Reproducibility of Orthogonal Structural Probes

Adam Hamden and Hitesh Pindikanti
University of Southern California
{ahamden, pindikan}@usc.edu

1 Introduction

Understanding the information encoded in the embedding space for word representations in neural machine learning models is an important task for model interpretability and extensibility. Structural probes are models that attempt to uncover syntactic tree representations in linear transformations of the neural embeddings (Hewitt and Manning, 2019). This paper attempts to reproduce the work on orthogonally constrained structural probes by Limisiewicz and Mareček (2021). The ultimate goal is to determine whether a linear transformation of the embedding can be learned such that the resulting projection can inform the structure of dependency trees and lexical hypernymy through embedding distances. Furthermore, the original paper explores whether the word embedding vector norm after the linear transformation can inform absolute word position in a sentence. The original paper claims that the addition of orthogonal constraints on the linear transformation makes the structural probe less susceptible to memorization during training.

2 Scope of reproducibility

In our experiment, we will run several orthogonally constrained structural probes to predict the lexical, positional, random, and dependency depths and distances on the Universal Dependencies English Web Treebank dataset. The lexical hypernymy task also utilizes the WordNet tree (Miller, 1995). The experiments will probe the English BERT large case model (Devlin et al., 2019). We will focus on replicating the results of the paper for the specific layers identified by the paper as achieving the optimal results on the model for a particular training configuration. The baseline used in the evaluation is the traditional structural probe (Hewitt and Manning, 2019).

Orthogonal structural probes reveal a nearly equivalent Spearman's rank correlation on the dependency, lexical hypernymy, and position in a sentence depth and distance tasks when compared to the traditional structural probes on the English BERT large case model (Devlin et al., 2019). We will verify that introducing orthogonal constraints on the probe does not sacrifice the probe's performance relative to an unconstrained probe.

The correlation for random structures (depths and distances) is weak across the board which implies that orthogonal structural probes do not memorize training data structures but rather infer them from the embedding. We will verify that the correlation on the random tree depths and distances is lower on the orthogonally constrained probe than on the traditional probe, thus implying the constraint curbs memorization.

Training with joint objectives results in lower correlations for all configurations while yielding higher selectivity[1].

Additionally, the BERT subspace encoding linguistic hypernymy does not overlap with the subpspace encoding the dependency and position in a sentence information. We will verify that the subspace dimensions do not overlap.

2.1 Addressed claims from the original paper

We will test the following claims made by the original paper

- Orthogonal structural probes achieve equivalent results to structural probes on the dependency, lexical, and position in sentence objectives.
- Orthogonal structural probes are less prone to memorization than structural probes as in-

[1] selectivity score is defined as the difference between performance on the random trees and the average dependency, lexical, position in a sentence correlations

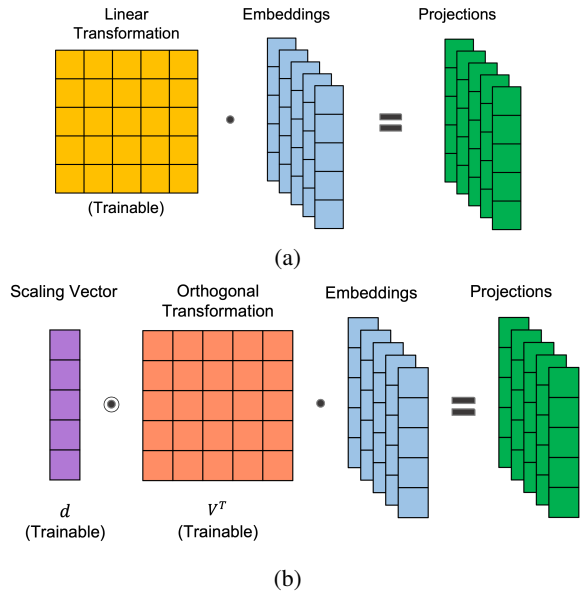


Figure 1: Comparison of the Structural Probe of (Hewitt and Manning, 2019) and the Orthogonal Structural Probe proposed by (Limisiewicz and Mareček, 2021).

formed by the lower Spearman's rank correlation on randomly generated trees.

- BERT subspace encoding linguistic hypernymy is separate from subspace encoding dependency and position in sentence implying representations are orthogonal in the embedding space.

