Probing a Semantic Dependency Parser for Translational Relation Embeddings

Riley Capshaw, Marco Kuhlmann, and Eva Blomqvist

Linköping University, Linköping, Sweden
{riley.capshaw, marco.kuhlmann, eva.blomqvist}@liu.se

Abstract. Translational relation models are primarily applied to the task of Knowledge Graph embedding. We present a structural probe for testing whether a state-of-the-art semantic dependency parser learns contextualized word representations which fit a translational relation model. We find that the parser does not explicitly learn a translational relation model. We do, however, find that a simple transformation of the word representations is enough to induce a TransE model with 73.45% label recall, indicating that translational relation models are at least implicitly learned by the parser. We believe that our findings can in the future be used to develop new Natural Language Understanding systems that are more useful for Knowledge Graph generation and completion.

Keywords: Knowledge Graph Embedding · Deep Learning · Natural Language Understanding · Semantic Dependency Parsing

1 Introduction

A Knowledge Graph (KG) represents entities as nodes and relations as labeled, directed edges (s, p, o), indicating that a specific predicate p holds between the subject s and the object o. A *knowledge graph embedding* projects a KG into a high-dimensional vector space, often with geometric constraints enforced on the learned representations. For example, TransE [4] maps s and o to n-dimensional vectors $\mathbf{h}, \mathbf{t} \in \mathbb{R}^n$ and maps p to a translation vector $\mathbf{r}$ such that $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ whenever (s, p, o) holds. This type of geometric representation makes models highly interpretable due to the ease of explaining why relations are identified. This *interpretability* is a desirable feature of models in many domains. For example, geometric interpretations of word embeddings have helped to identify and handle gender bias in language models [3]. In this paper we use a *structural probe* to examine whether Natural Language Processing (NLP) models for relation prediction explicitly represent relations as translation operations. Our experiments focus on the task of *semantic dependency parsing* due to its structural and conceptual similarities with KG completion from natural language. We argue that in the future, in addition to studying the internal representation of an NLP component, such a probe can be used to identify steps in NLP pipelines that can be

harmonized with models used in downstream KG completion systems, which is shown to enhance the usefulness of NLP concepts from those steps [16].

This paper is structured as follows. Section 2 discusses necessary background information. Section 3 describes the experimental setup for probing a semantic dependency parser. Section 4 presents and analyzes the probing results. Finally, we discuss our conclusions and their implications for future work in Section 5.

2 Background and Related Work

The Semantic Web community has a long tradition of combining methods from NLP and Machine Learning to support various tasks related to ontologies, Linked Data, and lately the broader notion of KGs. For instance, early work on formalizing natural language to automate ontology learning heavily relied on classical NLP pipelines (e.g. FRED [5]), while more recent work also applied Deep Learning to similar tasks [2]. For a comprehensive introduction to KG creation from text (e.g. using NLP and Information Extraction techniques), we refer to the recent tutorial paper by Hogan et al. [8]. In this section we specifically target KG embeddings, then introduce semantic dependency parsing, and finally introduce the concept of probing neural networks, to set the stage for our experiments.

Knowledge Graph Embedding techniques have been developed to encode KGs into continuous vector spaces. KG embedding models which consider only entities and relations can be broadly categorized as either translational models (a.k.a. translational distance models) or semantic matching models [15]. Semantic matching models (e.g. RESCAL [11]) are not as relevant for this study due to not enforcing geometric interpretations for relation embeddings. Translational models primarily embed relations as vectors corresponding to translation operations. TransE [4] is the simplest such model, though it has difficulty modeling relationships which are not 1-to-1. Improvements upon TransE model more complex relationships as translation operations by contextualizing entity representations per relation (e.g. TransR [10]) or relaxing constraints (e.g. ManifoldE [17]).

