# AND does not mean OR: Using Formal Languages to Study Language Models' Representations

**Aaron Traylor**
Dept. of Computer Science
Brown University

**Roman Feiman**
Dept. of Cognitive, Linguistic, and Psychological Sciences
Brown University

**Ellie Pavlick**
Dept. of Computer Science
Brown University

{aaron_traylor, roman_feiman, ellie_pavlick}@brown.edu

## Abstract

A current open question in natural language processing is to what extent language models, which are trained with access only to the *form* of language, are able to capture the *meaning* of language. In many cases, meaning constrains form in consistent ways. This raises the possibility that some kinds of information about form might reflect meaning more transparently than others. The goal of this study is to investigate under what conditions we can expect meaning and form to covary sufficiently, such that a language model with access only to form might nonetheless succeed in emulating meaning. Focusing on propositional logic, we generate training corpora using a variety of motivated constraints, and measure a distributional language model's ability to differentiate logical symbols ($\neg$, $\wedge$, $\vee$). Our findings are largely negative: none of our simulated training corpora result in models which definitively differentiate meaningfully different symbols (e.g., $\wedge$ vs. $\vee$), suggesting a limitation to the types of semantic signals that current models are able to exploit.

## 1 Introduction

A current open question in natural language processing is to what extent language models (LMs; neural networks trained to predict the likelihood of word forms given textual context) are capable of truly understanding language. Bender and Koller (2020) argue that, since such models are trained exclusively on the *form* of language, they cannot possibly learn the *meaning* of language. We argue that the question of whether language models can learn meaning cannot be settled *a priori*. While language models only have direct access to form, linguistic form often correlates with meaning. The strength of the correlation varies across both different aspects of language and different tests of linguistic competence. While several intuitive tests of understanding (e.g., demonstrating knowledge of the word *dog* by identifying pictures of dogs) are out of scope for LMs, many tasks which NLP aspires to solve (e.g., question answering, machine translation) operate entirely on natural language input and output. Thus, a relevant question is whether models which operate only on the forms of language can nonetheless learn to differentiate meanings.

Our goal is to focus on a tractable subproblem in order to improve our intuitions about the types of distributional signals that LMs can use to extract information relevant to meaning. We simulate a language modeling setup using propositional logic, in which we can naturally operationalize *form* to be strings of symbols in the language and *meaning* to be truth conditions. We define the *semantic transparency* of a text-only training corpus to be the degree to which an LM trained on that corpus learns to differentiate between aspects of form that affect truth conditions and aspects of form that do not. We have two primary research questions. First, what constraints on corpus generation produce greater semantic transparency? And second, are any such constraints sufficient for an LM to adequately differentiate meanings?

## 2 Experimental Design

### 2.1 Dataset Generation

We consider the *form* of a sentence to be simply the observed, syntactically-valid strings of characters and the *meaning* to be the truth conditions. Propositional logic is a simple language in which we can characterize both form and meaning. We use the grammar in Table 1, with standard semantics.

We focus our analysis on whether the representations of logical operators ($\wedge$, $\vee$, $\neg$) are influenced by distributional patterns that go beyond their superficial syntactic similarity evident in the grammar. That is, if a trained LM identifies that the meanings



*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers)*, pages 158–167
August 1–6, 2021. 

| | |
|---|---|
| $S \rightarrow$ | $(S \wedge S) \mid (S \vee S) \mid (\neg S) \mid (\texttt{sym})$ |
| $\wedge \rightarrow$ | $\wedge_1 \mid \wedge_2 \cdots \mid \wedge_K$ |
| $\vee \rightarrow$ | $\vee_1 \mid \vee_2 \cdots \mid \vee_L$ |
| $\neg \rightarrow$ | $\neg_1 \mid \neg_2 \cdots \mid \neg_M$ |
| $\texttt{sym} \rightarrow$ | $\texttt{sym}_1 \mid \texttt{sym}_2 \cdots \mid \texttt{sym}_N$ |

Table 1: Propositional logic grammar.

of $\wedge_1 \cdots \wedge_k$ are identical to one another, and different from the meanings of $\vee_1 \cdots \vee_l$, we expect the embeddings for the $\wedge_i$ to be more similar to one another than they are to any of the $\vee_i$ or the $\neg_i$. We consider a corpus to be *semantically transparent* if an LM trained on the corpus learns semantically-clustered representations of the logical operators.

