Transparency Helps Reveal When Language Models Learn Meaning

Zhaofeng Wu[♋*] **William Merrill**[♏] **Hao Peng**[♌] **Iz Beltagy**[♌] **Noah A. Smith**[♌♍]
[♋]MIT [♏]New York University [♌]Allen Institute for Artificial Intelligence
[♍]Paul G. Allen School of Computer Science & Engineering, University of Washington
zfw@csail.mit.edu willm@nyu.edu {haop,beltagy,noah}@allenai.org

Abstract

Many current NLP systems are built from language models trained to optimize unsupervised objectives on large amounts of raw text. Under what conditions might such a procedure acquire meaning? Our systematic experiments with synthetic data reveal that, with languages where all expressions have context-*independent* denotations (i.e., languages with **strong transparency**), both autoregressive and masked language models successfully learn to emulate semantic relations between expressions. However, when denotations are changed to be context-*dependent* with the language otherwise unmodified, this ability degrades. Turning to natural language, our experiments with a specific phenomenon—referential opacity—add to the growing body of evidence that current language models do not represent natural language semantics well. We show this failure relates to the context-dependent nature of natural language form-meaning mappings.

1 Introduction

Despite language models' (LMs) centrality to recent progress on NLP benchmarks, a formal characterization of what can be learned from unsupervised training on large text corpora, and of what modern language models actually do learn, remains elusive. Empirically, Tenney et al. (2019), Kovaleva et al. (2019), Wu et al. (2021), *among others*, all discovered that pretrained LMs possess unsatisfactory semantic representations. Traylor et al. (2021) found co-variation between form and meaning to be insufficient for an LM to represent lexical semantics. Li et al. (2021), on the other hand, identified evidence of LMs representing dynamic semantics (Kamp, 1981; Heim, 1982; Groenendijk and Stokhof, 1991).

From first principles, Bender and Koller (2020) argued that it is a priori impossible for an ungrounded system that has access only to linguistic forms to learn the mapping between those forms and their grounded denotations. They claimed, as a thought experiment, that a learner that has access to all Java code (i.e., form) on GitHub can never learn execution (i.e., meaning). They nevertheless acknowledged that the existence of unit tests, which assert the expected output given input to blocks of code, could constitute a weak form of grounding which potentially enables the learning of meaning.

Formalizing this idea, Merrill et al. (2021) theoretically proved the possibility of learning (or more technically, emulating) semantic relations between expressions in a certain class of formal languages—those that are **strongly transparent** whose expressions have context-independent denotations—using an assertion oracle, analogous to the assertions in unit tests. In addition, with an example, they showed the existence of non-emulatable languages even with an assertion oracle.

Yet, the practical implications of these theoretical results have not been explored. While assertions enable the emulation of strongly transparent languages, it is unclear if existing LM architectures and objectives *achieve* emulation given training data with assertions. Furthermore, we do not know if natural language (NL) is similarly non-emulatable as Merrill et al.'s (2021) constructed example, especially since non-transparency does not always imply non-emulatability. We thus pose two research questions:

RQ1. Can current LM architectures and pretraining objectives emulate the meaning of strongly transparent languages?

*This work was done when Zhaofeng Wu was at AI2. Our code and trained models are released at https://github.com/ZhaofengWu/transparency.

Transactions of the Association for Computational Linguistics, vol. 11, pp. 617–634, 2023. https://doi.org/10.1162/tacl_a_00565
Action Editor: Marco Baroni. Submission batch: 12/2022; Revision batch: 2/2023; Published 6/2023.
© 2023 Association for Computational Linguistics. Distributed under a CC-BY 4.0 license.

RQ2. Can modern LMs fully emulate the meaning of natural language which is non-transparent?

We answer RQ1 in the positive (§3): On a strongly transparent propositional logic language, autoregressive and masked language models pretrained on only expressions (form), à la GPT-2 (Radford et al., 2019) and RoBERTa (Liu et al., 2019c), can consistently compare and evaluate their values (meaning). We find that necessary grounding of the pretraining data distribution is crucial to this ability. We also investigate the role of transparency for emulatability in a controlled setting as an intermediate study before analyzing non-transparent natural language. We ablate strong transparency from the logic language while keeping other factors unchanged. We observe a substantial drop in the LMs' ability to emulate meaning, highlighting the importance of transparency for emulatability.

We then turn to natural language (§4). Referential opacity is an extensively studied phenomenon in semantics and philosophy (Quine, 1956; Kripke, 1972, *among others*) but has not been examined in modern NLP. We prove that this phenomenon entails non-transparency and analyze how well existing LMs represent it. Our analyses based on probing and sentence similarity point to a lack of its representation in the largest GPT-2 and BERT (Devlin et al., 2019) models (RQ2). Theoretically, this is a natural language parallel to the emulation difficulty for our non-transparent formal language, and further reinforces the connection between transparency and meaning emulatability. Practically, through the lens of strong transparency, our results supplement prior studies that identified pretrained LMs' insufficient semantic representations (Tenney et al., 2019; Yu and Ettinger, 2020, 2021; Wu et al., 2021, *among others*).

2 Background

We follow Merrill et al.'s (2021) operationalization of the learning of meaning by emulation and their definition of strong transparency. We summarize their nomenclature and theoretical results in this section and provide some examples. We refer readers to Merrill et al. (2021) for more details.

At a high level, we take an *inferential* (Speaks, 2021, §2.2.3) view of meaning. An LM is taken to understand a language L if it can resolve semantic relations (e.g., equivalence) between expressions in L.[1] This is achieved through two procedures: μ_L maps expressions into representations based on training data from L, and δ uses the representations of two expressions to resolve a semantic relation between them.

2.1 Languages

We consider a **language** $L \subseteq \Sigma^*$ over an alphabet Σ and denote $(\Sigma^*)^2 = \Sigma^* \times \Sigma^*$. We term members of L **sentences**. We consider an **expression** $e \in \Sigma^*$ with associated left and right **context** $\kappa = \langle l, r \rangle \subseteq (\Sigma^*)^2$. $ler \in L$ is a sentence. We denote the empty string with λ and the empty context with λ^2.

Definition 1 (L_t)**.** We use the following context-free grammar (CFG) to specify a propositional logic language as a running example:

$$\begin{aligned} S &\rightarrow (e \wedge e) \mid (e \vee e) \mid (\neg e) \\ e &\rightarrow (e \wedge e) \mid (e \vee e) \mid (\neg e) \mid \mathsf{T} \mid \mathsf{F} \end{aligned} \tag{1}$$

S is the distinguished start symbol and T and F stand for True and False. We call this language L_t where t stands for ''transparent'' (see §2.5). It underlies our investigation in §3.

For example, the sentence $(((\neg\mathsf{T}) \vee \mathsf{F}) \vee (\neg\mathsf{T}))$ belongs to L_t because it can be generated by this CFG using the steps illustrated in Figure 1. In this sentence, the expression F has context $\langle(((\neg\mathsf{T})\vee\ ,\ ) \vee (\neg\mathsf{T}))\rangle$.

2.2 Meaning

We consider the denotation of an expression e, $[\![e \mid \kappa]\!]_L$, to be its meaning in the context κ.[2] We write $[\![e \mid \kappa]\!]_L = \varnothing$ if e is invalid in κ.

