Syntax-guided Neural Module Distillation to Probe Compositionality in Sentence Embeddings

Rohan Pandey
Language Technologies Institute
Carnegie Mellon University
rspandey@cs.cmu.edu

Abstract

Past work probing compositionality in sentence embedding models faces issues determining the causal impact of implicit syntax representations. Given a sentence, we construct a neural module net based on its syntax parse and train it end-to-end to approximate the sentence's embedding generated by a transformer model. The distillability of a transformer to a Syntactic NeurAl Module Net (SynNaMoN) then captures whether syntax is a strong causal model of its compositional ability. Furthermore, we address questions about the geometry of semantic composition by specifying individual SynNaMoN modules' internal architecture & linearity. We find differences in the distillability of various sentence embedding models that broadly correlate with their performance, but observe that distillability doesn't considerably vary by model size. We also present preliminary evidence that much syntax-guided composition in sentence embedding models is linear, and that non-linearities may serve primarily to handle non-compositional phrases.

1 Introduction

The principle of semantic compositionality suggests that the meaning of a sentence should derive from its subconstituents in a regular, structured fashion (Montague, 1970). In recent years, transformers (Vaswani et al., 2017) have become effective at producing sentential meaning representations useful for downstream tasks such as Natural Language Inference, Image-Text Matching, and Document Classification (Conneau et al., 2017; Radford et al., 2021). However, it has famously been conjectured that "You can't cram the meaning of a whole %&!$# sentence into a single $&!#* vector" (Mooney, 2014). Since recent models do appear to capture sentence meaning effectively, one wonders how they compose arbitrarily many word meanings together such that their relational structure is captured in a single, fixed-dimensional sentence embedding.

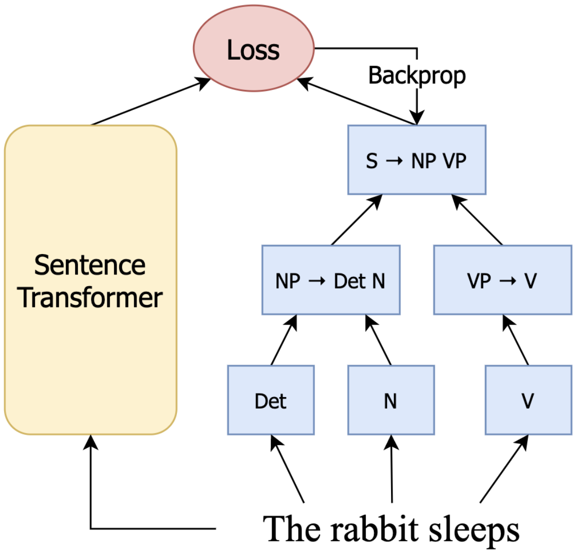

Figure 1: Distilling a transformer to a neural module net structured by the sentence's syntax

Much work has sought to probe these models for syntax representations and their causal relevance to embedding output. Conneau et al. (2018) train linear probes to determine if models encode syntactic features like tree distance and depth (Krasnowska-Kieraś and Wróblewska, 2019; Hewitt and Manning, 2019). One line seeks out direct mappings between neural representations and tree structures (McCoy et al., 2018; Chrupała and Alishahi, 2019; Jawahar et al., 2019; Soulos et al., 2020; Murty et al., 2023). Other work raises methodological issues with probing (Eger et al., 2019; Zhu and Rudzicz, 2020) such as choice of formalism (Kuznetsov and Gurevych, 2020) and semantic entanglement (Maudslay and Cotterell, 2021). Ravichander et al. (2021) raise the possibility that probing may identify causally un-used features; Tucker et al. (2021) partly address this concern to show that some syntactic features are causally relevant. Another line of work explores the geometry of semantic representations (Reif et al., 2019; Hernandez and Andreas,

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 3143–3149
May 2-6, 2023 ©2023 Association for Computational Linguistics

2021) and the linearity (Barančíková and Bojar, 2019) of syntactic analogies (Zhu and de Melo, 2020).

Rather than directly analyze sentence embedding models, Neural Module Nets (Andreas et al., 2016b) seek to improve compositionality by modularizing semantic functions. We see this effort as ultimately similar to probing for structural representations since the former explores whether explicit structure improves performance and the latter explores whether performant models implicitly learn structure. Geiger et al. (2021) discovers logical tree causal structures in BERT and Wu et al. (2021) then guides model distillation using this structure.