We generate four different training corpora, motivated by different assumptions one might make about how natural language corpora arise. These constraints are as follows, ordered roughly from weakest to strongest:

**1. Syntactic Constraint.** Speakers only generate sentences which are syntactically well-formed (that can be parsed by a syntactic parser). Here, this amounts to sampling from the grammar without additional constraints.

**2. Truthfulness Constraint.** Speakers of the language are constrained to generate sentences that are true in some context, i.e., that evaluate to `True` in at least one possible world. To implement this, we again sample from the grammar but additionally check with a satisfiability checker and omit sentences which are not satisfiable. E.g., $(\texttt{sym}_1 \wedge (\neg(\texttt{sym}_1)))$ would not appear.

**3. Informativity Constraint.** Speakers generate sentences not just to state true facts, but to provide listeners with information about a particular state of affairs. To simulate such a constraint, we randomly sample a set of "target worlds" $T$ and a set of "alternative worlds" $A$ such that $T \cap A = \emptyset$. We then generate the shortest sentence $s$ such that $s$ is true in every world in $T$ and $s$ is false in every world in $A$. We experiment with several sizes of T and A, but report only on $|T| = |A| = 2$ as this provides the right balance of contextual diversity. See Appendix for additional discussion.

**4. Explicit Grounding.** We consider a setting in which speakers explicitly dictate the full state of affairs, without ambiguity. This is not intended as a realistic model of how corpora are generated, but rather to provide an upper bound on semantic transparency by giving models a corpus in which form is perfectly correlated with meaning. We generate this corpus in the same way as the Truthfulness corpus, but append an explicit marker of the truth values[1] of the variables in the sentence, e.g.: $(\texttt{sym}_1 \wedge (\neg(\texttt{sym}_2)))$ `<sep>` $\texttt{sym}_1$ `T` $\texttt{sym}_2$ `F`.

**Sampling Parameters.** Each dataset consists of 100K training and 1K validation sentences. We set the number of non-reserved symbols (N in the above grammar) to 5,000, and the number of "synonyms" of each logical symbol (K,L,M) to be 5. Thus, a sentence in one of our datasets might look like $(\texttt{sym}_1 \wedge_3 (\neg_4(\texttt{sym}_{85})))$, and would be true if and only if $\texttt{sym}_1$ is true and $\texttt{sym}_{85}$ is false [2].

We generate sentences using a probabilistic context-free grammar with the rules shown above. The tree depth $d$ of a generated sentence is controlled by a parameter $\gamma$ such that $P(d|d-1) = \gamma^d$. The number of unique variables in a sentence[3] is sampled from a non-zero Poisson distribution parameterized by $\lambda$. We set $\lambda = 2$ and $\gamma = .85$ in the reported experiments, but don't find parameter choice affects our conclusions. Note that the Informativity dataset is generated deterministically, and thus sampling parameters do not apply and sentences in that dataset are shorter. Dataset statistics and data generation parameter sensitivity are in the Appendix.

### 2.2 Models and Training

We consider LSTM and Transformer LMs of differing sizes, shown in Table 2. Each model is trained on one of the above four datasets until convergence on the associated validation set using early stopping with a patience of 15 epochs. The LMs were implemented in PyTorch (Paszke et al., 2019) and took roughly 5 hours to converge on TitanV, TitanRTX, and QuadroRTX GPUs [4]. We randomly initialize the embedding layer. Hyperparameter details can be found in the Appendix. We train 5 random restarts of each setting. Due to the regular nature of our synthetic data, we found larger mod-

[1] Sampled from the set of satisfying variable assignments.

[2] We began by experimenting with many different dataset sizes and vocab counts. However, we did not find that models behaved differently on larger datasets and so focused on the smaller ones for convenience. See Appendix for results with different model sizes.

[3] We set a maximum number of variables per sentence in order to bound the number of possible variable assignments.