The meaning of a propositional logic expression can be the value derived from its conventional semantics, i.e., either T or F. For instance, $[\![(\mathsf{T} \wedge (\neg\mathsf{F}))|\lambda^2]\!]_{L_t} = \mathsf{T}$, and $[\![(\neg\mathsf{F})|\langle(\mathsf{T}\wedge\ ,\ )\rangle]\!]_{L_t} = \mathsf{T}$. For natural language, extensionally, the meaning of a sentence is its truth value, also either

[1] This inferentialist perspective can be contrasted with *denotationalism*, which says that ''understanding'' is the task of mapping an expression to a logical representation of its meaning (Speaks, 2021, §2.2.3). Inferentialism implicitly underlies natural language inference-based evaluation of NLP models (e.g., Bowman et al., 2015).

[2] We overload L to represent both the surface form and a mapping between form and denotation.

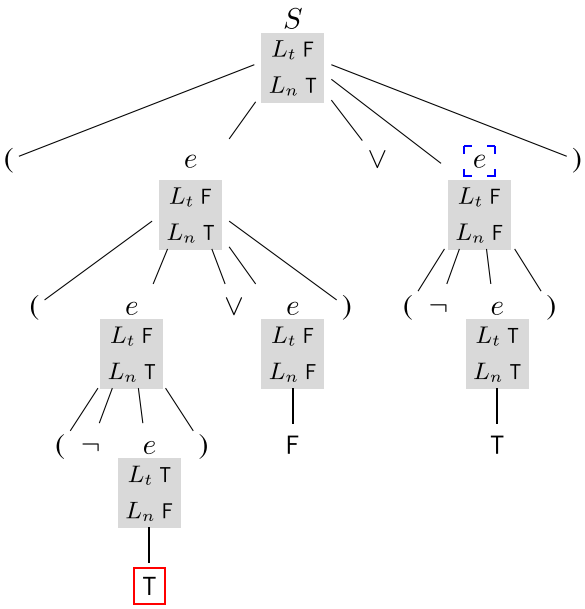

Figure 1: An example sentence in our propositional logic language as specified in Eq. 1. The e node c-commands the T node, inverting its meaning in L_n (§3.5). We mark the denotation of each node under L_t or L_n.

T or F (Frege, 1892); intensionally, the meaning is its truth condition, which could be viewed as a set of possible worlds where the sentence is true (Carnap, 1947). For a summary of the extension and intension of other expressions in NL, see Kearns (2011, §1.3). As an example in English, extensionally, $[\![$An author of this paper believes that Corgis are the cutest dogs.$|\lambda^2]\!] = \mathsf{T}$.

2.3 Assertion Oracle

To represent assertions in unit tests, Merrill et al. (2021) considered an assertion oracle which outputs if two expressions have the same denotation under the same context. Specifically, for expressions $e, e' \in \Sigma^*$ and $\kappa \in (\Sigma^*)^2$, the assertion oracle is defined as

$$\aleph_L(e, e' \mid \kappa) = \begin{cases} 1 & \text{if } [\![e|\kappa]\!]_L = [\![e'|\kappa]\!]_L \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

LM pretraining corpora could provide ℵ-like signals. For instance, pretraining sequences of the form e=e' are a natural analog to an ℵ query. We adopt this view to pretrain our propositional logic language in §3. In English and many other natural languages, copulas are a straightforward counterpart: ‘‘Corgis are the cutest dogs.’’ is equivalent to ‘‘Corgis=the cutest dogs.’’ This can be further reduced to all propositions: ‘‘Corgis run.’’ is equivalent to ℵ(Corgis run., T) under the extensional framework.[3]

2.4 ℵ-emulation: Learning Meaning

Merrill et al. (2021) say that a class of languages $\mathcal{L}$ is ℵ-emulatable if, intuitively, a learner μ_L with $\aleph_L$-access produces context-independent representations that allow another function δ to check the equivalence of any two expressions under any context without further $\aleph_L$-access. Formally, $\mathcal{L}$ is ℵ-emulatable if there exists an oracle Turing machine μ_L (that can query $\aleph_L$) and a standard Turing machine δ such that, for all $L \in \mathcal{L}$, context $\kappa \in (\Sigma^*)^2$, and valid expressions e, e' in κ,

$$[\![e|\kappa]\!]_L = [\![e'|\kappa]\!]_L \iff \delta\left(\mu_L(e), \mu_L(e') \mid \kappa\right) \quad (3)$$

Back to Corgis, an English learner μ can observe the equivalence of $e =$ ‘‘Corgis’’ and $e' =$ ‘‘the cutest dogs’’ in many different contexts κ and develop their representations. We say that natural language is emulated if there exists δ that can decide the equivalence between such expressions from the representations alone.

The standard pretraining-probing setup is an intuitive instantiation of μ_L and δ. A model μ_L can query $\aleph_L$ while pretraining on language L, which can then produce a representation $\mu_L(e)$ for any expression e. An equivalence probe δ can take the (frozen) representation of two expressions and decide their equivalence in some context. Importantly, because δ is frozen, it cannot make any more queries to $\aleph_L$. We adopt this paradigm for analysis in §3 and §4 and elaborate below.

2.5 Strong Transparency

Definition 2. A language L is **strongly transparent** if all of its expressions have context-independent denotations. That is, for all $e \in \Sigma^*$, $\kappa \in (\Sigma^*)^2$, either $[\![e|\kappa]\!]_L = [\![e|\lambda^2]\!]_L \neq \varnothing$ or $[\![e|\kappa]\!]_L = \varnothing$.

Under conventional propositional logic semantics, L_t (Def. 1) is strongly transparent because the value of every expression is determined by itself and unaffected by its context. Natural language, on the other hand, is non-transparent. We

[3] Assuming that propositions are more frequently true than false, which tends to be the case pragmatically (Grice, 1975).

prove in §4 that the NL phenomenon of referential opacity violates strong transparency.

Merrill et al. (2021) theoretically proved that all strongly transparent languages are ℵ-emulatable. In other words, it is possible to learn to emulate the meaning of these languages with only assertion oracle access. The converse is not necessarily true[4] and hence there may be a weaker condition than strong transparency that also entails ℵ-emulatability.

In what follows, we study how their theoretical results realize empirically. We examine in §3 if LM architectures and objectives can emulate the meaning of a strongly transparent language. In §4, we return to natural language which is non-transparent and thus Merrill et al.'s (2021) results do not predict its meaning emulatability.

3 How Well Do Language Models Fare?

While strongly transparent languages are in theory ℵ-emulatable, it is unknown if existing LM architectures, coupled with their pretraining objectives, are able to successfully achieve ℵ-emulation, or more intuitively, to learn their meaning.

To test this, we synthetically create a strongly transparent language based on propositional logic. We pretrain LMs with the same architecture and similar data scale as GPT-2 and RoBERTa on a generated pretraining corpus. We then train an equivalence probe to study if the pretrained representations enable ℵ-emulation. The probe is trained with a sentence pair binary classification objective and tested on unseen sentences sampled from the same grammar. Alternatively, we also try to directly evaluate the value of unseen sentences, without probe training. To isolate the effect of strong transparency, we also minimally perturb this language to be non-transparent and study how this affects emulatability.

3.1 Data

We use a PCFG to construct our propositional logic dataset because its recursive nature and context-freeness bear some resemblance to natural language,[5] and because it is convenient for sampling. The rules are specified in Eq. 1 and the probabilities are hand-designed. The denotation of an expression can be computed according to the conventional semantics of propositional logic, which, as argued in §2.5, makes L_t transparent. Figure 1 shows an example. See §A for more details.

Our CFG rules prevent the atomic sentences T and F from occurring in the corpus (and (T) and (F) too) and only allow compositional sentences. This ensures the absence of pretraining sequences like sentence=T and guarantees that there is no direct grounding to denotations during pretraining, but only indirect grounding via ℵ. This makes the task more difficult than the ℵ-emulation setup but more realistically transferable to natural language (§5).