Our work builds on these findings by strictly taking syntax as the causal structure of sentential semantics and linearity as the geometry of syntax-guided composition; we conduct experiments to test the distillability of transformer-based sentence embedding models to a Syntactic NeurAl Module Net (SynNaMoN), an architecture we introduce that implements these two priors. The extent to which a model can be distilled to a SynNaMoN tells us about its internal syntax representations & compositional ability.

2 Methods

2.1 Syntactic Neural Module Net

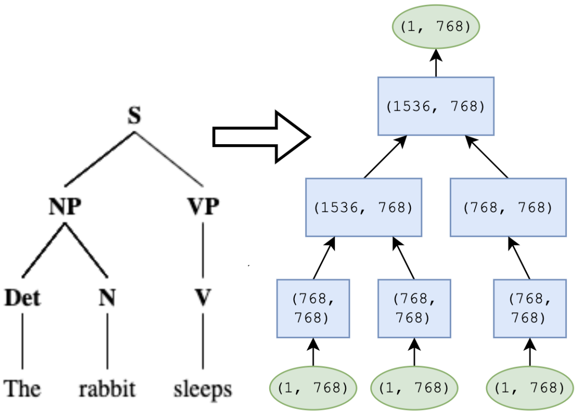

Figure 2: Constructing a sentence's SynNaMoN from its syntax tree; module input and output dimensionalities labeled on the right.

Unlike prior work (Andreas et al., 2016a; Cirik et al., 2018), SynNaMoN modules don't approximate high-level objectives like 'Find' or 'Count' but rather correspond to specific syntactic rules like 'S $\rightarrow$ NP VP' and 'NP $\rightarrow$ DT JJ NN'. Each module receives an input of dimensionality $(1, N * D)$ where N is the number of constituents on the syntax rule's right-hand side, and D is the dimensionality of the embedding space (768)—in other words, the input embeddings are concatenated. Though computationally more expensive, concatenation enables the module to learn an arbitrary function over the inputs rather than restricting it to a function over their sum or mean; this enables the module to converge on its ideal composition function which is likely not invariant under summation or averaging. Finally, though our implementation of SynNaMon includes 'part-of-speech' modules at the bottom of the parse tree, one could conceivably remove this bottom layer with the hypothesis that the word embeddings already capture part-of-speech information.

2.2 Internal Module Architecture

To explore the geometry of semantic composition under syntax, we implement 3 module architectures: a linear layer (**Linear**), a linear layer + a ReLU activation (**Nonlin**), and a linear layer + ReLU + another linear layer (**Double**). We explore these 3 architectures to see whether syntax is enough of an inductive bias to linearly approximate sentence embeddings, or if adding non-linearities and additional layers considerably improves performance. The extent to which adding parameters improves our approximation of the teacher model beyond the syntactic structure alone could reveal how much isn't captured by this inductive bias.

2.3 Linguistic Formalism

We choose to use the Transformational Grammar presented by Penn Treebank (Marcus et al., 1994), but in principle any Constituency Grammar could be easily used with SynNaMoN, and Dependency Grammars can be adapted with some effort. Since prior work has shown how the choice of linguistic formalism can significantly influence probing results (Kuznetsov and Gurevych, 2020), we float the possibility of such an effect being at play in this work as well. If a student SynNaMoN fails to capture much of the teacher embedding model, perhaps it isn't because of the teacher's non-compositional causal structure, but rather because the formalism used to structure the SynNaMoN is inadequate. Indeed, recent state-of-the-art neural approaches to syntax parsing have learned grammatical tagsets that often differ starkly from human-produced syntactic theories (Kitaev et al., 2022). We leave these problems to future work which may explore the exciting possibility that certain linguistic formalisms (perhaps even semantic rather than syntactic) are better proxies for a model's compositional structure than others.

3 Experiments

Our main experiment runs 5 sentence embedding models (BERT-base (Devlin et al., 2019), MP-Net (Song et al., 2020), GTR-T5-base, GTR-T5-large, and GTR-T5-xl (Ni et al., 2021)) on 3 SynNaMoNs with differing internal architectures (Linear, Nonlin, Double; see Sec. 2.2). For BERT-base, we extract input word embeddings for each token and use the CLS token as the sentence embedding as is common practice. For the other 4 models, we encode each token alone to serve as its embedding and use the output as the sentence embedding. When words are encoded as more than 1 token, we compute the mean across the subtokens to serve as its word embedding.