3 Methodology

We examine both the *Structural Probe* proposed by (Hewitt and Manning, 2019) and the new *Orthogonal Structural Probe* proposed by (Limisiewicz and Mareček, 2021). The Figure 1 shows a comparison between the two approaches. We try to replicate the results of *Orthogonal Structural Probe* and verify the claims made by original authors. Additionally, we perform ablation studies by including new distance and correlation metrics to train and evaluate the models respectively.

3.1 Model descriptions

The author in the original paper optimizes the *Orthogonal Structural Probes* over distance and depth probes in several types of structures: dependency, lexical hypernymy, position in a sentence, and randomly generated trees. The distance is calculated between pairs of words and the depth of a word is calculated from the root of the tree. The following descriptions of the tasks are directly referenced from the original paper (Limisiewicz and Mareček, 2021):

Syntactic Dependency: The syntactic structure in the Universal Dependencies parse trees which are annotated in English Web Treebank dataset.

Lexical Hypernymy: The hypernymy tree referenced from WordNet (Miller, 1995). The author considered lexical distances between pairs of nouns and pairs of verbs in sentences and lexical depth for each noun and verb.

Position in a Sentence: Given a sentence, the depth is the index of a word and the distance is the difference between the indices of pairs of words.

Random Structures: A randomly generated tree from the words in the sentence. This structure acts as a control that will reveal any memorization during training. We expect to get low correlations on the random tree since there is no coherent structure we should reasonably uncover.

If h_i, h_j are the word vectors at positions i and j in a sentence, then the tree distance is approximated by the squared norm of the differences between the transformed vectors shown in Equation 1 where V is the orthogonal matrix and d is the scaling vector formed by performing single value decomposition of the transformation matrix used in *Structural Probe*.

$$d_L(h_i, h_j)^2 = \|\bar{d} \odot V^T(h_i - h_j)\|^2_{\bar{d}V^T} \quad (1)$$

To approximate a word's depth in a dependency tree from its syntactic root, we use the squared norm of the word vector h_i i.e., $\|h_i\|^2_{\bar{d}V^T}$.

The author of the original paper uses squared L2-norm for distance and depth probes, but we try using different norms and analyse their results in Section 4.

The final training loss for *Orthogonal Distance Probe* and *Orthogonal Depth Probe* shown in Equations 2 and 3 respectively are normalized by the prediction count in a sentence and averaged across a batch. We use two regularization terms: *Double Soft Orthogonality Regularization* (DSO) (Bansal et al., 2018) of the orthogonal matrix V and L1-norm sparsity regularization of the scaling vector d.

$$L_{o,dist.} = \frac{1}{s^2} \sum_{i,j} |d_T(\omega_i, \omega_j) - d_{\bar{d}_o V^T}(h_i, h_j)^2| + \lambda_o DSO(V) + \lambda_S \|\bar{d}\|_1 \quad (2)$$

$$L_{o,depth} = \frac{1}{s}\sum_{i} |\|\omega_i\|_T - \|h_i\|^2_{\bar{d}_o V^T}| + \lambda_o DSO(V) + \lambda_S \|\bar{d}\|_1 \quad (3)$$

We use the pretrained 24-layered English BERT large case model (Devlin et al., 2019) provided through HuggingFace (Wolf et al., 2020) to train our probes on top of each layer. We optimize the distance and depth probe on several structures as explained earlier.

The number of parameters in the *Orthogonal Structural Probe* for BERT Large for all eight objectives are $1,056,768$. The details for this calculation can be found in the original paper (Limisiewicz and Mareček, 2021).

3.2 Data descriptions

The dataset is downloaded from Universal Dependencies English Web Treebank[2] (Silveira et al., 2014). This will be used as our gold standard corpus which is constructed with the original English Web Treebank LDC2021T13[3]. The corpus is in CoNLL-U format[4] defined for Universal Dependencies. In total, the dataset provides 16,622 annotated sentences with train (12,543), validation (2,002), and test (2,077) partitions. This information covers five topics of internet content that include weblogs, newsgroups, emails, reviews, and Yahoo! answers.

3.3 Hyperparameters

We use the same hyperparameters as the author of the original paper (Limisiewicz and Mareček, 2021) mentioned in their paper and code in order to replicate the results. The publicly released code contained additional hyperparameters that were not specified in the paper. In these instances, we did our best to determine which configurations were used. We detail this issue further in Section 5.2.

The original paper uses a batch size of 12 and a decaying learning rate with an early stopping mechanism starting with an initial learning rate of 0.02. The learning rate is divided by 10 when the validation loss does not decrease after an epoch. Training is stopped when three consecutive learning rate updates do not result in a smaller loss. *Orthogonal Regularization* (λ_O) is set to 0.05 and *Sparsity Regularization* (λ_S) is set to 0 by default to replicate the results.