Semantic Dependency Parsing is a task within NLP where individual sentences are annotated with binary *dependency* relations that encode shallow semantic phenomena between words [13]. While conceptually similar to semantic role labeling [6], semantic dependency parsing instead covers all content words to form a *semantic dependency graph* (SDG). A *semantic dependency parser* is a system that produces a SDG for a sentence annotated with linguistic features. Most parsers internally generate contextualized vector representations for each word, but to our knowledge, none provide easily recoverable relation representations, nor has their relation to translational models been studied prior to this work.

For this study we use the data from two SemEval shared tasks on semantic dependency parsing [13,14], where sentences from the Penn Treebank were annotated with three target semantic dependency representations. We only use the English DELPH-IN MRS (DM) [12] subset due to it being released publicly[1].

[1] Available at: https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-1956

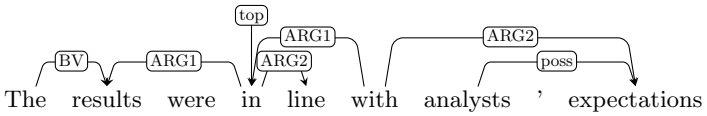

Fig. 1. Example SDG #22007003 from the DM dataset.

Figure 1 shows a DM SDG. Note how it focuses on argument relationships (e.g. `ARG1` and `ARG2`), such as 'results' and 'line' being arguments for the predicate 'in', and how semantically vacuous words like 'were' are left disconnected. For this example, one triple in our resulting SDG would be (in, `ARG2`, line).

Neural Network Probes were presented by Alain and Bengio [1] as a way of analyzing the intermediate layers of a deep neural network without influencing its training. They argue that convex classifiers (e.g., softmax classifiers trained by minimizing cross-entropy) can be used to approximate the upper bound on linear separability of features. Hence, we use linear probes to bound our experiments and to give a point of comparison due to a lack of directly comparable related work. Our experiments instead use structural probes, which analyze the structure of a learned vector space. For example, Hewitt and Manning [7] probe language models for syntactic structure by using ℓ_2 distance as an indication of parse tree distance. This differs from our work in that we also consider the direction of difference vectors to be informative.

3 Experimental Setup

We probe the semantic dependency parser developed by Kurtz et al. [9]. The parser first contextualizes every word of an input sentence using three layers of bidirectional long short-term memory (BiLSTM) units. Then, it scores edges as $\mathbf{h}_i^\top \mathbf{U} \mathbf{d}_j$, where $\mathbf{U}$ is a rank-three tensor such that $\mathbf{h}_i^\top \mathbf{U}^{[r]} \mathbf{d}_j$ is the scalar score for relation r from word i (specialized as the head) to word j (specialized as the tail or dependent). We use the parser to predict the SDG for every sentence in the training and testing sets from the SemEval shared task DM data. For every predicted dependency r between pairs of words h and t, we record the respective m-dimensional representations $\mathbf{h}, \mathbf{t} \in \mathbb{R}^m$ generated by each BiLSTM layer as the triple $(\mathbf{h}, r, \mathbf{t})$. The resulting training set has 600,871 triples with 50 distinct relations, while the testing set has 24,719 triples with only 39 distinct relations.

To establish an upper bound for recall using only $\mathbf{h}$ and $\mathbf{t}$, we formulate a simple *linear classifier probe* as in Alain and Bengio [1]. Given $\mathbf{x}$ as a combination of $\mathbf{h}$ and $\mathbf{t}$, we train softmax $(\mathbf{W}\mathbf{x} + \mathbf{b})$ by minimizing cross-entropy, where $\mathbf{W}$ and $\mathbf{b}$ are the weights and biases to learn. By setting $\mathbf{x} = \mathbf{t} - \mathbf{h}$, we can directly evaluate the translation vector. We contrast this with concatenation ($\mathbf{x} = [\mathbf{t}; \mathbf{h}]$), which preserves all input features but enlarges $\mathbf{W}$, and addition ($\mathbf{x} = \mathbf{h} + \mathbf{t}$).