[4] Code publicly available at https://github.com/attraylor/semantic-transparency-code.

| Model | Syntactic | Truthfulness | Informativity | Grounded |
|---|---|---|---|---|
| Small LSTM (192K) | 21.2 / 87.7 / 87.7 | 17.6 / 88.7 / 88.6 | 21.5 / 99.6 / 99.5 | 21.2 / 87.5 / 87.5 |
| Medium LSTM (545K) | 17.6 / 90.2 / 90.1 | 17.5 / 89.6 / 89.5 | 20.9 / 99.9 / 99.8 | 8.3 / 89.3 / 86.8 |
| Small Trans. (311K) | 11.8 / 86.9 / 84.6 | 12.4 / 87.2 / 85.4 | 21.7 / 98.4 / 98.2 | 10.3 / 86.2 / 83.1 |
| Medium Trans. (377K) | 11.4 / 91.3 / 90.6 | 9.9 / 92.0 / 91.3 | 18.1 / 99.5 / 99.5 | 9.1 / 91.7 / 89.8 |

Table 2: Summary of language modeling performance. For each model, on each training dataset, we report **PPL / %Syn / %Sem** where PPL is the perplexity on heldout data (drawn from the same distribution as the training corpus), %Syn is the percentage of generated sentences that are syntactically well formed (i.e., parseable), estimated on a set of 1,000 generations sampled from the trained model, and % Sem is the percentage of generated sentences that are semantically well formed (i.e., satisfiable), estimated on the same set of 1,000.

els overfit the training data quickly, and thus focus on smaller models.

## 3 Results and Discussion

**Language Modeling Performance.** We first sanity check that the trained models indeed function as LMs before evaluating the lexical representations. We compute the models' perplexity on heldout data. However, since perplexity is not comparable across conditions (since each constraint leads to differently distributed corpora) we also sample 1,000 generated sentences from each model and compare by measuring whether the sentences are 1) syntactically well-formed (i.e., parseable) and 2) semantically well-formed (i.e., satisfiable). Even in the case of models trained with the Syntactic constraint, as seen in Table 2, most of the sentences produced are nonetheless satisfiable. We see no difference between the Syntactic, Truthfulness, and Explicit Grounding conditions on these metrics. (The Informativity numbers are likely higher due to the shorter sentences that result from that generative process.) The fact that models trained only on satisfiable sentences nonetheless generate sentences which do not abide by such constraints suggests the models fail to encode less overt distributional patterns, which depend, for example, on recognizing abstract relations such as "sameness" of symbols in order to recognize violations (e.g., (A $\wedge(\neg$ A)). The failure to capture such properties of the data even in this simplified setting might have negative implications for the models' ability to infer abstract semantic relationships from more complex natural language corpora.

**Representations of Logical Symbols.** Again, our first question is: What constraints on corpus generation yield the greatest amounts of semantic transparency? We quantify this by measuring how well the embeddings learned by the trained LMs correspond to our truth-theoretic notions of semantic equivalence: e.g., are $\wedge_1$ and $\wedge_2$ more similar to one another than $\wedge_1$ and $\vee_1$? We use a nearest neighbors probing classifier to evaluate whether models distinguish the operators at the lexical level. We run $k$-fold cross validation, in each iteration choosing one symbol per class (i.e., one $\wedge$, one $\vee$, one $\neg$) as the class exemplars, and then classifying the remaining points using cosine similarity. We set $k$ to 125, so that we observe every symbol combination as exemplars. We report accuracy averaged across folds and random restarts.

Probing classifier results are shown in Figure 1. Figure 2 shows an embedding visualization for one model (Medium Transformer). We find that training on the Syntactic and on the Explicit Grounding dataset leads to the least and the most distinguishable operators respectively for all models, and the other conditions end up between these values.

These results address our first question: there is some difference in semantic transparency between differently constrained datasets. Interestingly, the Transformer models perform better in the Truthfulness condition than in the Syntactic condition, which the LSTMs fail to differentiate. This suggests that, even if it does not necessarily manifest in the models' generations (Table 2), the Transformer architecture may nonetheless be capable of picking up on some of the more abstract distributional patterns via which syntax and semantics are correlated. Further work on larger models would be required to explore this in depth.