Library	Version
numpy	1.19.5
tensorflow	2.4.1
tensorflow-hub	0.8.0
transformers	4.3.2
tqdm	4.46.0
unidecode	1.1.1
nltk	3.5
networkx	2.5
pytest	6.1.2
ufal.chu-liu-edmonds	1.0.1

Table 1: Python libraries used in the code

3.4 Implementation

We extend the code provided by the authors of the original paper[5] to include our ablation studies. The code is written in Python programming language and Table 1 shows the packages used in the implementation in addition to specific versions. The extended code is available at https://github.com/hiteshpindikanti/OrthogonalTransformerProbing.

3.5 Experimental setup

We run our experiments on high-performing computing cluster: Discovery, provided by the USC's Center for Advanced Research Computing (CARC). Discovery provides multiple resource configurations to run our experiments including NVIDIA Tesla K40, V100, A100, A40 GPUs.

3.6 Computational requirements

The authors of the original paper trained the Orthogonal Structural Probes on a *GeForce GTX 1080 Ti* GPU where as we used the *NVIDIA A40 GPU* with 16GB RAM and 2 cpu-per-task. The difference in the runtimes is shown in Table 2

We observe higher runtimes than what author of the original paper has claimed. We suspect that the reason is due to the difference in the GPU hardware and the CUDA versions that created a difference in the Tensorflow optimizations. NVIDIA A40 GPU is relatively old hardware with an older CUDA version which was difficult to configure for current

[2]The dataset is available at https://github.com/UniversalDependencies/UD_English-EWT

[3]https://catalog.ldc.upenn.edu/LDC2012T13

[4]https://universaldependencies.org/docs/format.html

[5]https://github.com/Tom556/OrthogonalTransformerProbing

Task	Original Time	Replication Time
Probing for depth	3 mins	5 mins
Probing for distance	5 mins	10 mins
Joint probing for distance and depth in the same structure type	7 mins	12 mins
Joint probing for depths in all structures	13 mins	20 mins
Joint probing for distance in all structures	18 mins	30 mins
Probing for all objectives	35 mins	60 mins

Table 2: Difference in Runtimes of original author's claim and our replication runtimes. Note that we ran some tasks just to verify the runtimes, and not all the tasks come under our scope of reproducibility

version of Tensorflow. We also see a relatively similar trends in the runtimes with respect to various tasks in author's experiments and our replication experiments.

Since the max runtime for any experiment is around an hour, this made it feasible to run multiple experiments and ablations by scheduling multiple jobs to be run parallel. On average, it took 17 ± 2 seconds to run an epoch. Similar to the original paper, we ran the experiments six times to yield an average metric scores for all the objectives. Additionally, we ran about one experiment of all the eight objectives for each of the 24 layers of BERT and six more trials for our ablation experiments. Overall we approximately used 40 GPU hours to run all experiments and tests.

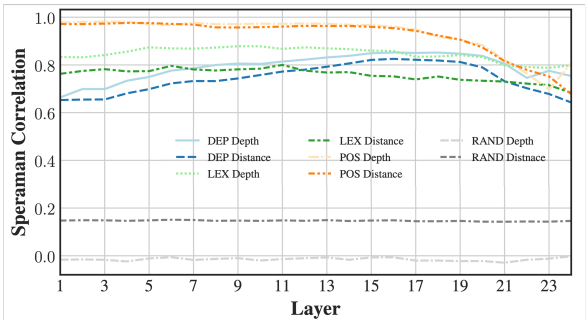
4 Results

We ran experiments with identical hyperparameters as the original paper described in Section 3.3 to reproduce the author's results. We calculate the Spearman's correlations between predicted values and gold tree depths and distances and compare our reproduced results with the author's claimed results in Table 3.

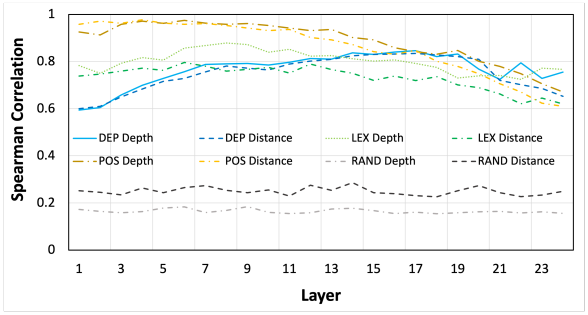
We verify the claims made my the author of the original paper mentioned in Section 2.1 in the following subsections:

4.1 Results Closeness with Structural Probes

The original paper claims that the results obtained by *Orthogonal Structural Probes* are close to those of *Structural Probes*. Our reproduced results also follows the same claim with some minor variations in the Spearman's correlation values. We suspect



(a) Original Paper Results



(b) Reproduced Results

Figure 2: Comparison of Spearman Correlations across layers for joint training

these minor variations account for the randomness in the code and possibly other trivial hyperparameter tuning which was not mentioned in the original paper.

