Forcing Neural Link Predictors to Play by the Rules

Sebastian Riedel
Goal

Calculate Truth Scores for Statements

If higher, more likely true

Learn from data and prior knowledge, use neural networks, graphical models etc.
in natural language, predicate logic, atoms, rules etc.
**Running Example**

$$S(\text{Enrique can speak Spanish}) = 0.9$$

<table>
<thead>
<tr>
<th>Prior Knowledge (Text)</th>
<th>Enrique works in Mexico City</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enrique lives in Mexico</td>
</tr>
</tbody>
</table>

| Training data (Text)   | Luis speaks Spanish         |
|                        | Luis lives in Mexico        |

<table>
<thead>
<tr>
<th>Training data (Structure)</th>
<th>livesIn(Ivan, Mexico)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>speaks(Ivan, Spanish)</td>
</tr>
</tbody>
</table>

When someone lives in Mexico, they will likely speak Spanish.
Compliance

\[ S(\text{I can accept tickets to NBA court-side events}) = 0.1 \]

You must not accept tickets to expensive sports events

NBA court-side tickets costs at least $1000
Fact Checking

\[
S(\text{US Unemployment is 42%}) = 0.1
\]

The official unemployment rate is the US is 4.4%

There are 420 billion potential working hours

only 240 billion working hours were actually recorded
DrugA interacts with DrugB

\[ S(\text{DrugA interacts with DrugB}) = 0.9 \]

DrugA interacts with Protein1

Protein1 has been shown to stimulate Protein2

DrugB increases the rate of Protein2 ...
Knowledge Graphs

\[ S(Andrew\text{McCallum} \text{livesIn} \text{Amherst}) = 0.9 \]
Part 1: Learning To Score

\[ S(\text{Enrique can speak Spanish}) = 0.9 \]

Enrique works in Mexico City

Enrique lives in Mexico

Luis can speak Spanish

Luis lives in Mexico

livesIn(Ivan, Mexico)

speaks(Ivan, Spanish)

When someone lives in Mexico, they will likely speak Spanish
Part 2: Injecting Prior Knowledge

\[ S(\text{Enrique can speak Spanish}) = 0.9 \]

Enrique works in Mexico City

Enrique lives in Mexico

Luis can speak Spanish

Luis lives in Mexico

livesIn(Ivan, Mexico)

speaks(Ivan, Spanish)

When someone lives in Mexico, they will likely speak Spanish
Part 1: Learning To Score

\[ S(\text{Enrique can speak Spanish}) = 0.9 \]

Enrique works in Mexico City
Enrique lives in Mexico
Luis can speak Spanish
Luis lives in Mexico
livesIn(Ivan, Mexico)
speaks(Ivan, Spanish)

When someone lives in Mexico, they will likely speak Spanish
S\left(\text{Enrique can speak Spanish}\right) = 0.9

Enrique works in Mexico City
Enrique lives in Mexico
Luis can speak Spanish
Luis lives in Mexico
livesIn(Ivan, Mexico)
speaks(Ivan, Spanish)
S(Enrique canSpeak Spanish) = 0.9

Enrique worksIn MexicoCity
Enrique livesIn Mexico
Luis canSpeak Spanish
Luis livesIn Mexico
livesIn(Ivan, Mexico)
speaks(Ivan, Spanish)
S(Enrique canSpeak Spanish) = 0.9
DistMult (Yang et. al, 2015)

\[ S(\text{Enrique canSpeak Spanish}) = 0.9 \]
Symmetric

\[ S(\text{Spanish } \text{canSpeak} \text{ Enrique}) = 0.9 \]
Complex (Trouillon et al, 2016)

$S(\text{Enrique canSpeak Spanish}) = 0.9$
TransE (Bordes et al. 2013)

\[ S(\text{Enrique canSpeak Spanish}) = 0.9 \]
Rescal (Nickel et. al 2013)

\[ S(\text{Enrique canSpeak Spanish}) = 0.9 \]
ERMLP (Dong et al. 2014)