The dataset has 819.2M pretraining sequences and 1M/10K/10K probe training/validation/test sentence pairs. All splits have disjoint sentences. The average sentence length is around 48.6. §A contains more details including tokenization.

3.2 Pretraining

We pretrain from scratch an autoregressive LM (**ALM**) and a masked LM (**MLM**), respectively simulating GPT-2-small and RoBERTa-base[6] with their original architecture, objective, and, to the extent possible, hyperparameters. They have near-identical model size hyperparameters, leading to 86.8M ALM parameters and 87.0M for MLM. We sample sentence pairs (a, b) with the same denotation and format the pretraining sequences in the form of a=b, such as (T∧F)=(F∨F), simulating ℵ-access (but restricting queries to be sentences, a more challenging setup: see Eq. 2). §3.3 will discuss a necessary form of data augmentation. We train for 100K steps, 20% of RoBERTa-base's training duration and hence data size, which we found sufficient for convergence on our data. §B summarizes hyperparameters.

3.3 Analysis: Probing L_t

Probing is a commonly adopted method to quantify the extent to which a representation encodes a particular type of linguistic information (Alain and Bengio, 2017; Liu et al., 2019a; Hewitt and

[4]Consider, for example, a finite non-transparent language whose denotation space can be learned by enumeration.

[5]There are aspects of natural language that a PCFG does not capture, such as recursion constraints (Karlsson, 2010) and non-context-free phenomena (Shieber, 1985). Nevertheless, the goal of this research question is not to maximally simulate NL, but rather investigate the distributional learnability of compositional semantics. Future work could investigate the effect of moving away from a strict PCFG.

[6]We do not follow BERT because next sentence prediction is not applicable here, but they are otherwise similar.

Manning, 2019, *among others*). The representation is frozen, on top of which a lightweight classifier is trained to predict the information of interest. As shown in §2.4, this paradigm conveniently corresponds to the formalization in Merrill et al. (2021), and hence we use it to investigate whether or not pretrained representations encode sufficient semantic information for equivalence decisions.

We probe semantic equivalence from the pretrained models for pairs of *unseen* sentences. We embed each sentence separately through the pretrained model, taking the last token representation for ALM and the average for MLM.[7] Voita et al. (2019) and Haviv et al. (2022) have shown that the positional information is diluted at the top transformer layers of MLMs, but it is crucial for the truth value in our language. We, therefore, take a weighted sum (a.k.a. scalar mix) of all layers for compensation for MLM.[8] We also found that these simple methods for sentence representations sometimes do not perform well. We hence additionally consider a variant where the probe is an attention-weighted mixture of all token positions. We refer to these two representations as –ATTN and +ATTN, respectively. See §B for more on their details. We train a bilinear classifier probe on top of the sentence representations (Li et al., 2021) and evaluate it with accuracy on a held-out test set. For each setting, we train the same probe with five different random seeds and report their mean and standard deviation. We report hyperparameters in §B.

Past work has cast doubt on whether probes faithfully reflect the representation's encoding of the information of interest, or if they directly learn the task (Hewitt and Liang, 2019). This is an especially important issue here as our +ATTN sentence representation injects additional trainable parameters compared to a simple (bi)linear classifier. To answer this question in our setting, we follow previous studies (Conneau et al., 2018; Tenney et al., 2019; Wu et al., 2021, *among others*) and train a randomly initialized and similarly frozen control model with the same architecture:

	ALM (à la GPT-2)		**MLM** (à la RoBERTa)	
	Random	**Trained**	**Random**	**Trained**
		Probing: –ATTN		
L_t	$49.9_{\pm 0.3}$	$98.8_{\pm 0.0}$	$50.0_{\pm 0.4}$	$50.1_{\pm 0.2}$
L_n	$50.0_{\pm 0.3}$	$79.9_{\pm 0.2}$	$49.9_{\pm 0.1}$	$49.5_{\pm 0.1}$
		Probing: +ATTN		
L_t	$49.9_{\pm 0.6}$	$100.0_{\pm 0.0}$	$50.0_{\pm 0.4}$	$63.8_{\pm 1.7}$
L_n	$50.1_{\pm 0.4}$	$82.5_{\pm 20.9}$	$50.2_{\pm 0.2}$	$49.7_{\pm 0.3}$
		Direct evaluation		
L_t	50.0	$97.0_{\pm 6.8}$	50.0	$95.4_{\pm 4.7}$
L_n	50.0	$91.1_{\pm 19.9}$	50.0	$50.4_{\pm 0.8}$

Table 1: Probing and direct evaluation accuracy (%) on random and pretrained models with autoregressive and masked LMs on our propositional logic test set. We report the results with both our transparent language L_t and the perturbed language L_n (§3.5). Probing checks the equivalence of two sentences, while direct evaluation computes the value of one sentence. For probing, we test two ways to obtain sentence representations, reporting the mean and standard deviation across five probe training seeds. For direct evaluation, we report the mean and standard deviation across our five templates.[9]

If LMs emulate meaning similarly to Merrill et al.'s (2021) algorithm, we would expect the pretrained model to yield higher probing accuracy than the random model.

Results. The L_t rows in the top two sections of Table 1 summarize the results. With a simple sentence representation (–ATTN), the pretrained ALM achieves near-perfect probing accuracy for L_t, though MLM performs at chance level. An attention-based sentence representation enables 63.8% accuracy[10] for MLM and improves ALM's performance to 100%. Importantly, in this variant, the random baselines still perform at chance level, demonstrating that the additional parameters do not lead to an overly powerful probe. We discuss the accuracy differences between ALM and MLM in §5. These results demonstrate that pretraining enables meaning emulation, though the meaning representation can be more deeply encoded than what can be extracted with a (bi)linear

[7] The lack of a next sentence prediction task (Fn. 6) leads to no supervision for a [CLS] token.

[8] Formally, μ's output contains all layer representations.

[9] ALM Trained +ATTN L_n has a degenerate seed that led to around 50% accuracy, hence the large variance. It is possible that additional configuration-specific hyperparameter tuning, which we did not perform, could reduce this instability.

[10] With additional linear layers, it could go up to $83.4\%_{\pm 2.0}$ while the random model still performs at chance level. We did not include this in Table 1 for consistency with other settings.

	–Reflexivity	+Reflexivity
–Symmetry	a=b $50.5_{\pm 0.4}$	a=b, a=a, b=b $92.7_{\pm 0.1}$
+Symmetry	a=b, b=a $50.3_{\pm 0.3}$	a=b, b=a, a=a, b=b $98.8_{\pm 0.0}$

Table 2: ALM probing accuracy (–ATTN; %) on our propositional logic test set with pretraining data with different properties, where a, b are expressions in L_t. We report the mean and standard deviation across five probe training seeds.

probe. We note that it is expected that the performance of pretrained models does not reach 100%. While Merrill et al. (2021) showed its theoretical possibility, their setup assumes active learning with unlimited access to $\aleph$ and allows the ''probe'' δ to be an arbitrarily powerful function, among other differences.

Grounding. We found that independently sampling pretraining sequences results in unsuccessful emulation with probing performance at random. Instead, it is crucial to ground = with reflexivity and symmetry.[11] We achieve this by augmenting the pretraining data: if a=b is a pretraining sequence, we ensure a=a, b=b (reflexivity), and b=a (symmetry) are too. This imposes a constraint on the pretraining data distribution that eases the learning of ='s meaning. Table 2 shows that both properties are important. We consider the implication in §5.