In order to heuristically select a learning rate, 5 training runs were conducted with SynNaMoNs optimizing for BERT-base, and learning rate manually set at increments between 10^{-5} and 10^{-3}. We finally chose a rate of 5×10^{-5}, but recognize from results that optimal learning rate will likely vary by teacher model & SynNaMoN internal architecture. Analysis would best be reported on the optimal scores achieved by a SynNaMoN after hyperparameter tuning, but due to compute restrictions (1 NVIDIA K80 GPU with 12GB of RAM), this was unfeasible.

Additionally, due to the number of modules (originally 900, each with 1M parameters on average), we encountered frequent out-of-memory errors both on CPU & GPU. Since each module corresponds to a syntax rule and is initialized upon encountering the rule in the dataset, we constrained our data to minimize the number of modules needed.

Specifically, we first constrained our trees to those of height 4 & 5 (n=16492) in PTB, and then further constrained the trees to those that use a subset of the 300 most common production rules among them. This resulted in 1494 trees, from which we generated a train-validation split of 1250-244. Furthermore, we ensured that all the productions present in trees of the validation split were also included amongst trees in the training split. All this finally resulted in 273 production rules present in our dataset, and the instantiation of 273 modules.

4 Results

In Tab. 1, we present scores for all 5 sentence embedding models across the 3 SynNaMon architectures. We compute the average MSE between sentence embeddings in the complete dataset for each model and divide each model's MSE loss by this mean distance to normalize results. The normalized scores we present may intuitively be seen as the portion of variance in a model's sentence embeddings that a SynNaMoN fails to explain. From a probing perspective, the lower a model's score, the more it can be causally approximated by composition along syntactic lines.

Sent. Emb. Model	Linear	Nonlin	Double
BERT-base-CLS	.765	4.17	.625
MP-Net-base	.606	.963	.538
GTR-T5-base	.541	.844	.499
GTR-T5-large	.550	.898	.502
GTR-T5-xl	**.536**	**.775**	**.498**

Table 1: Best validation MSE loss of sentence embedding models on each SynNaMoN probe, normalized by chance-level MSE between embeddings

First, notice that GTR-T5-xl outperforms all the other models across all the SynNaMoN architectures. This seems to confirm our expectation that larger models should produce more compositional sentence embeddings. However, GTR-T5-xl only marginally outperforms other sizes of GTR-T5 (except on Nonlin, for which it does far better), suggesting that size actually isn't a significant factor in compositionality. The lower performance of GTR-T5-large further corroborates this, but considering its anomalously lower average embedding MSE, the issue requires more work. The fact that GTR-T5 models all display high compositionality despite variance in size suggests something about their architecture or training approach is important—perhaps the representational bottleneck.

All 3 GTR-T5 models perform better than MP-Net, which in turn outperforms BERT CLS. This first fact is slightly surprising considering that on standard sentence representation tasks (Reimers and Gurevych, 2019), MP-Net (63.30) marginally outperforms all GTR-T5 (base: 59.40, large: 62.38, xl: 62.88) models. Evaluation of these sentence embedding models on large-scale, human-interpretable compositionality tasks may reveal that GTR-T5 does indeed produce better compositional representations than MP-Net. Although BERT's

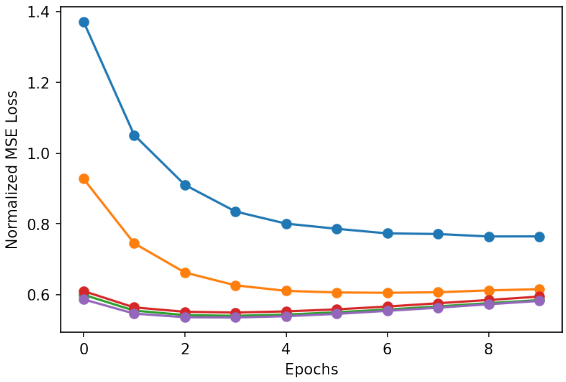

Figure 3: Normalized validation learning curves for Linear SynNaMoN on sentence embedding models (blue: BERT, orange: MP-Net, red: GTR-T5-large, green: GTR-T5-base, purple: GTR-T5-xl)

CLS token embedding is widely used for sentence representation, these results show that it fails to capture nearly as much compositional information as more targeted sentence embedding models.