The *structural probes* are formulated nearly identically to translational embedding models. We first define a linear transformation matrix $\mathbf{M} \in \mathbb{R}^{m \times m}$ to be

learned. Then, for every triple $(\mathbf{h}, r, \mathbf{t})$ we normalize $\mathbf{Mh}$ and $\mathbf{Mt}$ to unit length, then calculate $\mathbf{r} = \mathbf{Mt} - \mathbf{Mh}$, where $\mathbf{r}$ is the vector representing a particular instance of r as a translation operation. Since a perfect model would result in $\mathbf{r}$ being equal for all triples with the same r, we also define a representative vector $\mathbf{r}_r$ for each r such that $\mathbf{r}_r - \mathbf{r} \approx \mathbf{0}$. We then approximate all $\mathbf{r}_r$ and $\mathbf{M}$ through gradient descent by minimizing the scoring function

$$f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{Mh} + \mathbf{r}_r - \mathbf{Mt}\|_2^2 \,, \tag{1}$$

where $\|\cdot\|_2^2$ is the squared ℓ_2 norm. We do not enforce strict translation and instead define a margin γ such that $f_r(\mathbf{h}, \mathbf{t}) \leq \gamma$. Since a given $\mathbf{h}$ and $\mathbf{t}$ can have only one relationship, we enforce $f_{r'}(\mathbf{h}, \mathbf{t}) > \gamma$ for all $r' \neq r$ to maximize the separation of regions corresponding to relations. We also encourage all $\mathbf{r}$ for a given relation to have roughly the same direction with the constraint $\|\mathbf{r}\|_2^2 > 2\gamma$. Incorporating these gives us the following margin-based loss function:

$$\mathcal{L} = \left[f_r\left(\mathbf{h}, \mathbf{t}\right) - \gamma\right]_+ + \sum_{r' \neq r} \left[\gamma - f_{r'}\left(\mathbf{h}, \mathbf{t}\right)\right]_+ + \left[2\gamma - \|\mathbf{r}\|_2^2\right]_+ \,, \tag{2}$$

where $[\cdot]_+$ is equivalent to $\max(0, \cdot)$. In order to see if a translational model is explicitly learned by the parser, we also define a probe where the input vectors are not transformed: $\mathbf{r} = \mathbf{t} - \mathbf{h}$. Finally, to test whether the classifiers actually fit a relational model, we recalculated both structural probes' scores without retraining by considering a prediction to be correct only if the constraint $f_r\left(\mathbf{h}, \mathbf{t}\right) \leq \gamma$ was satisfied. For all experiments, we set $\gamma = 0.25$ and trained for 100 epochs.

All scores are reported as micro-averaged recall. Since we are only interested in the probes' abilities to reconstruct the parser's relation predictions, we do not predict the absence of a relation between words. If we did, incorrect predictions would only affect *precision*, correct predictions would affect neither precision nor recall, and micro-averaged recall would no longer be equal to label accuracy.

4 Results and Analysis

This section presents and analyzes the results from the experiments outlined in Section 3. Table 1 presents all recall scores broken down by probe types, where 'Constrained' refers to structural probes where the margin constraint is strictly enforced at prediction time. As a baseline for comparison, guessing only the most frequent relation (`ARG1`) scores 37.32%. Most probes outperform this baseline, and the scores increase consistently with layer depth for all but one probe.

For the linear probes (L_1 to L_3), we see the highest overall recall achieved by the concatenation probe (L_3), likely due to preserving the most input information. It outperforms the difference probe (L_2) by 1.73, which in turn outperforms the addition probe (L_1) by 5.01. This indicates that the translation vectors already encode much of the information necessary for classification.

For the structural probes, probe S_1 (which does not transform the input vectors) still outperformed the baseline, yet its scores for almost none of the

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

Table 1. Scores for all probing experiments in terms of recall as a function of layer.

triples satisfied the margin constraint (C_1). This shows that it did *not* actually fit a translational model. Instead, it likely defaulted to a linear partitioning of the vector space akin to a multiclass perceptron. Probe S_2 performed on par with L_1 and still performed well when the margin was enforced at prediction time (C_2), indicating that it *did* learn a translational model. We also analyzed the effect of relaxing γ at prediction time by increasing it until all $f_r(\mathbf{h},\mathbf{t}) \leq \gamma'$. For probe C_1 this occurred at $\gamma' = 5\gamma = 1.25$, and for probe C_2 this occurred much sooner at $\gamma' = 2.4\gamma = 0.6$. This indicates that the margin may need adjustment at prediction time. Recall that if the margin was satisfied for some part of the loss, then the loss for that part was zero.