4.2 Less Memorization than Structural Probes

The original paper claims that *Orthogonal Structural Probes* are less prone to memorization. We observe this claim by looking the random tree structure experimentation. *Orthogonal Structural Probes* have a relatively less Spearman's rank correlation values than the *Structural Probes* for random generated trees. This also follows in the findings of our reproduced results. Although the reproduced values are not as low as the original paper, we still see a marked decrease compared to the *Structural Probe's* values.

4.3 BERT Subspace Encoding of different Structures

The original paper claims that different structures are encoded in different layers/subspace in the BERT model. We compare the Spearman's rank correlation values for each objective at every layer in Figure 2. We observe a similar trend in our reproduced results as in the original paper. The mid-upper layers contain more syntactic informa-

Objective	Structural Probe	Orthogonal Probe	Orthogonal Probe (Reproduced)
Dependency Parsing Depth	0.851 ± 0.001 (18th layer)	**0.858** ± 0.001 (17th layer)	0.846 ± 0.004
Dependency Parsing Distance	**0.843** ± 0.001 (17th layer)	**0.842** ± 0.001 (17th layer)	0.834 ± 0.003
Lexical Hypernymy Depth	**0.892** ± 0.002 (8th layer)	0.882 ± 0.002 (8th layer)	0.879 ± 0.002
Lexical Hypernymy Distance	**0.816** ± 0.008 (6th layer)	0.803 ± 0.005 (6th layer)	0.796 ± 0.004
Position in Sentence Depth	**0.989** ± 0.001 (1st layer)	0.983 ± 0.001 (6th layer)	0.975 ± 0.002
Position in Sentence Distance	**0.980** ± 0.001 (4th layer)	0.979 ± 0.001 (4th layer)	0.977 ± 0.003
Random Structures Depth	0.206 ± 0.010 (17th layer)	**0.136** ± 0.007 (18th layer)	0.154 ± 0.004
Random Structures Distance	0.242 ± 0.005 (19th layer)	**0.220** ± 0.006 (18th layer)	0.226 ± 0.003

Table 3: Original paper's results and our Reproduced Orthogonal Probe Results. Similar to original paper, we ran our experiments six times to calculate the mean and the standard deviation on the same layer of BERT model as the original paper.

tion, while the mid-lower layers have more lexical information. Word positions in a sentence can be better predicted in the starting layers than the ending layers. This seems reasonable since positional embedding are added before the first layer in BERT, as noted in the original paper. Orthogonal structural probes on the randomly generated trees have consistently low scores throughout all the layers.

We also observe that our reproduced results in the Figure 2b are a somewhat erratic in nature. We suspect that the reason behind this is because the original work performed multiple runs and averaged them to get smooth trends, whereas we plotted the results obtained from only a single run.

4.4 Additional Experiments and Ablations

4.4.1 Tree Distance Methods

It seemed interesting to use a different tree distance methods to train and evaluate the models and compare those results, hence we tried using L1 , L3, and L-infinity distances along with original L2 distance to approximate tree distances in the predicted embeddings. The results are shown in Table 4. To run these experiments, we use the same hyperparameters as described in Section 3.3.

We observe that L1 and L2 distance performed better than L3. This is because we calculate the cube of the vector components, which increases the weights on outliers; this in turn results in skewed results. L-infinity only considers the vector component with the largest magnitude, which clearly is a case of classic information loss. This explains the relatively low score when we approximate tree distances using L-infinity norm. L1 distance, also known as Manhattan distance, seems to be a relatively good metric. The difference in scores compared to the L2 distance metric can be explained by the fact that the hyperparameters were tuned to optimize for the case of L2 distance metric. If we tried to tune the hyperparameters for other distance metrics, we might get some improvement in their respective scores.

4.4.2 Correlation Metrics

We used three different correlation metrics: Pearson, Spearman, and Kendall correlations to evaluate the model. The results for this experiment is shown in Table 5. We use the same hyperparameters mentioned in Section 3.3 with the L2 distance metric to approximate the tree distances between word embeddings.

