Probing a Semantic Dependency Parser for Translational Relation Embeddings

Riley Capshaw   Marco Kuhlmann   Eva Blomqvist

Department of Computer and Information Science
Linköping University, Sweden
Introduction
Experimental Setup
Results
Conclusions

Outline

1 Introduction
2 Experimental Setup
3 Results
4 Conclusions
Abstract

In order to assess whether Translational Relation Embedding models are compatible with the NLP task of Semantic Dependency Parsing, we present a series of probing experiments.

We show that there seems to be some compatibility, but that further work is needed to take advantage of it (i.e., such a model is not explicitly learned by the parser).

We hope that this can be used to improve the compatibility between components in pipelines for Knowledge Graph generation and completion.
Motivation

- Translational models provide explicit edge/label embeddings.
- Improve the interpretability of the parser’s label decisions.
- Improve compatibility of a parser with machine readers and other NLP/NLU pipelines (harmonize representations)
Translational Relation Embeddings (TransE)

Given facts as triples of the form $(s, p, o)$, represent the entities $s, o$ as vectors $h, t \in \mathbb{R}^n$.

Map $p$ to the translation vector $r$ such that

$$r \approx t - h$$


The results were in line with analysts’ expectations.

Figure: Example SDG #22007003 from the DM dataset.

- Encode shallow semantic phenomena between words.
- Formulated as a directed acyclic graph.
- Example triple: (in, ARG1, results)
Semantic Dependency Parsing

Semantic Dependency Parsers generally use deep neural networks in an encoder-decoder model:

- Encoder: 3-Layer of BiLSTM units
- Decoder: Biaffine classifiers

Figure: Dozat, Timothy, and Christopher D. Manning. “Simpler but more accurate semantic dependency parsing.” 2018.
Use unintrusive simple classifiers to “study” what a neural network learns:

- **Linear probes**: How easily can features at a given layer linearly separate the target classes?
- **Structural probes**: How well can the features at a given layer be structured in a particular way?
Probing Neural Networks: Structural probes

Given a pre-trained encoder (contextualizer) for a semantic dependency parser, we want to see how well a translational model would work as the parser’s decoder \textit{without} further training the parser.

That is, we want to see if we can \textit{explain} the parser’s predictions based on a linear restructuring of its encoder’s vector space.
For a given layer of a neural network:

- Take that layer’s output for words $i, j$ as the vectors $x_i, x_j$.
- Train a translational relation model to predict the predicted output relation for the samples given only $x_i$ and $x_j$ as input.
- Interpret the accuracy as a measure of the ability of the model to explain the parser’s predictions.

Note that structural probes tend to probe for a feature external to the dataset, such as syntactic trees in language models. We are looking for relational structuring in a parser’s predictions, which is arguably external.
To understand how well the structural probes fit the vector space, we use linear probes as a control to provide a *theoretical upper bound*.

For a given layer of a neural network:

- Combine that layer’s output for words $i$ and $j$ as the vector $\mathbf{x}$.
- Train a simple linear softmax classifier $\text{softmax}(\mathbf{Wx} + \mathbf{b})$ by minimizing cross-entropy to predict the predicted output label for words $i$ and $j$ given only their representations.
- Interpret the accuracy as a measure of the linear separability of the layer’s features.

---

1 Most probes predict the *true* output label, *not* the predicted one.
Parser and Data

- We used a pre-existing semantic dependency parser\(^2\).
- We trained the parser on the English DELPH-IN MRS (DM) portion of the 2014 and 2015 SemEval shared tasks on broad-coverage semantic dependency parsing.
- For every *predicted* dependency \(r\) between pairs of words \(h\) and \(t\), we record the respective \(m\)-dimensional representations \(h, t \in \mathbb{R}^m\) generated by each BiLSTM layer as \((h, r, t)\).
- Predictions are made for every sentence in both the training and testing sets.

Given the predicted training and testing sets, we trained the following three linear probes:

- \( W[t - h] + b \) (subtraction/translation)
- \( W[h + t] + b \) (addition)
- \( W[h; t] + b \) (concatenation)

And the following two structural probes:

- \( h + r_r - t \)
- \( Mh + r_r - Mt \)
Structural Probes

For the structural probes, we first define a scoring function:

\[ f_r(h, t) = \| h + r_r - t \|_2^2, \text{ or} \]

\[ f_r(h, t) = \| Mh + r_r - Mt \|_2^2 \]

Where \( M \) (if present) and \( r_r \) are learned parameters.

Then, given a margin \( \gamma \), we define our learning constraints:

1. \( f_r(h, t) \leq \gamma \) (relaxed translation)
2. \( f_{r'}(h, t) > \gamma \) for all \( r' \neq r \) (label separation)
3. \( \| t - h \|_2^2 > 2\gamma \) (enforce directionality)

where \( \| \cdot \|_2^2 \) is the squared \( \ell_2 \) norm.
Combining the constraints and scoring function, we get the following loss function:

\[
\mathcal{L} = \left[ f_r (h, t) - \gamma \right]_+ + \sum_{r' \neq r} \left[ \gamma - f_{r'} (h, t) \right]_+ + \left[ 2\gamma - \|r\|^2 \right]_+ ,
\]

where \([\cdot]_+\) is equivalent to \(\max(0, \cdot)\).