3.4 Analysis: Direct Evaluation on L_t

The process of training a probe introduces additional complexity, such as ±ATTN, that potentially complicates our analysis. Therefore, we also test a stronger condition where there is no additional classifier: Can the pretrained models *evaluate* expressions, without any further training (e.g., a probe)? For MLM, it is the most straightforward to compare if the model assigns a higher probability to T or F in sentence=[MASK]. However, this is a sequence that never occurs in the pretraining corpus since a standalone T or F is not part of our language (Eq. 1). Therefore, we use five templates on the right-hand side that are minimal in our language: (T∧[MASK]), (F∨[MASK]), ([MASK]∧T), ([MASK]∨F), (¬[MASK]). For the first four templates, we expect the masked position to be filled with the truth value of the proposition, and the negated value for the last one. For ALM, we compare if the model assigns a higher probability to the sequence where [MASK] is filled in with T vs. F.

Results. The bottom section of Table 1 shows the mean and standard deviation of the evaluation accuracy across our five templates. Without training, a random model always has 50.0% accuracy on expectation. Both ALM and MLM achieve a high evaluation accuracy, above 95%, corroborating the LMs' capability to represent the meaning of L_t.

These results respond to the argument in Bender and Koller (2020):

> We let GPT-2 complete the simple arithmetic problem *Three plus five equals*. The five responses below [...] show that this problem is beyond the current capability of GPT-2, and, we would argue, any pure LM.

We showed that form-only supervision *does* allow such evaluation on a strongly transparent language, at least when the supervising data distribution satisfies symmetry and reflexivity.

3.5 Non-transparency

Building towards non-transparent natural language, it is important to understand strong transparency's effect on emulatability. We design a minimally perturbed version of L_t that is non-transparent, L_n. The syntax stays the same, but we change the semantics such that ¬ has a side effect: When followed by T or F, it inverts the meaning of these literals that occur in certain other environments. Specifically, each (¬T) node changes the meaning of all the literals T in its c-commanded subtree (i.e., the e subtree headed by the (¬T) node's sibling, if there is one; Reinhart, 1976) to F. An additional (¬T) does not invert back. Similarly, (¬F) changes the meaning of the literal F to T. For example, in the sentence in Figure 1, the T node is c-commanded by (or, a descendant of a sibling of) the $e \rightarrow$ (¬T) node, so its meaning is changed to F. On the other hand, the $e \rightarrow$ (¬T) node does

[11] Reflexivity states that $a = a$, and symmetry $a = b \Rightarrow b = a$. Equality further requires transitivity: $a = b \wedge b = c \Rightarrow a = c$, but it is not tested in our probing setup and we found it unimportant for probing accuracy in preliminary experiments.

not invert the meaning of the unboxed T because they do not constitute a c-command relation. This alternation is inspired by binding theory in generative grammar (Chomsky, 1981, 1983), where the $(\neg\mathsf{T})$ node is the binder that c-commands the bindee. ince the meaning of T and F now depends on the existence of a binder, L_n is non-transparent.[12]

Results. We conduct the same pretraining/ probing/direct evaluation procedure on L_n. Table 1 reports the results. on-transparency decreases ALM's probing accuracy with both –ATTN and +ATTN, though not to random level. The variance across different probe training seeds also increases compared to L_t, indicating that the pretrained representation is less robust. Directly evaluating ALM with L_n similarly leads to both decreased average accuracy and increased variance. MLM, on the other hand, achieves random probing and evaluation accuracy. Overall, the lack of strong transparency reduces models' meaning emulation ability, though not always to chance performance.

4 What About Natural Language?

While existing LM architectures and objectives are able to emulate the meaning of synthetic languages, it is unclear how these observations transfer to natural language (NL). Merrill et al. (2021) hinted that, since NL is non-transparent and likely more complex than their constructed non-emulatable language, it is probable that a pretraining procedure, even with $\aleph$-access, cannot emulate its meaning either. This, however, remained an untested hypothesis.

We formalize this intuition and prove that a specific NL phenomenon, referential opacity, makes NL non-transparent.[13] This phenomenon has been widely studied in semantics (Quine, 1956; Kripke, 1972, *among others*), yet it has received little attention in modern NLP. We fill this gap from the perspective of strong transparency and study the representation of this phenomenon in modern LMs with a probing-based and a sentence similarity-based analysis.

4.1 Referential Opacity

To illustrate referential opacity, we use the classic example in semantics:

Example 1.

(a) Lois Lane believes Superman is a hero.

(b) Lois Lane believes Clark Kent is a hero.

Note that (a) and (b) have different truth conditions: Their truth values differ if Lois Lane does not know Superman and Clark Kent are the same person. Formally, $[\![$Lois Lane believes Superman is a hero.$|\lambda^2]\!] \neq [\![$Lois Lane believes Clark Kent is a hero.$|\lambda^2]\!]$.[14] On the other hand, $[\![$Superman$|\lambda^2]\!] = [\![$Clark Kent$|\lambda^2]\!]$.[15] In other words, two expressions that have the same denotation, when embedded in the same context, yield sentences with different truth conditions. Such contexts are called **referentially opaque**, and, in this case, they are induced by a propositional attitude verb ''believes'' whose meaning depends on the cognitive state of its subject (Anderson and Owens, 1990).

Now we formalize referential opacity:

Definition 3. In natural language, an expression e is contextually valid in $\kappa = \langle l, r\rangle$ if none of $[\![l|\lambda, er]\!], [\![e|l, r]\!], [\![r|le, \lambda]\!]$ is $\varnothing$.[16]

Definition 4. A context $\kappa = \langle l, r\rangle$ in natural language is **referentially opaque** if there exist expressions e_1, e_2, both contextually valid in κ, such that $[\![e_1|\lambda^2]\!] = [\![e_2|\lambda^2]\!]$ and $[\![le_1r|\lambda^2]\!] \neq [\![le_2r|\lambda^2]\!]$.

Def. 4 matches the linguistic phenomenon: Let e_1=''Superman'', e_2=''Clark Kent'', and the opaque context κ=$\langle$''Lois Lane believes'', ''is a hero.''$\rangle$, and we recover our analysis of Ex. 1 above.

[12] This is a straightforward way to introduce a $\neg$ with side effect to a hierarchical structure. An alternative is to rely on a linear structure and invert all literals linearly following $\neg$. Nevertheless, our version leverages the hierarchical reasoning that the model originally needs to possess to evaluate an expression, while this version requires a new type of reasoning that is linear. So that change would be less minimal.

[13] Deictic expressions are another example, though they have been extensively studied under coreference resolution.

[14] In this section we consider the language L to be English, or any NL that exhibits this phenomenon, and $[\![\cdot|\cdot]\!]$ to be intensions (§2.2). We drop the subscript L for brevity.

[15] It is possible to argue that $[\![$Superman$|\lambda^2]\!] \neq [\![$Clark Kent$|\lambda^2]\!]$ if we consider their intension to be different. Nevertheless, we adopt the view of Heim and Kratzer (1998, §12.3) to not introduce intensionality by default (i.e., with $\kappa = \lambda^2$), but rather to evoke it by context: ''The usual denotations are extensions. But for nonextensional contexts, Intensional Functional Application allows a switch to intensions. The switch is triggered by particular lexical items [...]''.

[16] This is a technical detail needed for proving Theorem 1.

Now, we prove that the existence of referentially opaque contexts implies non-transparency. We assume compositionality, for which we provide a working definition: $[\![ler|\lambda^2]\!] = f([\![l|\lambda, er]\!], [\![e|l, r]\!], [\![r|le, \lambda]\!])$ for some meaning composition function f.[17] Intuitively, the proof shows that if all expressions have fixed meaning (i.e., are strongly transparent), referential opacity would not arise.