In addition, we observe little difference between the quality of the representations learned in the Informativity condition and those learned in the Truthfulness condition; one exception might be in the Medium LSTM, though we cannot confirm that this difference is robustly reproducible. Thus,

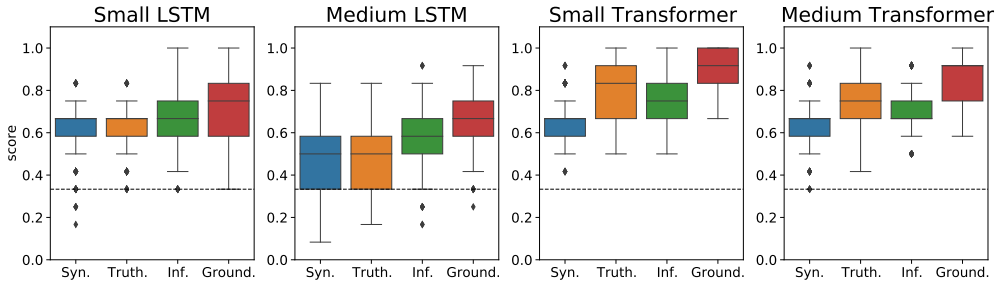


Figure 1: Each value in this graph represents average classification score across 125 iterations of a simple nearest neighbor probing classifier averaged across 5 random seeds of the model (625 accuracy numbers per box and whiskers plot). The dotted line is random chance / maximum class accuracy (33%).

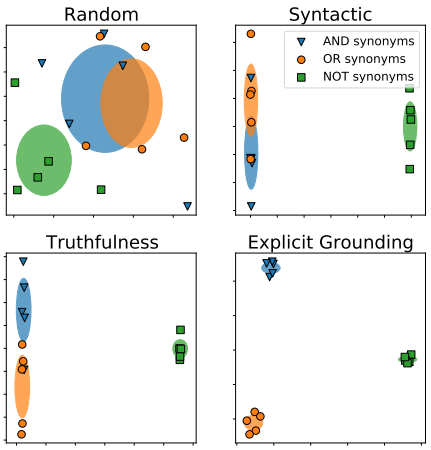


Figure 2: PCA of the representations created by the Medium Transformer model.

based on our experiments, there is no evidence that Informativity alone yields greater semantic transparency. However, we note that the experimental setup for Informativity is not directly comparable to the others (e.g., sentences are shorter and less diverse than in Truthfulness) and thus further study would be needed to make strong claims, positive or negative.

Finally, we note that in nearly all cases, models are able to differentiate $\neg$ from the other operators, likely because it is a unary operator and thus syntactically different from the binary operators. Thus the difference in accuracy is almost entirely due to whether the representations of $\wedge$ and $\vee$ are differentiated (as shown in Figure 2). This gives a negative answer to our second question concerning whether any constraints are sufficient for an LM to adequately differentiate meaning. Apart from the Small Transformer on the Explicit Grounding condition, none of the models can completely distinguish between symbols that are similar in form but different in meaning.

## 4 Related Work

It is an open question whether neural models can learn abstract functions (Marcus, 2001). Our work builds upon a large body of research intended to probe which aspects of language and meaning are being captured by large LMs. Most closely related is work that assesses whether models can perform symbolic reasoning about language (Kassner et al., 2020) e.g., quantifiers or negation (Talmor et al., 2020; Ettinger, 2020; Kassner and Schütze, 2020; Wang et al., 2018) or by measuring the systematicity of models' inferences (Goodwin et al., 2020; Kim and Linzen, 2020; Yanaka et al., 2020; Warstadt et al., 2019). Such work has tended to find that LMs reason primarily contextually as opposed to abstractly. Our evaluation method– which asks whether word embeddings cluster according to their truth-conditional meaning– is related to recent work which defines text-only models as "grounded" if the learned embedding space is isomorphic to the similarity function defined over a ground-truth meaning representation (Merrill et al., 2021). More distantly related is work on LMs' ability to reason about numbers (Wallace et al., 2019) or perform multi-hop reasoning (Yang et al., 2018). Prior work that examines neural networks' ability to perform logical reasoning is superficially related (Evans et al., 2018). In this way, our work builds on past work that uses synthetic rather than natural language datasets in order to probe model behavior in the absence of confounds. Notable examples are SCAN for measuring compositionality and generalization (Lake and Baroni, 2018) and Kassner et al. (2020) which investigates LM knowledge acquisition and fact memorization using a synthetic dataset of entity-relation tuples.

