An Interpretable Knowledge Transfer Model for Knowledge Base Completion

Qizhe Xie, Xuezhe Ma, Zihang Dai, Eduard Hovy

Carnegie Mellon University
Language Technologies Institute

August 2, 2017
Outline

Introduction
  Task
  Motivation

Model

Experiments
  Main Results
  Performance on Rare Relations
  Interpretability
  Analysis on Sparseness
Introduction
  Task
  Motivation

Model

Experiments
  Main Results
  Performance on Rare Relations
  Interpretability
  Analysis on Sparseness
Task: Knowledge base completion (KBC)

- Recover missing facts in knowledge bases
  - Given lots of triples such as
    
    
    (Leonardo DiCaprio, won award, Oscar)
  - Predict missing facts (Leonardo DiCaprio, Profession, ?)

- Embedding-based approaches
Data Sparsity Issue

Figure 1: Frequencies of relations are subject to Zipf’s law.
Problems Our Model Tackle

- **Data-sparsity: Transfer learning**
  - On WN18, the rarer the relation is, the greater the improvements are

- **Interpretability: $\ell_0$-regularized representation**
  - Reverse relations, undirected relations and similar relations are identified by the sparse representation

- **Model size: Compression**
  - On FB15k, the number of parameters can be reduced to 1/90 of the original model
Outline

Introduction
  Task
  Motivation

Model

Experiments
  Main Results
  Performance on Rare Relations
  Interpretability
  Analysis on Sparseness
Notation and Previous Models

- Data: Triples \((h, r, t)\)
  - Training data: \((h = Leonardo DiCaprio, r = \text{won award}, t = Oscar)\)
  - Test data: \((h = Leonardo DiCaprio, r = \text{Profession}, t = ?)\)

- Energy function \(f_r(h, t)\) of triples \((h, r, t)\)
  - Minimize the energy of true triples and maximize the energy of false triples
    - TransE [Bordes et al., 2013]:
      \[
      f_r(h, t) = \|h + r - t\|_\ell
      \]
      Parameters: entity embeddings \(h, t\), relation embeddings \(r\)
    - STransE [Nguyen et al., 2016]:
      \[
      f_r(h, t) = \|W_{r,1}h + r - W_{r,2}t\|_\ell
      \]
      Parameters: relation-specific projection matrices \(W_{r,1}, W_{r,2}\) and embeddings
  - All parameters are trained by SGD
STransE: Parametrizing Each Relation Separately

- Prone to the data sparsity problem
Sharing Parameters through Common Concepts

- Relation-concept mapping example with attention weights:

![Diagram showing relation-concept mapping with attention weights]

- Parametrize concepts instead of relations
- Relation matrices are weighted averages of concept matrices with attention weights

\[ W_{r_{1,1}} = 0.2D_1 + 0.8D_2 \]
Sharing Parameters through Common Concepts

- Suppose a ground-truth mapping is given, then
  - Transfer learning can be done effectively through parameter sharing
  - We can interpret similar relations
  - All parameters are trainable by SGD

- Concepts need to be learned end-to-end
- How do we obtain the mapping?
Dense Mapping

- Dense attention: Construct a dense bipartite graph and train attention weights

![Diagram of a dense bipartite graph with relations and concepts]

- Problems:
  - Uninterpretable: In practice, even with $\ell_1$ regularization, we get a distributed weights $W_{r_i,1} = 0.2D_1 + 0.52D_2 + 0.1D_3 + 0.15D_4 + 0.03D_5$
  - Inefficient: Computation involves all concept matrices
  - Unnecessary: Intuitively, each relation can be composed of at most $K$ concepts
Sparse Mapping

- Problem: Not differentiable
- An approximate approach:
  - Given current embeddings, a correct mapping should minimize the loss function
  - For each relation, assign a single concept to the relation and compute the loss
  - Greedily choose the top $K$ concepts that minimize the loss
Block Iterative Optimization

- Randomly initialize mappings and concepts.
- Repeat
  - Optimize embeddings and attention weights with SGD
  - Reassign mappings
A Better Sampling Approach: Domain sampling