Theorem 1. *A compositional language with referentially opaque contexts is not strongly transparent.*

Proof. Suppose by contradiction we have such a language L that is strongly transparent. Let e_1, e_2 be expressions in some opaque context $\langle l, r\rangle$ in L.

$$
\begin{aligned}
[\![le_1r|\lambda^2]\!] = \; & f\left([\![l|\lambda, e_1r]\!], [\![e_1|l, r]\!], [\![r|le_1, \lambda]\!]\right) \\
& \text{By compositionality} \\
= \; & f\left([\![l|\lambda, e_2r]\!], [\![e_1|\lambda^2]\!], [\![r|le_2, \lambda]\!]\right) \\
& \text{By strong transparency} \\
= \; & f\left([\![l|\lambda, e_2r]\!], [\![e_2|\lambda^2]\!], [\![r|le_2, \lambda]\!]\right) \\
& \text{By referential opacity premise} \\
= \; & f\left([\![l|\lambda, e_2r]\!], [\![e_2|l, r]\!], [\![r|le_2, \lambda]\!]\right) \\
& \text{By strong transparency} \\
= \; & [\![le_2r|\lambda^2]\!] \\
& \text{By compositionality}
\end{aligned}
$$

This violates $[\![le_1r|\lambda^2]\!] \neq [\![le_2r|\lambda^2]\!]$, the referential opacity premise. So L is not strongly transparent. □

Therefore, as a non-transparent example in NL, we study whether referential opacity is reflected in the representation of current LMs.

4.2 Data

We cast referential opacity as a sentence pair binary classification problem. We generate sentence pairs like Ex. 1 as our dataset. Ex. 1 consists of two parts that correspond to the two conditions in Def. 4: two co-referring expressions ($[\![e_1|\lambda^2]\!] = [\![e_2|\lambda^2]\!]$), and a referentially opaque context that embeds the entity ($[\![le_1r|\lambda^2]\!] \neq [\![le_2r|\lambda^2]\!]$). Next, we separately introduce how we generate them. Our final dataset consists of 45K/6K/6K training/development/testing sentence pairs for GPT-2 and 97K/12K/12K for BERT. §C provides more details, including more fine-grained dataset statistics for different experimental settings below.

Co-referring Expressions. The co-referring expressions in Ex. 1 are proper names, ‘‘Superman’’ and ‘‘Clark Kent.’’ Not only is this hard to collect data for, but, due to the rigidity of proper names (Kripke, 1972), it is also theoretically more challenging to analyze as the classic intensionality framework is more difficult to apply (Von Fintel and Heim, 2011).[18] We hence consider co-referring expressions that are one proper name and one definite description, such as ‘‘Yuri Gagarin’’ and ‘‘the first person in space,’’ which can be more straightforwardly accounted for with intensionality (Heim and Kratzer, 1998, §12; Von Fintel and Heim, 2011). We use the LAMA dataset (Petroni et al., 2019), specifically the T-REx split (Elsahar et al., 2018) following recent factual probing work (Jiang et al., 2020; Shin et al., 2020; Zhong et al., 2021), to obtain a list of such entities. To make sure the model representation captures the coreference, we follow Petroni et al. (2019) and use LAMA to prompt the LM with these equivalences and only keep entities that are correctly predicted.[19]

Contexts. We construct referentially opaque and referentially transparent contexts to embed these co-referring expressions. We only consider referential opacity involving propositional attitude verbs, where the context is referentially opaque iff its main verb conveys propositional attitude. There are other types of referential opacity, such as counterfactuals (Von Fintel and Heim, 2011; Kearns, 2011, §7) and substitutions that shift the syntactic status of constituents (e.g., Fine, 1990), that we omit in this work for simplicity, though they could be targets of future studies. We manually design two classes of templates, depending on the verb's argument structure. The first has an embedded clause, e.g.,

Example 2. Label = non-equivalent[20]

[17] This is a mild assumption, considering the generality of compositionality (Fodor and Pylyshyn, 1988) and that our definition is weak, e.g., weaker than that of Andreas's (2019).

[18] Though see Shabasson (2018) for a theorization.

[19] Previous work (Poerner et al., 2020; Dufter et al., 2021; Cao et al., 2021) questioned whether such prompting measures model ‘‘understanding.’’ Our setup, though, does not depend on ‘‘understanding’’, but only requires association.

[20] Consider, for example, if this person is Yuri's neighbor and wants to meet him for dinner, but, being an avid flat-earther, is not fond of space traveling and is unaware that he has been to space. She would say she wants to meet Yuri Gagarin but has no interest in meeting the first person in space.

(a) She wants to meet Yuri Gagarin.

(b) She wants to meet the first person in space.

The second contains only the main clause, such as

Example 3. Label = equivalent

(a) He speaks Lao.

(b) He speaks the official language of Laos.

The two sentences in a pair only differ by the entity reference: one is a name and one is a definite description. A sentence pair is non-equivalent iff it has a referentially opaque context, or within our scope of study, iff its main verb is a propositional attitude verb. We gather the list of verbs from past linguistic studies and verify with native speaker judgment (see §C).

4.3 Models

We consider GPT-2-XL and BERT-large-cased[21], the largest variants in these two families, as representative autoregressive and masked LMs. They have 1.5B and 340M parameters, respectively. We obtain sentence representations in the same way as in §3, except without attention-weighting and simply using the [CLS] embedding for BERT.

4.4 Analysis: Probing

We use the same bilinear probe in §3 as a binary classifier over sentence pairs, determining the equivalence, or the referential transparency, of each pair. However, because of the lexicalized nature of referential opacity, the probe could easily overfit and recognize not their equivalence but the existence of a propositional attitude verb.

To overcome this, we introduce attractors (Linzen et al., 2016; Gulordava et al., 2018; Pandia and Ettinger, 2021, *among others*).[22] We always conjoin a clause with a propositional attitude verb and one with a non-attitude verb, disallowing the aforementioned heuristics. The equivalence label now depends on if the entity alternation occurs under the non-attitude verb, which would result in an equivalent sentence pair, or the attitude verb, which would lead to non-equivalence. For example:

		GPT-2	BERT
Simple	Equiv.	$100.0_{\pm 0.00}$	$100.0_{\pm 0.00}$
	Non-equiv.	$100.0_{\pm 0.00}$	$100.0_{\pm 0.00}$
	Overall	$100.0_{\pm 0.00}$	$100.0_{\pm 0.00}$
Coord.	Equiv.	$85.3_{\pm 0.03}$	$72.4_{\pm 0.03}$
	Non-equiv.	$15.5_{\pm 0.03}$	$29.6_{\pm 0.03}$
	Overall	$50.4_{\pm 0.00}$	$51.0_{\pm 0.00}$

Table 3: Probing accuracy (%) for referential opacity on GPT-2-XL and BERT-large-cased. We report the mean and standard deviation across 10 seeds. We consider two types of sentences, simple sentences without attractors and coordinated sentences with attractors. For each type, we show both the label-specific accuracy (Equivalent/Non-equivalent) and the overall accuracy.

Example 4. Label = equivalent

(a) He speaks Lao and she wants to meet Yuri Gagarin.

(b) He speaks the official language of Laos and she wants to meet Yuri Gagarin.

Example 5. Label = non-equivalent

(a) He speaks Lao and she wants to meet Yuri Gagarin.

(b) He speaks Lao and she wants to meet the first person in space.

Despite both examples having the same verbs, the sentence pair in Ex. 4 is equivalent, but Ex. 5 is not. We are not using attractors for out-of-domain evaluation; instead, the training and test sets are i.i.d., but we break down the test set performance by categories.