$S(\text{Enrique canSpeak Spanish}) = 0.9$
ConvE (Dettmers et al. 2017)

\[ S(\text{Enrique canSpeak Spanish}) = 0.9 \]
Training Objective

- Terms in training objective for observed facts

  $\sum_{\text{subject, predicate, object}} \text{loss} \left( s \left( \text{subject, predicate, object} \right) \right)$

  - High
  - Low

- Plus some term with negative sampling

- Optimised by Stochastic Gradient Descent
Learning Visualised

Random Entity Embeddings & Training and Test Edges

- Training Edge
- Test Edge

Graph showing:
- `language` nodes
- `speaks` relationship
- `livesIn` relationship
Learning Visualised

Predictions of a randomly initialised Model

- **speaks**
- **livesIn**
- **language**

- **Training Edge**
- **Test Edge**
- **False Negative**
- **False Positive**
- **Match**
Learning Visualised

Predictions after training

- speaks
- language
- livesIn

- Training Edge
- Test Edge
- False Negative
- False Positive
- Match
Part 2: Injecting Prior Knowledge

\[ S(\text{Enrique can speak Spanish}) = 0.9 \]

Enrique works in Mexico City

Enrique lives in Mexico

Luis can speak Spanish

Luis lives in Mexico

\text{livesIn(Ivan, Mexico)}

\text{speaks(Ivan, Spanish)}

When someone lives in Mexico, they will likely speak Spanish
With sufficient training data

**Learning Visualised**

- *speaks*
- *livesIn*
- *language*

### Edges
- **Training Edge**
- **Test Edge**
- **False Negative**
- **False Positive**
- **Match**
Learning Visualised

With limited training data

language

livesIn

Training Edge
Test Edge
False Negative
False Positive
Match
Limited training Data

(Obviously) Leads to errors

language

livesIn

Training Edge
Test Edge
False Negative
False Positive
Match
Errors

May violate our **Prior Knowledge**

When A **lives in** C and C’s **language** is L then A likely **speaks** L
Solution

Add a loss term that punishes this specific violation

\[ \ldots + \text{loss}(v_A, v_B, v_C, v_{\text{lives}}, v_{\text{speaks}}, v_{\text{lang}}) \]
Re-training

with the specific loss term added

language

livesIn

Training Edge
Test Edge
False Negative
False Positive
Match
Re-training

fixes the specific error

- Training Edge
- Test Edge
- False Negative
- False Positive
- Match

language

livesIn
Re-training

but not all

livesIn

language
Loss

So minimise the loss of the worst case

\[ \text{expensive} \]

\[ \ldots + \operatorname{Max}_{A,B,C} \text{loss}(v_A, v_B, v_C, v_{\text{lives}}, v_{\text{speaks}}, v_{\text{lang}}) \]
Avoid combinatorial optimisation by maximising over *embeddings*

\[
\text{... + Max}_{v_A, v_B, v_C} \text{ loss}(v_A, v_B, v_C, v_{lives}, v_{speaks}, v_{lang})
\]

cheaper
Full Loss

Terms in training objective for observed facts

\[ \sum \text{loss} \left( s \left( \text{subject, predicate, object} \right) \right) \]

subject, predicate, object in training data

+ \text{Max} \text{loss} \left( v_A, v_B, v_C, v_{\text{lives}}, v_{\text{speaks}}, v_{\text{lang}} \right)

Minimised in two player game
Player 1: Link Predictor

Estimates entity and relation embeddings to fix the graph (gradient descent)
Player 1

Predictions after training
Player 2: Adversary

Synthesises entity *embeddings* that *break the rules* (*gradient ascent*)
Player 2: Adversary

Synthesises entity *embeddings* that *break the rules* (*gradient ascent*)
Player 1: Link Predictor

Estimates entity and relation embeddings to fix the graph and violations
Player 2: Adversary

Estimates entity and relation embeddings to fix the graph and violations
Player 1: Link Predictor

Estimates entity and relation embeddings to **fix the graph and violations**
Closed Form Solutions

