Language Models Use Monotonicity to Assess NPI Licensing

Jaap Jumelet[↑]   **Milica Denić**[↑]   **Jakub Szymanik**[↑]
Dieuwke Hupkes[λ]   **Shane Steinert-Threlkeld**[↓]

[↑] Institute for Logic, Language and Computation, University of Amsterdam
[λ] Facebook AI Research
[↓] Department of Linguistics, University of Washington

{j.w.d.jumelet, m.denic, J.K.Szymanik}@uva.nl   dieuwkehupkes@fb.com   shanest@uw.edu

Abstract

We investigate the semantic knowledge of language models (LMs), focusing on (1) whether these LMs create categories of linguistic environments based on their semantic *monotonicity* properties, and (2) whether these categories play a similar role in LMs as in human language understanding, using negative polarity item licensing as a case study. We introduce a series of experiments consisting of probing with diagnostic classifiers (DCs), linguistic acceptability tasks, as well as a novel *DC ranking* method that tightly connects the probing results to the inner workings of the LM. By applying our experimental pipeline to LMs trained on various filtered corpora, we are able to gain stronger insights into the semantic generalizations that are acquired by these models.[1]

1 Introduction

Neural language models (LMs) have become powerful approximators of human language, making it increasingly important to understand the features and mechanisms underlying their behavior (Linzen et al., 2018, 2019). In the past few years, a substantial number of studies have investigated the linguistic capabilities of LMs (Gulordava et al., 2018; Giulianelli et al., 2018; Lakretz et al., 2019; Wu et al., 2020; Ettinger, 2020, i.a.). Such work has focused primarily on *syntactic* properties, while fewer studies have been done on what kind of *formal semantic* features are encoded by language models. In this paper, we focus explicitly on what LMs learn about a semantic property of sentences, and in what ways their knowledge reflects well-known features of human language processing.

As the topic of our studies, we consider **monotonicity**, a semantic property of linguistic environments that plays an important role in human language understanding and inference (Hoeksema, 1986; Valencia, 1991; Van Benthem, 1995; Icard III and Moss, 2014): the monotonicity of a linguistic environment determines whether inferences from a general to a particular term or vice versa are valid in that environment. For example, the fact that the inference from "*Mary didn't write a paper*" to "*Mary didn't write a linguistics paper*" is valid shows us that the position where "*a paper*" occurs is *downward monotone*: the inference is valid when a more general term ("*a paper*") is replaced with a more specific one ("*a linguistics paper*").

To investigate monotonicity we focus on **negative polarity items** (NPIs): a class of expressions such as *any* or *ever* that are solely acceptable in downward monotone environments (Fauconnier, 1975; Ladusaw, 1979). Psycholinguistic research has confirmed this connection between NPIs and monotonicity: humans judge NPIs acceptable in a linguistic environment if they consider that environment to be downward monotone (Chemla et al., 2011). Previous research has established that LMs are relatively successful in processing NPIs (Warstadt et al., 2019), but without investigating *how* they came to these successes.

We raise the following research questions:

RQ1 Do language models encode the monotonicity properties of linguistic environments?

RQ2 To what extent do they employ this information when processing negative polarity items?

We developed a series of experiments, in which we first evaluate the general capacities of LMs in handling monotonicity and NPIs and then investigate the generalization heuristics of the LM by doing experiments with modified training corpora. First, we establish that LMs are able to encode a notion of monotonicity by probing them with diagnostic classifiers (DCs, Hupkes et al., 2018) (§5.1).

---

[1] All code and data can be found at https://github.com/jumelet/monotonicity-npi-lm

In our second experiment we demonstrate that our LMs are reasonably successful with NPI licensing using an NPI acceptability task (§5.2). Next, we introduce a novel *DC ranking* method to investigate the overlap between the information that the model uses to make judgments about NPIs and the information that the DCs use to predict monotonicity information, finding that there is a significant overlap (§5.3).

We then investigate two potential confounds that may obfuscate our results. First, we consider whether the signal that is picked up by the monotonicity DC is not simply a proxy that tells the model that an NPI may occur at that position (§5.4). To assess this, we train new LMs on a corpus from which all sentences with NPIs have been removed, re-run the montonicity probing task, and find that even in the absence of NPI information, LMs are still able to encode a notion of monotonicity.

Next, we consider whether an LM bases its NPI predictions on simple co-occurrence heuristics, or if it can extrapolate from a general notion of monotonicity to cases of NPIs in environments in which they have never been encountered during training (§5.5). We again train new LMs on modified corpora, this time removing NPIs only in one specific environment, and repeat the NPI acceptability and DC ranking experiments. The results of this setup demonstrate that LMs indeed use a general notion of monotonicity to predict NPI licensing.

Contributions With this work, we contribute to the ongoing study of the linguistic abilities of language models in several ways:

- With a series of experiments we demonstrate that LMs are able to acquire a general notion of monotonicity that is employed for NPI licensing.

- We present two novel experimental setups: *filtered corpus training* and *DC ranking*, that can be used to assess the impact of specific information during training and compare the information used by DCs with the information used with the model, respectively.

- By using experimental results from psycho-semantics to motivate hypotheses for LM behavior, we find that our models reflect behavior similar to human language processing.

In the remainder of this paper, we will first provide some linguistic background that helps to situate and motivate our experiments and results (§2).

We then discuss related work on NPI processing in LMs in §3. In §4, we discuss our methods and experimental setup. §5.1 through §5.5 explain and present the results. We conclude in §6 with a general discussion and pointers to future work.