We also recalculated scores where a prediction is considered correct only if the margin constraint \((f_r (h, t) \leq \gamma)\) was satisfied.
## Results: Layer-by-layer

<table>
<thead>
<tr>
<th>Category</th>
<th>ID</th>
<th>Probe</th>
<th>Layer 0</th>
<th>Layer 1</th>
<th>Layer 2</th>
<th>Layer 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>$L_1$</td>
<td>$W[\mathbf{h} + \mathbf{t}] + b$</td>
<td>66.73</td>
<td>81.34</td>
<td>88.85</td>
<td>91.01</td>
</tr>
<tr>
<td></td>
<td>$L_2$</td>
<td>$W[\mathbf{t} - \mathbf{h}] + b$</td>
<td>67.82</td>
<td>85.18</td>
<td>94.07</td>
<td>95.89</td>
</tr>
<tr>
<td></td>
<td>$L_3$</td>
<td>$W[\mathbf{h}; \mathbf{t}] + b$</td>
<td>72.69</td>
<td>89.06</td>
<td>96.54</td>
<td>97.52</td>
</tr>
<tr>
<td>Structural</td>
<td>$S_1$</td>
<td>$\mathbf{h} + r_r - \mathbf{t}$</td>
<td>38.87</td>
<td>48.44</td>
<td>56.73</td>
<td>60.09</td>
</tr>
<tr>
<td></td>
<td>$S_2$</td>
<td>$\mathbf{Mh} + r_r - \mathbf{Mt}$</td>
<td>60.37</td>
<td>76.88</td>
<td>86.96</td>
<td>90.76</td>
</tr>
<tr>
<td>Constrained</td>
<td>$C_1$</td>
<td>$\mathbf{h} + r_r - \mathbf{t}$</td>
<td>0.82</td>
<td>0.89</td>
<td>0.84</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>$C_2$</td>
<td>$\mathbf{Mh} + r_r - \mathbf{Mt}$</td>
<td>35.62</td>
<td>56.37</td>
<td>65.10</td>
<td>73.57</td>
</tr>
</tbody>
</table>

Scores for all experiments in terms of recall as a function of layer. In general, as the depth increases, the score increases.
## Results: Final Layer

<table>
<thead>
<tr>
<th>Category</th>
<th>ID</th>
<th>Probe</th>
<th>Layer 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>$L_1$</td>
<td>$W[h + t] + b$</td>
<td>91.01</td>
</tr>
<tr>
<td></td>
<td>$L_2$</td>
<td>$W[t - h] + b$</td>
<td>95.89</td>
</tr>
<tr>
<td></td>
<td>$L_3$</td>
<td>$W[h; t] + b$</td>
<td>97.52</td>
</tr>
<tr>
<td>Structural</td>
<td>$S_1$</td>
<td>$h + r_r - t$</td>
<td>60.09</td>
</tr>
<tr>
<td></td>
<td>$S_2$</td>
<td>$Mh + r_r - Mt$</td>
<td>90.76</td>
</tr>
<tr>
<td>Constrained</td>
<td>$C_1$</td>
<td>$h + r_r - t$</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>$C_2$</td>
<td>$Mh + r_r - Mt$</td>
<td>73.57</td>
</tr>
</tbody>
</table>

Scores for all probing experiments in terms of recall.
Results: Linear Probes

<table>
<thead>
<tr>
<th>Category</th>
<th>ID</th>
<th>Probe</th>
<th>Layer 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>$L_1$</td>
<td>$W[h + t] + b$</td>
<td>91.01</td>
</tr>
<tr>
<td>Linear</td>
<td>$L_2$</td>
<td>$W[t - h] + b$</td>
<td>95.89</td>
</tr>
<tr>
<td>Linear</td>
<td>$L_3$</td>
<td>$W[h; t] + b$</td>
<td>97.52</td>
</tr>
</tbody>
</table>

Observations:
- Concatenation (preservation of all features) performed best.
- Translation somewhat worse, but contains most needed information.
- Addition much worse.
## Results: Structural Probes

<table>
<thead>
<tr>
<th>Category</th>
<th>ID</th>
<th>Probe</th>
<th>Layer 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural</td>
<td>$S_1$</td>
<td>$h + r_r - t$</td>
<td>60.09</td>
</tr>
<tr>
<td></td>
<td>$S_2$</td>
<td>$Mh + r_r - Mt$</td>
<td>90.76</td>
</tr>
<tr>
<td>Constrained</td>
<td>$C_1$</td>
<td>$h + r_r - t$</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>$C_2$</td>
<td>$Mh + r_r - Mt$</td>
<td>73.57</td>
</tr>
</tbody>
</table>

Constrained means that a correct prediction is considered incorrect at prediction time if the constraints are not all satisfied.
Results: Structural Probes

<table>
<thead>
<tr>
<th>Category</th>
<th>ID</th>
<th>Probe</th>
<th>Layer 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural</td>
<td>$S_1$</td>
<td>$h + r_r - t$</td>
<td>60.09</td>
</tr>
<tr>
<td></td>
<td>$S_2$</td>
<td>$Mh + r_r - Mt$</td>
<td>90.76</td>
</tr>
<tr>
<td>Constrained</td>
<td>$C_1$</td>
<td>$h + r_r - t$</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>$C_2$</td>
<td>$Mh + r_r - Mt$</td>
<td>73.57</td>
</tr>
</tbody>
</table>

Observations:

- $S_2$ performs roughly on par with the linear probes.
- $C_1$ implies that $S_1$ largely fell back to linear classification, which implies *no* explicit structuring learned by the parser.
- $C_2$ performed decently well. May imply that there is some latent structure learned by the parser.
Conclusions and Future Work

- The parser does not explicitly learn a translational relation model.
- The parser may implicitly learn such a model, or one may be easily learned from its contextualized word representations.
- This implies a compatibility with such a model, which would yield useful relation embeddings.
- Such embeddings could, in future work, be used to enhance end-to-end neural pipelines, such as knowledge graph generation systems or machine readers.
Thank you!