We train a probe on GPT-2-XL and BERT-large over 10 random seeds. Details are in §D. Table 3 reports the results. As expected, both models overfit with the attractor-less simple sentences, achieving perfect accuracy. With attractors in coordinated sentences, however, both models obtain near-random performance overall. Because the training and test sets are i.i.d., this means that semantic equivalence based on referential opacity cannot be probed in our setup from these two models, suggesting an inadequate representation

[21] Not RoBERTa as in §3, because BERT's [CLS] token can act as and is commonly taken to be the sentence representation (Devlin et al., 2019; Karpukhin et al., 2020, *among others*).

[22] Another option is to have disjoint training and testing verbs. This did not work in preliminary experiments because verbs that induce referential opacity are semantically closer, as they always convey propositional attitude. So the model could use this similarity in the word embedding space to extrapolate.

of this phenomenon.[23] Interestingly, both models tend to predict equivalence more than non-equivalence (more prominent with GPT-2 than BERT), likely due to the nuanced nature of this task: Without training, a human would likely judge equivalence on referentially opaque sentence pairs too.[24] See §E for a set of experiments that show that LMs can potentially learn to capture referential opacity with semantic supervision following pretraining.

4.5 Analysis: Sentence Similarity

As in §3.4, the simplicity of a training-free analysis can be desirable. To this end, we directly measure the cosine similarity between the two sentence representations in a pair. While this semantic similarity would be high for both groups of sentences by our construction, equivalent sentence pairs should have more similar representations than those that are not. While factors other than semantics, such as syntax, also affect sentence representations, we strictly control them in our synthetic data generation to be identical between referentially transparent and opaque sentences. We do not consider attractor sentences (§4.4) in this analysis.

For significance testing, we employ an exact permutation test (Fisher, 1935) and a bootstrap test (Efron and Tibshirani, 1993) with 1,000 iterations, performed across verbs, where the test statistic is the difference between the averaged cosine similarity of the two groups. Both tests are two-sided with the null hypothesis being that the model representation does not distinguish between the two classes of verbs. For GPT-2-XL, the permutation test gives $p = 0.64$ and bootstrap gives $p = 0.66$, barring us from rejecting the null hypothesis. For BERT-large, they give $p = 0.45$ and $p = 0.57$ respectively, where we again observe no significant difference between the two classes. Nonetheless, we note that the inability to reject the null hypothesis does not entail it is true.

Reimers and Gurevych (2019) noted that computing sentence pair cosine similarity using BERT's [CLS] token, as we did, does not correlate well with textual similarity benchmarks. This phenomenon is commonly attributed to the anisotropic nature of pretrained representations (Ethayarajh, 2019). This does not undermine the validity of our method, which instead relies on the correlation between the cosine similarity and *the model's representation of semantic closeness*. We ensure this correlation by controlling for all factors other than semantics (syntax, lexical choices, entities, etc.). Nevertheless, we also postprocess BERT's [CLS] representation using BERT-flow (Li et al., 2020) which has been shown to increase the correlation with textual similarity benchmarks. We obtain a similar result: Bootstrap gives $p = 0.49$. While the two-sided permutation test gives $p = 0.03$ with potential significance, the one-sided version gives $p = 0.99$; in other words, the calibrated space represents opaque sentence pairs to be more similar than transparent ones, contrary to our expectation that equivalent sentence pairs should be closer in the representation space than non-equivalent ones when all other factors are controlled.

The results from these two sets of analyses in §4.4 and §4.5 are consistent and show no evidence of modern LMs representing referential opacity, demonstrating that they cannot fully emulate the meaning of NL. Our finding adds to recent observations that pretrained LMs do not represent semantic phenomena well (Tenney et al., 2019; Kovaleva et al., 2019; Wu et al., 2021, *among others*). Theoretically, it also strengthens the connection between strong transparency and meaning emulatability with NL-based empirical evidence.

5 Discussion

Through analyses based on probing and direct evaluation, we have seen that existing LM architectures and objectives can learn to emulate the meaning of a strongly transparent language L_t when the training data reflects equivalence relations. While non-transparency (L_n) causes this ability to decrease, the trained models still outperform a random model in certain setups. We believe this result hints at the strength of current LM architectures and objectives.[25] There seems to be a limit to this strength, though—in natural

[23] There might still be other more complex heuristics, but even so, the probe still fails. Hence we do not need additional attractors to rule out all possible heuristics.

[24] Though, with training, it is relatively straightforward to perform this task for a human, so it is reasonable to test the ability in LMs.

[25] Especially since our setting is more challenging than Merrill et al.'s (2021) algorithm, without their unlimited $\aleph$-access, active learning, arbitrarily powerful δ, etc. Plus, we restrict $\aleph$ queries to be sentences and disallow comparing a sentence with $\mathtt{T}$ or $\mathtt{F}$ using $\aleph$.

language, neither GPT-2 nor BERT represents the non-transparent phenomenon of referential opacity well.

Our results shed light on the relationship between the strong transparency of a language and whether its semantics can be emulated. We observed co-variation between the two: When slightly perturbed to be non-transparent, our logic language becomes harder to emulate; and there is no evidence for LMs representing the semantics of a non-transparent NL phenomenon. Nevertheless, the above-random emulation performance with L_n suggests that there could be language properties that potentially better predict emulatability, leaving room for future theoretical endeavors.

We also found that, with a similar size and training procedure (§3.2), ALM is more suitable for representing the meaning of our propositional logic languages than MLM, in our setup. ALM achieves better probing accuracy than MLM under both methods of obtaining sentence representations that we explored. Also, MLM completely fails to emulate meaning facing non-transparency, but not ALM. Ultimately, though, we hope to understand if this difference transfers to natural language. Our NL investigation reveals that both ALM (GPT-2) and MLM (BERT) achieve chance-level probing performance on the one phenomenon that we inspected, likely due to its difficulty. It would be interesting for future efforts to further examine their differences, if any, in learning and representing the meaning of other NL phenomena.

Our results also lead to the question: Why can LMs achieve above-random results on L_n but not referential opacity? While it is entirely possible that the latter is simply more difficult than our synthetic non-transparency, there are other factors at play. First of all, natural language is much more variable than our synthetic language: Utterances can be untruthful (though they are in general governed by Gricean quality; Grice, 1975), subjective (such as our earlier claim about Corgis' cuteness, §2.3), intensional (see Merrill et al., 2021 for a discussion), etc. But putting these variations aside, we saw from §3 that even the synthetic language requires an explicit grounding of = to enable emulation, and this is missing from NL pretraining. It is certainly not the case that, for every expression such as ''Corgis are the cutest dogs.'' that exists in the pretraining corpus, the variations ''The cutest dogs are Corgis.'', ''Corgis are Corgis.'', ''The cutest dogs are the cutest dogs.'' are also guaranteed to appear. So perhaps there needs to be a more foundational change in our pretraining objective. As Brown et al. (2020) foretold, ''A more fundamental limitation of [...] scaling up any LM-like model [...] is that it may eventually run into (or could already be running into) the limits of the pretraining objective.'' Our results point to one such possibility: We believe research into a more explicit representation of semantic relations in future pretraining processes, such as based on paraphrases, could be fruitful.

What we did not investigate, though, is whether partial equivalence grounding enables emulation: what if, for example, only 1% of the pretraining data has this form of grounding, while the rest does not? And the above format already exists for certain sentences in NL. This, too, could be an exciting future research question.

6 Related Work

Bender and Koller (2020) initiated the discussion on the possibility of a learner acquiring meaning from training on linguistic forms alone. From first principles, they argued for its impossibility. Empirically, Traylor et al. (2021) also found that LMs cannot well-represent lexical-level symbols when the pretraining data is distributionally constrained to supply relevant signals. Merrill et al. (2021), on the other hand, proved theoretically that it is possible to emulate the meaning of strongly transparent languages with assertion oracle access. We showed in this work that, empirically, LMs also attain the capability. The work of Patel and Pavlick (2022) is also conceptually similar to our work, discovering that the internal representation of LMs is to a large extent isomorphic to the conceptual spaces of directions and colors. They adopted in-context learning (Brown et al., 2020; *among others*) to elicit the isomorphism, while we used the more traditional probing paradigm.