## 5 Conclusion

Using propositional logic corpora to simulate a controlled language modeling setting, we ask: 1) Do properties of the training corpus affect LMs' abilities to differentiate the meanings of logical operators? and 2) Do any training corpora lead to models that differentiate these meanings to a satisfactory degree? Our results imply a positive answer to (1): Models trained on corpora generated with different constraints appear to perform differently at the task of separating $\wedge$ from $\vee$. However, these differences are a function of both data and model. For example, the Transformer architecture seems better able to learn from weaker signal (corpora generated only with a Truthfulness constraint), while LSTMs require more explicit signal (direct access to truth values). On question (2), our results are largely negative for the syntactically similar operators. Even the most semantically transparent training data did not enable models to separate the representations of symbols with similar form but different meaning. Only the Small Transformer trained on the Explicit Grounding condition can perfectly differentiate $\wedge$ from $\vee$ at the lexical level, despite the task's controlled nature. However, every model did separate $\neg$ from both $\wedge$ and $\vee$, illustrating how syntactic differences can support differentiation of meaning.

Overall, we contribute a novel framework, based on syntax and semantics of propositional logic, via which we can explore questions of the linguistic capabilities and weaknesses of neural LMs. Our experiments represent a first step in this line of work, but further work is needed to fully appreciate the implications of these results in natural language settings, in particular, how closely the constraints explored here mirror real corpora, and how such learning is influenced by noise and ambiguity found in human language. One specific limitation of our experiments is that we constrain our analysis to the lexical representations– i.e., we assume that differences between the meanings of $\wedge$ and $\vee$ should be encoded in the lexicon, via context-invariant type embeddings. While this assumption is commonplace in formal semantics, neural LMs open the possibility of alternative representations of lexical and compositional semantics. Our results do not rule out the possibility that the relevant semantic distinctions are encoded elsewhere in the model, above the lexical layer. However, we take the combination of the lexical probing results and LM generation results as suggestive but not confirmational evidence of a more general negative finding.

## 6 Acknowledgements

We would like to thank Najoung Kim and the participants of the NALOMA 2020 Workshop for their thoughtful feedback on early versions of this work. This work was supported by IARPA under the BETTER program, contract number 19051600004.

## References

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Richard Evans, David Saxton, David Amos, Pushmeet Kohli, and Edward Grefenstette. 2018. Can neural networks understand logical entailment? In *International Conference on Learning Representations*.

Emily Goodwin, Koustuv Sinha, and Timothy J. O'Donnell. 2020. Probing linguistic systematicity. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1958–1969. Association for Computational Linguistics.

Nora Kassner, Benno Krojer, and Hinrich Schütze. 2020. Are pretrained language models symbolic reasoners over knowledge? In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818. Association for Computational Linguistics.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105. Association for Computational Linguistics.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2873–2882. PMLR.

Gary F Marcus. 2001. *The algebraic mind: Integrating connectionism and cognitive science*. MIT press.

Will Merrill, Yoav Goldberg, Roy Schwartz, and Noah A. Smith. 2021. Provable limitations of acquiring meaning from ungrounded form:what will future language models understand? *TACL*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5307–5315, Hong Kong, China. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019. Investigating BERT's knowledge of language: Five analysis methods with NPIs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2877–2887, Hong Kong, China. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. 2020. Do neural models learn systematicity of monotonicity inference in natural language? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6105–6117. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

# 7 Appendix

## 7.1 Dataset generation parameters

There are several parameters involved in the creation of our synthetic propositional logic datasets:

- Number of sentences in the training set
- Number of unique non-reserved variables (N)
- Number of each operator (K, L, M)
- Sentence depth parameter ($\gamma$)
- Poisson distribution parameter for unique non-reserved variables in sentence ($\lambda$)

In comparison to dataset sizes for large language models in modern natural language processing, the dataset size (100k training examples) and vocabulary size (5k symbols + 5 of each operator) of our main experimental results (Figure 1) are rather small. We sought to determine whether our choice for dataset size and non-restricted variable count greatly changed the final results– do our conclusions change based on these parameters? We trained models on different variations of our initial parameters.