When analyzing the class distribution of the probes' predictions, we noticed that all three linear probes completely missed five relations, indicating an inability to handle imbalanced classes. We found instead that probes S_2 and C_2 missed two relations and probes S_1 and C_1 missed only one. Despite having a lower overall recall, both structural probes seem to capture rare relations better.

5 Conclusions and Future Work

In this work, we explored the relationship between NLP and KGs by analyzing the structure of the vector space learned by a semantic dependency parser. Applying KG techniques to NLP tasks is an attempt to further harmonize the two areas, allowing for continuous representations of NLP concepts to be more naturally incorporated into deep-learning-based KG generation pipelines. Such representations capture information important for label predictions which a KG generation system would be unable to learn on its own. To do this exploration, we presented two structural probes inspired by translational embedding models. We found that the semantic dependency parser we probed does not explicitly learn a translational model. However, a single linear transformation matrix is sufficient to fit such a model to the parser's contextualized word representations with 73.45% recall, implying that the parser is implicitly learning a translational model. A promising path forward is then to implement a decoder for a semantic dependency parser based on translational relation models to yield explicit,

interpretable relation vectors alongside the full SDG. These can then be used in a KG completion system based on translational embeddings (of which there are many [15]), or in a full Natural Language Understanding pipeline which jointly trains all components, from NLP subsystems to the KG generation itself.

Acknowledgments

This research work was funded in part by CUGS (the National Graduate School in Computer Science, Sweden).

References

1. Alain, G., Bengio, Y.: Understanding intermediate layers using linear classifier probes. In: Proc. of ICLR. OpenReview.net (2017)
2. Arguello Casteleiro, M., et al.: Deep learning meets ontologies: experiments to anchor the cardiovascular disease ontology in the biomedical literature. Journal of Biomedical Semantics **9**(1) (2018)
3. Bolukbasi, T., et al.: Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In: Proc. of NeurIPS. Springer (2016)
4. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proc. of NeurIPS (2013)
5. Gangemi, A., et al.: Semantic web machine reading with FRED. Semantic Web **8**(6) (2017)
6. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. Computational linguistics **28**(3) (2002)
7. Hewitt, J., Manning, C.D.: A structural probe for finding syntax in word representations. In: Proc. of NAACL. ACL (2019)
8. Hogan, A., et al.: Knowledge graphs (2020), https://arxiv.org/abs/2003.02320
9. Kurtz, R., Roxbo, D., Kuhlmann, M.: Improving semantic dependency parsing with syntactic features. In: Proc. of the First NLPL Workshop on Deep Learning for Natural Language Processing. Linköping University Electronic Press (2019)
10. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proc. of AAAI. AAAI Press (2015)
11. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proc. of ICML. Omnipress (2011)
12. Oepen, S., Lønning, J.T.: Discriminant-based MRS banking. In: Proc. of LREC (2006)
13. Oepen, S., et al.: Semeval 2014 task 8: Broad-coverage semantic dependency parsing. In: Proc. of the 8th Int. Workshop on Semantic Evaluation. ACL (2014)
14. Oepen, S., et al.: Semeval 2015 task 18: Broad-coverage semantic dependency parsing. In: Proc. of the 9th Int. Workshop on Semantic Evaluation. ACL (2015)
15. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE TKDE **29**(12) (2017)
16. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph and text jointly embedding. In: Proc. of EMNLP. ACL (2014)
17. Xiao, H., Huang, M., Zhu, X.: From one point to a manifold: Knowledge graph embedding for precise link prediction. In: Proc. of IJCAI. IJCAI/AAAI Press (2016)