Next, observe that although distilling to a Double SynNaMoN is intuitively easier than to a Linear SynNaMoN due to increased parameterization, there aren't always major improvements in distillability. It is possible that the geometric expressivity of the Double SynNaMoN will kick in with scaling of training data, but we hypothesize that this Double score will still approach a limit for all sentence embedding models. This is because syntax only describes a subset of sentence meaning, and the strictness of SynNaMoN's structure prevents this non-compositional component from being learned.

For example, a strictly syntactic compositional interpretation of "village on the river", would represent the village as being literally on top of the river since this is the semantic geometry learned for syntactic structures of the form "NP on NP". A SynNaMoN that includes non-linearities may better learn the geometry of this non-literal "on" relation, but a transformer model would best learn to handle non-compositional phrases due to its lack of strict syntactic constraints. Our broader takeaway from comparing Linear & Double scores is that much composition along syntactic lines is linear, and non-linearities in transformers primarily serve a purpose other than syntax-guided composition—perhaps in handling non-compositional phrases.

On a less theoretical note, we observe that our learning curves for Linear SynNaMoN on GTR-T5 (Fig. 3) are clearly overfitted due to fixing hyperparameters as mentioned in Sec. 3. We remediate this issue in Tab. 1 by reporting the best scores (minimum across epochs) for each learning curve. Since we want to construct the best possible SynNaMoN for a transformer model (as this most accurately reveals the transformer's distillable compositional ability), scores could be slightly improved with further hyperparameter tuning.

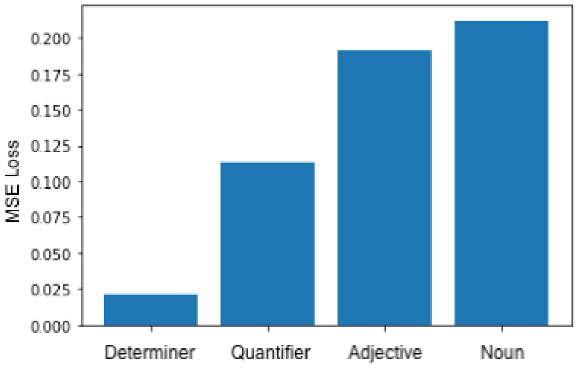

Figure 4: Generalization ability of Determiner Phrase module's linear geometry varies by part-of-speech

4.1 Analysis

Finally, we explore a single module to determine whether its compositional geometry meets intuitive notions of semantic generalization. Due to methodological difficulties with assessing a single module extracted from our end-to-end training paradigm, we train a Linear module for 'NP → Det N' on its own. Determiner-noun composition intuitively lies on a spectrum with adjective-noun composition on the other end and quantifier-noun composition in between. While quantifiers like 'some' and 'all' seem more like determiners, other quantifiers like 'several' and 'twelve' appear more comparable to adjectives like 'swarming' and 'grouped'. Intuitively then, we should expect the geometry of quantifier-noun composition to be intermediate to determiners and adjectives.

And this behavior is precisely what we find in our 'NP → Det N' module. Since it's trained on determiners, it obviously has the lowest MSE for this part-of-speech; we include noun-noun pairs (e.g. 'tree cow') as a control. As seen in Fig. 4, the module generalizes to quantifiers intermediately to determiners and adjectives. This demonstrates how SynNaMoN modules may enable interesting analyses of the compositional geometry of syntactic operations in sentence embedding models.

5 Conclusion

The human ability to apprehend the unitary meaning of a sentence corresponds to a neural model's ability to construct compositional sentence embeddings. In this work, we introduced Syntactic Neural Module Nets and used it in a distillation approach to assess how well syntax explains the sentential semantics computed by a transformer model. We showed that some models are more compositional by this metric, syntax-guided composition is largely linear, and modules learn composition functions that correspond to our semantic intuition.

Future work could explore this approach's alignment with other compositionality metrics and the non-compositional semantics left uncaptured by SynNaMoNs. We are also interested in how SynNaMoNs of different linguistic formalisms vary in distillability, as well as other potential use cases of SynNaMoNs beyond probing.