Another line of work has inspected the extent to which pretrained LMs encode various types of semantic information. Some have examined the representation of lexical semantics: Garí Soler and Apidianaki (2021) found that BERT representations reflect polysemy levels, and Vulić et al. (2020) showed that they also capture abundant type-level lexical knowledge. On the other hand, Ettinger (2020) and Ravichander et al. (2020) have discovered that pretrained LMs do not

satisfactorily encode negation and hypernymy, respectively. Moving beyond the lexical level, Wu et al. (2021) demonstrated that pretrained BERT and RoBERTa models less readily surface semantic dependency information than syntactic dependencies, while Li et al. (2021) identified evidence of dynamic semantics representation in these models.

7 Conclusion

We have empirically shown that pretrained language models are able to emulate the meaning of a strongly transparent language through pretraining on an assertion-inspired format, but this ability deteriorates when the language is minimally perturbed to be no longer strongly transparent. Furthermore, we found no representation of referential opacity, which is significant for being a non-transparent natural language phenomenon, in pretrained LMs.

Acknowledgments

We thank the TACL reviewers and action editor for helpful feedback on this work. We thank Kyle Richardson, Jesse Dodge, and other members of AI2 for insightful discussions. This work was funded in part by NSF award 1922658. WM was supported by an NSF graduate research fellowship.

References

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *5th International Conference on Learning Representations, Workshop Track Proceedings*.

C. Anthony Anderson and Joseph Owens, editors. 1990. *Propositional Attitudes: The Role of Content in Logic, Language, and Mind*. CSLI Lecture Notes; No. 20. Center for the Study of Language and Information, Stanford, CA.

Jacob Andreas. 2019. Measuring compositionality in representation learning. In *Proceedings of International Conference on Learning Representations*.

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, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.463

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1075

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. Knowledgeable or educated guess? Revisiting language models as knowledge bases. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.146

Rudolf Carnap. 1947. *Meaning and Necessity: A Study in Semantics and Modal Logic*. University of Chicago Press, Chicago.

Noam Chomsky. 1981. *Lectures on Government and Binding*. Studies in Generative Grammar. Foris Publications.

Noam Chomsky. 1983. *Some Concepts and Consequences of the Theory of Government*

and Binding. Linguistic Inquiry Monograph 6. MIT Press.

Alexis Conneau, German 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, Melbourne, Australia. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1198

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. https://doi.org/10.18653/v1/N19-1423

Philipp Dufter, Nora Kassner, and Hinrich Schütze. 2021. Static embeddings as efficient knowledge bases? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2353–2363, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.186

Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, Boca Raton, Florida, USA.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 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 55–65, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1006

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. https://doi.org/10.1162/tacl_a_00298

Kit Fine. 1990. Quine on quantifying in. *Propositional attitudes: The role of content in logic, language, and mind*, CSLI Lecture Notes; No. 20, pages 1–26. Center for the Study of Language and Information, Stanford, CA.

R. A. Fisher. 1935. *The Design of Experiments*. The Design of Experiments. Oliver and Boyd.

Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71. https://doi.org/10.1016/0010-027790031-5, PubMed: 2450716

Gottlob Frege. 1892. Über sinn und bedeutung. *Zeitschrift für Philosophie Und Philosophische Kritik*.

Aina Garí Soler and Marianna Apidianaki. 2021. Let's play mono-poly: BERT can reveal words' polysemy level and partitionability into senses. *Transactions of the Association for Computational Linguistics*, 9:825–844. https://doi.org/10.1162/tacl_a_00400

Herbert P. Grice. 1975. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics: Vol. 3: Speech Acts*, pages 41–58. Academic Press, New York. https://doi.org/10.1163/9789004368811_003

Jeroen Groenendijk and Martin Stokhof. 1991. Dynamic predicate logic. *Linguistics and Philosophy*, 14(1):39–100. https://doi.org/10.1007/BF00628304

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018.

Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1108

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. 2022. Transformer language models without positional encodings still learn positional information. https://doi.org/10.48550/arXiv.2203.16634

Irene Heim. 1982. *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. thesis, University of Massachusetts Amherst.

Irene Heim and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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 2733–2743, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1275

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. https://doi.org/10.18653/v1/N19-1419

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438. https://doi.org/10.1162/tacl_a_00324

Hans Kamp. 1981. A theory of truth and semantic representation. In Jeroen Groenendijk, Theo Janssen, and Martin Stokhof, editors, *Formal Methods in the Study of Language*, pages 277–322. Amsterdam: Mathematisch Centrum.

Fred Karlsson. 2010. *Syntactic Recursion and Iteration*. Studies in Generative Grammar. De Gruyter Mouton, Germany. https://doi.org/10.1515/9783110219258.43

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.550

Kate Kearns. 2011. *Semantics*. Macmillan Modern Linguistics. Palgrave Macmillan.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. 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 4365–4374, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1445

Saul A. Kripke. 1972. Naming and necessity. In *Semantics of Natural Language*, pages 253–355. Springer.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1813–1827, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.143

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535. https://doi.org/10.1162/tacl_a_00115

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1112

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019b. Inoculation by fine-tuning: A method for analyzing challenge datasets. 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 2171–2179, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1225

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019c. RoBERTa: A robustly optimized BERT pretraining approach. https://doi.org/10.48550/arXiv.1907.11692

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of International Conference on Learning Representations*.

Qing Lyu, Zheng Hua, Daoxin Li, Li Zhang, Marianna Apidianaki, and Chris Callison-Burch. 2022. Is ''my favorite new movie'' my favorite movie? Probing the understanding of recursive noun phrases. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5286–5302, Seattle, United States. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.388

William Merrill, Yoav Goldberg, Roy Schwartz, and Noah A. Smith. 2021. Provable limitations of acquiring meaning from ungrounded form: What will future language models understand? *Transactions of the Association for Computational Linguistics*, 9:1047–1060. https://doi.org/10.1162/tacl_a_00412

Lalchand Pandia and Allyson Ettinger. 2021. Sorting through the noise: Testing robustness of information processing in pre-trained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1583–1596, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.119

Roma Patel and Ellie Pavlick. 2022. Mapping language models to grounded conceptual spaces. In *International Conference on Learning Representations*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1250

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.71

Willard V. Quine. 1956. Quantifiers and propositional attitudes. *The Journal of Philosophy*, 53(5):177–187. https://doi.org/10.2307/2022451

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019.

Language models are unsupervised multitask learners. https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Abhilasha Ravichander, Eduard Hovy, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2020. On the systematicity of probing contextualized word representations: The case of hypernymy in BERT. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 88–102, Barcelona, Spain (Online). Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1410

Tanya Reinhart. 1976. *The Syntactic Domain of Anaphora*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge.

Daniel S. Shabasson. 2018. *The Two Indexical Uses Theory of Proper Names and Frege's Puzzle*. Ph.D. thesis, City University of New York.

Stuart M. Shieber. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343. https://doi.org/10.1007/BF00630917

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.346

Jeff Speaks. 2021. Theories of meaning. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Spring 2021 edition. Metaphysics Research Lab, Stanford University.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *Proceedings of International Conference on Learning Representations*.