We know that the Spearman and Kendall correlations work best with ordinal data, whereas Pearson correlations can be used with non-ordinal data too. When used with ordinal data (integer tree distances in our case) Pearson and Spearman correlations work identically. The same can be observed in the

Tree Distance	L1	L2	L3	L-inf
Dependency Parsing Depth	0.690	**0.840**	0.632	0.286
Dependency Parsing Distance	0.711	**0.829**	0.589	0.253
Lexical Hypernymy Depth	0.723	**0.875**	0.596	0.189
Lexical Hypernymy Distance	0.704	**0.790**	0.537	0.197
Position in Sentence Depth	0.788	**0.972**	0.623	0.376
Position in Sentence Distance	0.793	**0.972**	0.602	0.365
Random Structures Depth	0.170	**0.159**	0.165	0.134
Random Structures Distance	0.231	**0.228**	0.230	0.154

Table 4: Spearman's Correlation values of *Orthogonal Structural Probe* for different tree distance approximations. The experiments were run once with the same hyperparameters mentioned in Section 3.3

Correlation Metric	Pearson	Spearman	Kendall
Dependency Parsing Depth	0.838	**0.840**	0.412
Dependency Parsing Distance	0.825	**0.829**	0.404
Lexical Hypernymy Depth	0.852	**0.875**	0.437
Lexical Hypernymy Distance	0.757	**0.790**	0.387
Position in Sentence Depth	0.948	**0.972**	0.478
Position in Sentence Distance	0.964	**0.972**	0.482
Random Structures Depth	0.164	**0.159**	0.160
Random Structures Distance	0.237	**0.228**	0.245

Table 5: Evaluating the model with different correlation methods

table results. On the other hand, Kendall Correlation is a test of strength of dependence on two variables, which considers a lot more factors than just correlation. Hence we see a difference in the values compared to Pearson and Spearman correlations. We also observe a consistency in the values over the eight objectives with respect to each correlation metric.

5 Discussion

The claims within our scope of reproducibility were all verified by our set of experiments. We verified that that orthogonal structural probes achieve performance equivalent to the traditional structural probes. We further verified that orthogonal constraints results in less memorization when compared to traditional structural probes. Furthermore, we verified that the BERT subspace encodings of different structures were optimally found in different layers.

5.1 What was easy

The easiest part about reproducing this paper was the fact that both the code and the dataset were publicly available. This ensured that we did not deviate from the model parameters of the original paper, thus ensuring a fairer attempt at reproduction. In addition, the original paper directly extended the work of (Hewitt and Manning, 2019) which provided an additional reference on structural probes.

5.2 What was difficult

The experiments were run on the Discovery cluster with a different GPU cluster. Due to the nature of this change, our replication time was significantly longer by nearly two-fold. This posed a difficulty in running the ablations as we were not able to try out as many different experiments. Furthermore, the difference in CUDA versions addressed in Section 3.6 required some time for troubleshooting as it required non-obvious changes to the original code's Tensorflow setup.

The original paper mentions the set of hyperparameters used to train the probe such as learning rate and other regularization terms. A closer examination of the original code also revealed additional hyperparameters such as hidden layer size and cased/uncased configurations that were trained with but not specified in the original paper. This made it difficult to determine which hyperparameters corresponded to the author's results.

We also had to slightly modify the code to remove several hard coded file-paths assumed by the original authors.

5.3 Recommendations for reproducibility

The original authors did a great job in providing publicly available code and data with documentation for reproduction. Save for a few minor difficulties, we were able to run the complete set of experiments outlined by the paper in addition to verifying their findings.

We recommend that any hard coded file-paths be removed from the code and instead changed into

either command line arguments or configurations stored in a single file.

We further recommend that the paper specify a complete set of hyperparameters used to train the orthogonal structural probe that yielded the final results detailed in the paper.

6 Communication with original authors

We reached out to the authors of the original work, informing them of our intent to reproduce their results and asking if there is anything we should be aware of when running their code. Unfortunately, we did not receive a response from the authors. Nonetheless, we intend to share our results with them in addition to making them aware of the issues we encountered when running their published code. Our hope is to provide them with valuable advise that may assist in further works.

References

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. 2018. Can we gain more from orthogonality regularizations in training deep cnns?

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Tomasz Limisiewicz and David Mareček. 2021. Introducing orthogonal constraint in structural probes.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.