First, we swept across training set sizes (20k, 100k, and 500k examples) and number of symbols (500, 5k, 50k) while holding all other parameters constant ($\gamma$ = .85, $\lambda$ = 2, K, L, M = 5). We used the Medium Transformer model, which performed the best across our four models, and observed the results of the probing classifier on the embeddings after training separately on each model.

The results of the above sweep are shown in Figure 3. We do not find that the models perform dramatically differently on any of the datasets when dataset size and number of non-reserved symbols are varied.

We also experimented with changing the number of operator synonyms (e.g. $\wedge_1, \wedge_2, ...\wedge_K$) We experimented with three different sizes– (K, L, M) = 5, 25, 100– for each of our 4 datasets. Those results are shown in Figure 5, and average frequency is shown in Table 3. We found that adding additional synonyms of each operator hurt performance– likely because adding additional synonyms of $\wedge$ and $\vee$ made generalization more challenging, causing the models' performance to drop.

In a set of earlier experiments, to choose the sentence depth ($\gamma$) and Poisson distribution ($\lambda$) parameters, we hyperparameter searched on the Explicit Grounding condition across three values of each (nine datasets in total). Specifically, we tested $\lambda = 2, 3, 5$ and $\gamma = .7, .8, .85$. We then trained the transformer model once on each of the nine datasets, and the results are shown in Figure 6. We chose $\lambda = 2$ and $\gamma = .85$.

| K, L, M | Syn. | Tru. | Inf. | Grd. |
|---|---|---|---|---|
| 5 | 49.7k | 49.2k | 16.3k | 49k |
| 25 | 9.94k | 9.84k | 3.25k | 9.81k |
| 100 | 2.49k | 2.46k | 0.81k | 2.45k |

Table 3: Average count of each operator across each of the datasets.

## 7.2 Informativity dataset information

We tested different settings of $|T|$ (number of target worlds) and $|A|$ (number of alternative worlds). For $|T| = 1, |A| = 1$, the best choice of $s$ will always be a single $\texttt{sym}$ or its negation. For example, with variables $\texttt{sym}_1, \texttt{sym}_2$, we might sample max variables = 2 and thus $T = (\texttt{sym}_1 = \texttt{T}, \texttt{sym}_2 = \texttt{F}), A = (\texttt{sym}_1 = \texttt{F}, \texttt{sym}_2 = \texttt{F})$. The shortest sentence would then be $\texttt{sym}_1$, as it sufficiently distinguishes $T$ from $A$. However, with $|T| = 1, |A| = 2$, we might generate $T = (\texttt{sym}_1 = \texttt{T}, \texttt{sym}_2 = \texttt{F}), A = ((\texttt{sym}_1 = \texttt{F}, \texttt{sym}_2 = \texttt{F}), (\texttt{sym}_1 = \texttt{T}, \texttt{sym}_2 = \texttt{T}))$. Now the shortest sentence that can be generated is $(\texttt{sym}_1 \wedge_1 \neg_1(\texttt{sym}_2))$.

$|T| = 1, |A| = 2$ and $|T| = 2, |A| = 1$ result in sentences that are both short and structurally nearly identical, although inverted. This is due to the truth conditions allowed by each operator. We generate the datasets for each combination and report the results in Table 4. We excluded these datasets because of the simplicity and similarity of the sentences. We found that $|T| = 2, |A| = 2$ allows for sentences that are much more varied.