6 Limitations & Ethics Statement

Since longer sentences have more complex syntax, they require more modules on GPU and can run into out-of-memory issues. However, there is a hard upper-bound on total number of modules since there are limited syntax rules in the grammar. In addition, we may never need to train on long sentences if all modules can be effectively trained on short sentences and then generalize compositionally.

As an approach to probing language models, SynNaMoN contributes to an ethical NLP vision that seeks to address how models learn human biases that have societal effects from corpus data. Understanding syntax representations in models could be important in such a pursuit since some of these bias effects are syntactically mediated. For example, LMs with gender role biases could internally represent these biases as syntactic gender agreement e.g. ‘man’ agrees with ’doctor’ and ‘woman’ agrees with ‘nurse’ (Prates et al., 2020). By understanding the causal structure of sentential semantics in LMs, we can better disentangle syntax from spurious correlations transmitted by societal structures.

7 Acknowledgements

I am grateful for the advice provided by Tom McCoy, Emmy Liu, Chris Potts, and Paul Smolensky on questions of compositionality & probing. I also appreciate the intellectual company of Aryaman Arora & Justus Mattern at the onset of this project, and to Uncle Ike for providing us with a truly inspirational set of trees.

References

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016a. Learning to compose neural networks for question answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1545–1554.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016b. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Petra Barančíková and Ondřej Bojar. 2019. In search for linear relations in sentence embedding spaces. *arXiv preprint arXiv:1910.03375*.

Grzegorz Chrupała and Afra Alishahi. 2019. Correlating neural and symbolic representations of language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2952–2962.

Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. 2018. Using syntax to ground referring expressions in natural images. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Steffen Eger, Andreas Rücklé, and Iryna Gurevych. 2019. Pitfalls in the evaluation of sentence embeddings. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 55–60.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34:9574–9586.

Evan Hernandez and Jacob Andreas. 2021. The low-dimensional linear geometry of contextualized word representations. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 82–93.

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.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Nikita Kitaev, Thomas Lu, and Dan Klein. 2022. Learned incremental representations for parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3086–3095.

Katarzyna Krasnowska-Kieraś and Alina Wróblewska. 2019. Empirical linguistic study of sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5729–5739.

Ilia Kuznetsov and Iryna Gurevych. 2020. A matter of framing: The impact of linguistic formalism on probing results. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 171–182.

Mitch Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Rowan Hall Maudslay and Ryan Cotterell. 2021. Do syntactic probes probe syntax? experiments with jabberwocky probing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 124–131.

R Thomas McCoy, Tal Linzen, Ewan Dunbar, and Paul Smolensky. 2018. Rnns implicitly implement tensor product representations. *arXiv preprint arXiv:1812.08718*.

Richard Montague. 1970. English as a formal language.

Raymond J Mooney. 2014. Semantic parsing: Past, present, and future. In *Presentation slides from the ACL Workshop on Semantic Parsing*.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher D Manning. 2023. Characterizing intrinsic compositionality in transformers with tree projections. In *International Conference on Learning Representations*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Marcelo OR Prates, Pedro H Avelar, and Luís C Lamb. 2020. Assessing gender bias in machine translation: a case study with google translate. *Neural Computing and Applications*, 32(10):6363–6381.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of bert. *Advances in Neural Information Processing Systems*, 32.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

Paul Soulos, R Thomas McCoy, Tal Linzen, and Paul Smolensky. 2020. Discovering the compositional structure of vector representations with role learning networks. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 238–254.

Mycal Tucker, Peng Qian, and Roger Levy. 2021. What if this modified that? syntactic interventions with counterfactual embeddings. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 862–875, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Zhengxuan Wu, Atticus Geiger, Josh Rozner, Elisa Kreiss, Hanson Lu, Thomas Icard, Christopher Potts, and Noah D Goodman. 2021. Causal distillation for language models. *arXiv preprint arXiv:2112.02505*.

Xunjie Zhu and Gerard de Melo. 2020. Sentence analogies: Linguistic regularities in sentence embeddings. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3389–3400, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zining Zhu and Frank Rudzicz. 2020. An information theoretic view on selecting linguistic probes. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9251–9262, Online. Association for Computational Linguistics.