Aaron Traylor, Roman Feiman, and Ellie Pavlick. 2021. AND does not mean OR: Using formal languages to study language models' representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 158–167, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-short.21

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. 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 4396–4406, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1448

Kai Von Fintel and Irene Heim. 2011. Intensional semantics. *Unpublished Lecture Notes*.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.586

Zhaofeng Wu, Hao Peng, and Noah A. Smith. 2021. Infusing finetuning with semantic dependencies. *Transactions of the Association for Computational Linguistics*, 9:226–242. https://doi.org/10.1162/tacl_a_00363

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in

transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4896–4907, Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.397

Lang Yu and Allyson Ettinger. 2021. On the interplay between fine-tuning and composition in transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2279–2293, Online. Association for Computational Linguistics.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.398

A Propositional Logic Dataset Details

We hand-designed the PCFG probabilities in Eq. 1. To expand an e, the two binary rules each have 0.06 probability under L_t. The $\neg$ rule and expansion to T and F divide the remaining probability mass, with T and F having the same probability, half of the $\neg$ rule. As S does not expand to T or F, the other three rules proportionally split the probability mass. We consider each of $(,), \wedge, \vee, \neg, \mathsf{T}, \mathsf{F}, =$ as a separate token for tokenization. We enforce a maximum length of 248 tokens. We sample all sentences without replacement. The average L_t sentence length is ≈48.6 tokens. Sampling L_n results in slightly longer sentences, so we decrease the binary rule probabilities to be 0.03 each, but the specification is otherwise the same. The resulting L_n sentence on average has ≈51.7 tokens. We sample 819.2M pretraining sentences and 1M/10K/10K probe training/validation/test sentences. Then, for each split, we sample sentence *pairs*, with the same number as the number of sentences in that split.

B Propositional Logic Training Details

For pretraining, we mostly follow the original hyperparameters for GPT-2-small and RoBERTa-base. We train with batches of 8,192 sequences for 100k steps, equivalent to 1 epoch over our pretraining data. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with epsilon 10^{-8} for ALM for 10^{-6} for MLM, and β_2 0.95 for ALM and 0.98 for MLM. We set the learning rate to 6×10^{-4} warmed up over 10k steps with a 0.1 weight decay.

For probing, +ATTN trains a query vector that interacts with the key representation of each token, obtained with a trained key matrix transformation, and the resulting attention weights are used to average the token embeddings. We train all probes for 3 epochs with batch size 8 and 1,000 warmup steps and select checkpoint with validation accuracy. We use AdamW with 10^{-5} learning rate except only for L_n –ATTN ALM that benefits from a different learning rate 10^{-3}. We clip gradients to unit norm.

C Referential Opacity Dataset Details

We detail the generation of our referential opacity dataset, separately discussing its two aspects (§4.2).

C.1 Generating Co-referring Expressions

For fact probing on LAMA, we use the prompt in the form ‘‘The official language of Laos is known as __’’ which we found appropriate for the entity types in T-REx. If the LM correctly predicts ‘‘Lao’’, we consider this equivalence, or fact, captured by the model. As LAMA was designed to have 1-token answers with BERT’s tokenization, we let BERT fill in the blank. This is not a guarantee for GPT-2’s tokenization, so we run decoding for the same number of steps as the true answer’s length with beam size 5 and no sampling. To further ensure that the predictions are reliable and not due to noise, we only keep entity categories with overall prediction accuracy $> 25\%$. The resulting categories are ‘‘P37 official language’’, ‘‘P364 original language of film or TV show’’, ‘‘P140 religion’’, ‘‘P103 native language’’, and ‘‘P36 capital’’. This procedure results in 1,606 facts for GPT-2 and 2,962 facts for BERT.

C.2 Generating Contexts

We generate two types of contexts (§4.2). The first type contains an embedded clause, for which we construct templates for each entity category

in §C.1. For language entities, for example, one template is ‘‘[PRONOUN] [VERB] to speak [ENTITY].’’ A sentence pair is formed by filling in [ENTITY] with a definite description vs. a proper name for a fact. We only consider the pronouns ‘‘She’’ and ‘‘He’’ in this work. We consider 6 referentially transparent verbs (‘‘starts’’, ‘‘begins’’, ‘‘ceases’’, ‘‘stops’’, ‘‘managed’’, ‘‘failed’’) and 6 referentially opaque verbs (‘‘wants’’, ‘‘intends’’, ‘‘hopes’’, ‘‘begs’’, ‘‘preferred’’, ‘‘suggested’’). The second type of context contains only the main clause. We use the referentially opaque template ‘‘[PRONOUN] dislikes [ENTITY].’’ and an entity category-specific referentially transparent template such as ‘‘[PRONOUN] speaks [ENTITY].’’ In total, we have 64,672 sentence pairs for GPT-2 and 121,768 for BERT.

For our probing analysis, we also included attractors with coordinated sentences (§4.4). As there are a quadratic number of possible coordinations, we subsampled 59,548 such sentences for GPT-2 and 119,540 for BERT, similar to the number of attractor-less sentences. We split all sentence pairs 8/1/1 for training/validation/testing.

For our similarity analysis, for a cleaner significance test, we only consider sentence pairs with an embedded clause. This leaves 58,776 sentence pairs for GPT-2 and 111,312 for BERT.

D Referential Opacity Training Details

The probe is trained similarly to §B except for 1 epoch with batch size 256 and learning rate 10^{-5}.

E Can Language Models Learn to Represent Referential Opacity With Appropriate Supervision?

We showed in §4 that we do not observe evidence of pretrained language models representing the phenomenon of referential opacity. A natural question, then, is whether language models can *learn* to represent it. Following a similar setup as Lyu et al. (2022) and Liu et al. (2019b), we finetune the entire model on a portion of our training set for 1 epoch and conduct the same probing procedure on the resulting model. All training is done with the coordinated data introduced (§4.4). Finetuning uses the same hyperparameters in §D. Similar to §4.4, we report the mean and standard deviation across 10 random seeds for each setting.

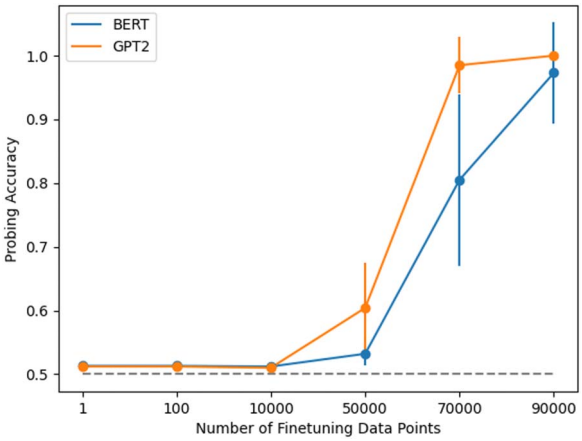

Figure 2: Probing accuracy after finetuning a pretrained LM on our (coordinated) referential opacity dataset with different numbers of finetuning examples. The mean and the standard deviation across 10 seeds are plotted. For clarity in visualizing the trend, *the x-axis is not in linear scale*.

We plot the probing accuracy along with the number of finetuning examples in Figure 2. Both GPT-2 and BERT continue to be unable to perform above-random with up to 10,000 finetuning examples, further demonstrating their inadequate semantic representation of referential opacity. Nevertheless, with enough finetuning examples, both models eventually achieve near-100% probing accuracy. It is, therefore, possible that they can potentially learn to represent referential opacity with sufficient semantic supervision, though we note a caveat: while we introduced coordinated data to prevent an obvious shortcut that the model could take (§4.4), it does not eliminate all possible shortcuts. It could be the case that the additional capacity afforded by finetuning enables the model to exploit a more sophisticated shortcut (unknown to us) instead of truly capturing this phenomenon.