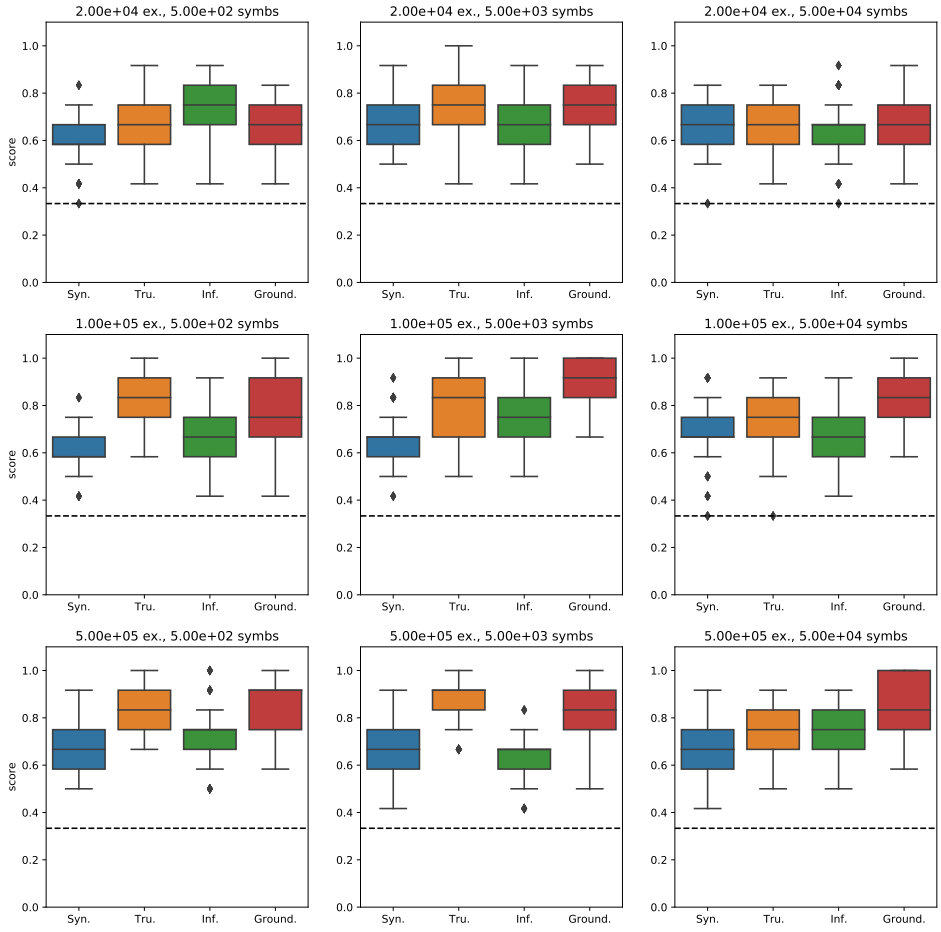


Figure 3: Average probing classifier score across example count / number of unique non-variable symbols for the Medium Transformer model.

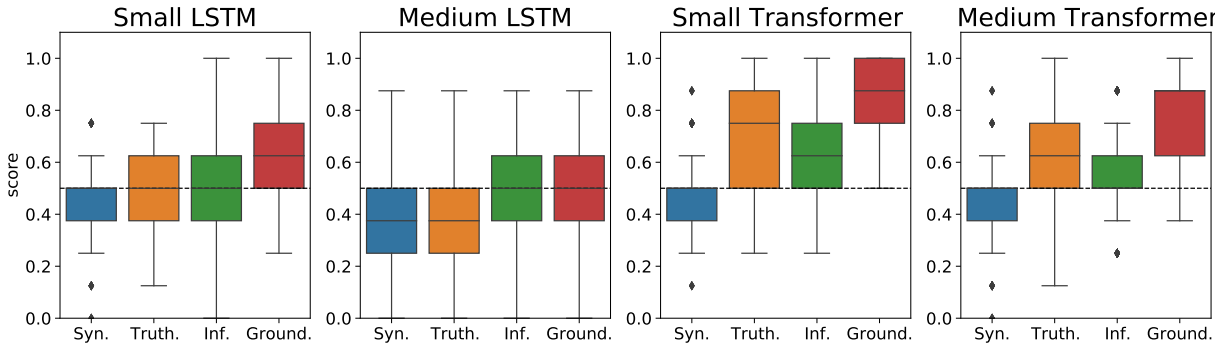


Figure 4: This graph contains the same experiments as Figure 1, but is only the accuracy on $\wedge$ and $\vee$, excluding the results of the negation operator.

| Inform. 1T/1A | | Inform. 1T/2A | | Inform. 2T/1A | |
|---|---|---|---|---|---|
| Sent. | Count | Sent. | Count | Sent. | Count |
| $a$ | 4523 | $(a \wedge b)$ | 27047 | $(a \vee b)$ | 27236 |
| $\neg(a)$ | 4460 | $(a \wedge \neg(b))$ | 21474 | $\neg((a \wedge b))$ | 21392 |
| | | $\neg((a \vee b))$ | 21338 | $(a \vee \neg(b))$ | 21260 |
| | | $(\neg(a) \wedge b)$ | 21061 | $(\neg(a) \vee b)$ | 21045 |
| | | $\neg(a)$ | 4544 | $a$ | 4559 |
| | | $a$ | 4536 | $\neg(a)$ | 4508 |

