr>
<tr>
<td>KBC task MQ</td>
<td>76.1</td>
<td>82.0 24.7</td>
<td>76.5</td>
</tr>
<tr>
<td>KBC task H@10</td>
<td>19.2</td>
<td>27.3 10.0</td>
<td>12.9</td>
</tr>
<tr>
<td>Freebase MQ</td>
<td>85.3</td>
<td>91.0 38.8</td>
<td>84.6</td>
</tr>
<tr>
<td>Freebase H@10</td>
<td>70.2</td>
<td>76.4 20.8</td>
<td>63.2</td>
</tr>
</tbody>
</table>

Table 2: Path query answering and knowledge base completion. We compare the performance of single-edge training (SINGLE) vs compositional training (COMP). MQ: mean quantile, H@10: hits at 10, %red: percentage reduction in error.
Implementation Details

• For each query, sample 10 negative entities
• Entity vectors constrained to unit ball
• Gradient clipping, Minibatch of 300, AdaGrad
• Train on length 1 until convergence, then train on full
• Explicitly parameterized inverse relationships (parent = child\(^{-1}\))
  • Exclude trivial queries where exact inverse was in training
• Experiment with parameterizing entities with word embeddings
Generating Paths

• Generate queries by random walks
  • Uniform sample of path length and start point
  • Uniform sample of available relationships
  • Uniform sample of next node given that relationship

• Large amounts of training data generated
Deduction vs Induction

- Deduction
  - Entities and relations seen during training, just not the exact query

- Induction
  - Required edge not seen during training

Table 3: Deduction and induction. We compare mean quantile performance of single-edge training (SINGLE) vs compositional training (COMP). Length 1 queries are excluded.
Pretrained word vectors

- Using pretrained word vectors can improve performance

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>WordNet EV</th>
<th>WV</th>
<th>Freebase EV</th>
<th>WV</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTN</td>
<td>70.6</td>
<td>86.2</td>
<td>87.2</td>
<td>90.0</td>
</tr>
<tr>
<td>Bilinear COMP</td>
<td>77.6</td>
<td>87.6</td>
<td>86.1</td>
<td>89.4</td>
</tr>
<tr>
<td>TransE COMP</td>
<td>80.3</td>
<td>84.9</td>
<td>87.6</td>
<td>89.6</td>
</tr>
</tbody>
</table>

Table 5: Model performance in terms of accuracy. EV: entity vectors are separate (initialized randomly); WV: entity vectors are average of word vectors (initialized with pretrained word vectors).
Composition improves performance

• Although a perfect model trained on single steps should work on multiple steps, it doesn’t
• Cascading errors cause problems but composition helps
Cascading Errors

- Models suffer from cascading errors
- Trained on up to 5 steps
Questions/Discussion