For some rules and models the max expression has a closed form solution

For DistMult, Complex, TransE:

\[
\text{Max} \quad \text{loss}(v_A, v_B, v_{rel1}, v_{rel2})
\]

\[
= \text{loss}_{\text{closed-form}}(v_{rel1}, v_{rel2})
\]

When A \(relation1\) B then A \(relation2\) B
Entailment in Vector Space

\[ \begin{bmatrix} 1 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \end{bmatrix} * \begin{bmatrix} 2 \\ 1 \end{bmatrix} = 3 \]

- Enrique
- worksIn
- Mexico

Where to put livesIn vector to be implied by worksIn
Entailment in Vector Space

\[
\begin{bmatrix}
2 \\
1 \\
\end{bmatrix}
\times
\begin{bmatrix}
1 \\
1 \\
\end{bmatrix} = 3
\]

- Enrique, Mexico
  - worksIn
- livesIn

Entailment: livesIn score is at least as high as worksIn
Entailment in Vector Space

\[
\begin{bmatrix}
2 \\
1
\end{bmatrix}
\times
\begin{bmatrix}
1 \\
1
\end{bmatrix} = 3
\]

- worksIn
- livesIn

When it’s the same vector

Enrique, Mexico
Entailment in Vector Space

When the first component is larger

\[ \begin{bmatrix} 1 \\ 2 \end{bmatrix} \times \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \end{bmatrix} \]

WorksIn

LivesIn

Enrique, Mexico
Entailment in Vector Space

When the first component is larger

\[
\begin{bmatrix} 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 3 \\ 1 \end{bmatrix} = 7
\]

- worksIn
- livesIn

Enrique, Mexico
Entailment in Vector Space

When the second component is larger

\[
\begin{bmatrix}
2 \\
1
\end{bmatrix} \times \begin{bmatrix}
1 \\
2
\end{bmatrix} = \begin{bmatrix}
3 \\
4
\end{bmatrix}
\]
Entailment in Vector Space

\[
\begin{bmatrix}
2 \\
1
\end{bmatrix}
\times
\begin{bmatrix}
x \geq 1 \\
y \geq 1
\end{bmatrix}
\geq 3
\]

- For any linear combination
Entailment in Vector Space

\[
\begin{bmatrix} a \\ b \end{bmatrix} \times \begin{bmatrix} x \geq 1 \\ y \geq 1 \end{bmatrix} \geq Z
\]

\[ \geq 0 \]

And any (non-negative) input entity pair

Compare Order Embeddings (Vendrov et al. 2016)
Results

<table>
<thead>
<tr>
<th>Model</th>
<th>FB122 Hits@3</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransE</td>
<td>59%</td>
</tr>
<tr>
<td>Kale-TransE (Guo 16)</td>
<td>62%</td>
</tr>
<tr>
<td>DistMult</td>
<td>67%</td>
</tr>
<tr>
<td>...with rules</td>
<td>71%</td>
</tr>
<tr>
<td>Complex</td>
<td>67%</td>
</tr>
<tr>
<td>...with rules</td>
<td>72%</td>
</tr>
</tbody>
</table>

- Applied to general link prediction problems
- Integrated rules such as:
  \[\text{nationalityOf}(P,N), \text{officialLang}(N,L) \Rightarrow \text{speaks}(P,L)\]
Conclusion

- Train Neural “Sentence Scorer” with limited data and prior knowledge
- Implemented via 2-player game
  - Player 1 learns to predict training facts, and follow rules
  - Player 2 creates tuples (in embedding space) that violate the rules
- Runtime “independent of domain size”
- Future work:
  - inject prior knowledge in natural language
  - extract explanations (see Sameer Singh’s talk)
Papers presented in this work


- Injecting Logical Background Knowledge into Embeddings for Relation Extraction, Rocktaschel, Tim, Singh, Sameer and Riedel, Enriqueastian, Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) 2015


- Adversarial Sets for Regularising Neural Link Predictors, Minervini, Pasquale, Demeester, Thomas, Rocktaschel, Tim, Riedel, Enriqueastian, Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI) 2017
Entailed Predicates need to live in the boxes of their premises