- Loss function involves negative sampling
- Sample from domain-specific entities with an adaptive probability
- E.g., negative sample of \((Steve Jobs, \text{was born in}, US)\):  
  - Uniform negative sample: \((Steve Jobs, \text{was born in}, CMU)\)  
  - Domain negative sample: \((Steve Jobs, \text{was born in}, China)\)
Introduction
  Task
  Motivation

Model

Experiments
  Main Results
  Performance on Rare Relations
  Interpretability
  Analysis on Sparseness
## Table 1: Link prediction results on two datasets. Hits@10 is the top-10 accuracy. Higher Hits@10 or lower Mean Rank indicates better performance.
Performance on Rare Relations

Figure 2: Average Hits@10 on WN18 relations
Performance on Rare Relations

Figure 3: Average Hits@10 on relations of different frequencies

(a) WN18

(b) FB15k

Figure 3: Average Hits@10 on relations of different frequencies
Interpretability: How Is Knowledge Shared?

Each relation’s head and tail have their own concepts.

(a) WN18  
(b) FB15k

Figure 4: Heatmap visualization of attention weights on WN18 and FB15k.
Interpretability: How Is Knowledge Shared?

- Each relation’s head and tail have their own concepts.
- Interpretation:
  - Reverse relations: hyponym and hypernym; award winning work and won award for.

Figure 5: Heatmap visualization of attention weights on WN18 and FB15k.
Interpretability: How Is Knowledge Shared?

- Each relation’s head and tail have their own concepts.
- Interpretation:
  - Reverse relations: hyponym and hypernym; `award_winning_work` and `won_award_for`.
  - Undirected relations: `spouse`; `similar_to`.

(a) WN18  (b) FB15k
Interpretability: How Is Knowledge Shared?

- Each relation’s head and tail have their own concepts.
- Interpretation:
  - Reverse relations: hyponym and hypernym; award_winner(work) and won_award_for.
  - Undirected relations: spouse; similar_to.
  - Similar relations: was_anominated_for and won_award_for; instance_hypernym and hypernym.

(a) WN18
(b) FB15k
Interpretability of $\ell_1$ regularized dense mapping

(a) WN18  
(b) FB15k

Figure 8: Heatmap visualization of $\ell_1$ regularized dense mapping

- The mapping cannot be sparse without performance loss.
A Byproduct of Parameter Sharing: Model Compression

Figure 9: Performance with different number of concepts

(a) FB15k

(b) WN18

- On FB15k, the model can be compressed by nearly 90 times.
Analysis on Sparseness

**Does sparseness hurt performance?**

<table>
<thead>
<tr>
<th>Method</th>
<th>WN18</th>
<th>FB15k</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MR</td>
<td>H10</td>
</tr>
<tr>
<td>Dense</td>
<td>199</td>
<td>94.0</td>
</tr>
<tr>
<td>Dense + $\ell_1$</td>
<td>228</td>
<td>94.2</td>
</tr>
<tr>
<td>Sparse</td>
<td>207</td>
<td>94.1</td>
</tr>
<tr>
<td></td>
<td>MR</td>
<td>H10</td>
</tr>
<tr>
<td>Dense</td>
<td>69</td>
<td>79.4</td>
</tr>
<tr>
<td>Dense + $\ell_1$</td>
<td>131</td>
<td>78.9</td>
</tr>
<tr>
<td>Sparse</td>
<td>67</td>
<td>79.6</td>
</tr>
</tbody>
</table>

*Table 2:* Performance of model with dense graph or sparse graph with only 15 or 22 concepts. The time gap is more significant when we use more concepts.

**How does our approach compare to sparse encoding methods?**

<table>
<thead>
<tr>
<th>Method</th>
<th>WN18</th>
<th>FB15k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pretrain + Sparse Encoding [Faruqui et al., 2015]</td>
<td>211</td>
<td>86.6</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>205</strong></td>
<td>94.2</td>
</tr>
</tbody>
</table>

*Table 3:* Different methods to obtain sparse representations
Conclusion

- Propose a knowledge embedding model which can discover shared hidden concepts
- Perform transfer learning through parameter sharing
- Design a learning algorithm to induce the interpretable sparse representation
- Outperform baselines on two benchmark datasets for the knowledge base completion task