Table 4: All sentences generated for the first three Informativity datasets fell into one of these templates. Arbitrary symbols are replaced with $a$ and $b$. This distinction happens because of the truth conditions that are allowed by the $\wedge$ and $\vee$ operators.

| Dataset | Sent. Len. | Average `sym` count | Average op count | Average Unique `syms` |
|---|---|---|---|---|
| Syntactic | 28.51 | 6.19 | 7.44 | 2.27 |
| Truthfulness | 28.25 | 6.14 | 7.37 | 2.33 |
| Inform. (2T/2A) | 10.92 | 2.83 | 2.70 | 2.20 |
| Expl. Ground | 34.06 | 8.51 | 7.40 | 2.33 |

Table 5: Averaged statistics per sentence for the different datasets (training sets). All datasets are 100K training examples and 1k heldout examples.

| Model | LR | symb dim | hidden dim | # heads | # layers | dropout |
|---|---|---|---|---|---|---|
| Small LSTM | .0001 | 4 | 32 | | 1 | 0.0 |
| Medium LSTM | .0001 | 32 | 64 | | 2 | 0.2 |
| Small Transformer | .0001 | 4 | 32 | 2 | 4 | 0.0 |
| Medium Transformer | 5e-05 | 32 | 128 | 4 | 4 | 0.2 |

Table 6: Hyperparameters for each model.

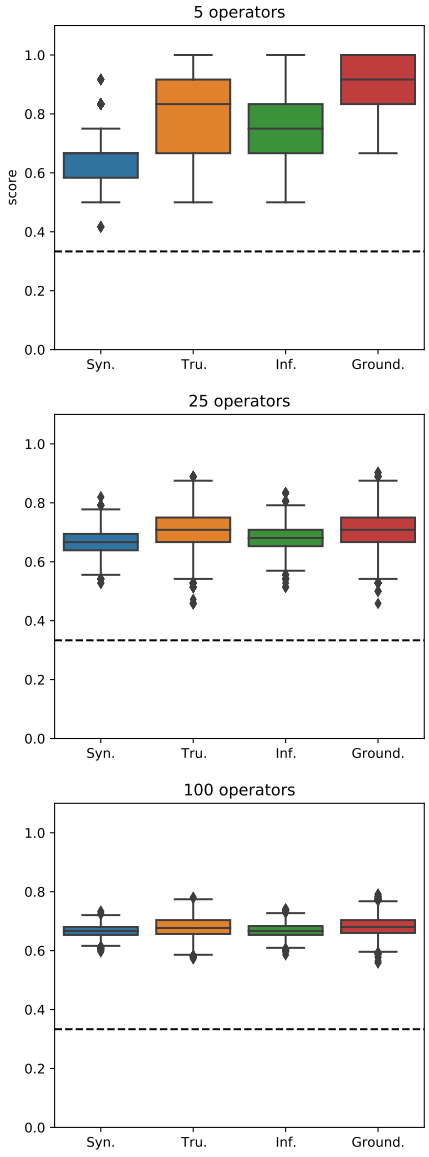


Figure 5: Sweep across number of operators using the Medium Transformer model.

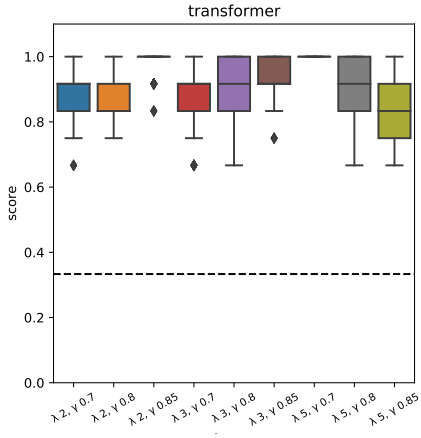


Figure 6: Sweep across $\lambda$ and $\gamma$ values for the Explicit Grounding dataset using a Transformer model.