2 Linguistic Background

Monotonicity Monotonicity is a property of a linguistic environment which determines what kind of inferences relating general and particular terms are valid in that environment. If inferences from a general to a particular term are valid, the linguistic environment is said to be *downward monotone* (DM). If inferences are valid the other way around, from a particular to a general term, the linguistic environment is said to be *upward monotone* (UM).

Examples of expressions inducing DM environments are negation and quantifiers like *nobody, no NP*, but also specific types of adverbs and the antecedents of conditional sentences. For instance, (1) below exemplifies that in these environments the inference from a sentence with a general term (*cookies*) to that sentence with a more particular term (*chocolate cookies*) is valid, but not vice versa.

(1)　a.　Mary did**n't** eat cookies. ⟹
　　　　Mary didn't eat *chocolate* cookies.
　　b.　**Nobody** ate cookies. ⟹
　　　　Nobody ate *chocolate* cookies.
　　c.　Mary **rarely** ate cookies. ⟹
　　　　Mary rarely ate *chocolate* cookies.

Common examples of UM environments are (non-quantified) positive sentences, quantifiers such as *somebody, many NP*, and other kind of adverbs. (2) exemplifies that in these environments the inference from a sentence with a more particular term (*chocolate cookies*) to the same sentence with a general term (*cookies*) is valid, but not vice versa.

(2)　a.　Mary ate *chocolate* cookies. ⟹
　　　　Mary ate cookies.
　　b.　**Everyone** ate *chocolate* cookies. ⟹ Everyone ate cookies.
　　c.　Mary **often** ate *chocolate* cookies. ⟹
　　　　Mary often ate cookies.

NPIs NPIs are expressions such as the English words *any, anyone, ever*, whose acceptability depends on whether its linguistic environment is downward monotone (Fauconnier, 1975; Ladusaw, 1979; Dowty, 1994; Kadmon and Landman, 1993; Krifka, 1995; Lahiri, 1998; Chierchia, 2006,

2013).[2] While the conditions for NPI acceptability are complex, a good approximation is that NPIs are acceptable (or *licensed*) in the syntactic scope of *NPI licensors* that induce a DM environment.[3] If we again consider the DM environment of (1-a) and the UM environment of (2-a), it can be seen that English *any* is an NPI, as it is acceptable when inside the syntactic scope of negation (a DM expression) as in (3-a), and not acceptable when they are in an UM environment as in (3-b).

(3)   a.   Mary didn't eat (any) cookies.
         b.   Mary ate (*any) cookies.

Importantly, monotonicity plays a role at the psychological level: human judgments about the monotonicity of a linguistic environment predict their judgments of NPI acceptability in that environment (Chemla et al., 2011; Denić et al., 2021). For example, how plausible someone finds the inference (1-a) predicts how acceptable they find the sentence (3-a). Summing up, NPI licensing has a syntactic component (NPIs must reside in syntactic scope of a licensor) and a semantic component (NPI licensors are DM expressions), that are connected on a psychological level (monotonicity judgments predict NPI acceptability). Our research aims to uncover whether this connection is exhibited by LMs as well.

3   Related work

The literature on interpreting LMs has grown substantially in the last few years (see, e.g. Belinkov and Glass, 2019; Alishahi et al., 2019; Rogers et al., 2021, for survey papers). Several studies investigate how they process NPIs, focused mainly on the *syntactic* aspect of NPI licensing.

Jumelet and Hupkes (2018) conclude that LSTM language models encode information about the dependency between the NPI and the NPI licensor, although this effect diminishes as the distance between the NPI and its licensor grows. Marvin and Linzen (2018) study NPI judgments of LMs on minimally different sentence pairs (with the NPI licensor either in an appropriate syntactic configuration or not) and find that their models are unable to reliably assign higher probability to sentences in

which NPIs are correctly licensed. The syntactic aspect of NPI licensing is also examined by Futrell et al. (2019), who demonstrate that LSTM LMs are susceptible to learning spurious licensing relationships, a finding that Warstadt and Bowman (2020) demonstrate to also hold for BERT (Devlin et al., 2019). Wilcox et al. (2019) investigate how explicit syntactic supervision of LMs affects their success with syntactic aspects of NPI licensing. The broad linguistic suites of Warstadt et al. (2020) and Hu et al. (2020) also contain a set of tasks related to NPI licensing, demonstrating that it is one of the most challenging tasks for LMs to handle. Weber et al. (2021) investigated the dynamics of NPI learning during training, and connected this to a multi-task learning paradigm, demonstrating that LMs are able to efficiently leverage information from related licensing environments.

Lastly, Warstadt et al. (2019) examine BERT's ability in determining NPI acceptability. They demonstrate that BERT has significant knowledge of the dependency between NPIs and their licensors, but that this success varies widely across different experimental methods. Our study builds on that of Warstadt et al. (2019). Although they demonstrate that BERT is generally successful with NPI licensing, their results do not reveal whether BERT has constructed a more general category of DM expressions that is independent of collocational cues, nor whether it has understood that this category matters for NPI licensing.

4   Methods

Before getting to the main experimental part of our work, we briefly discuss the training corpus, model architecture and evaluation corpus we consider.

Training Corpus   The base training corpus we consider in our experiments is the corpus used by Gulordava et al. (2018). This corpus is a collection of sentences from Good and Featured English Wikipedia articles and consists of over 90M tokens. The vocabulary of the corpus consists of the 50.000 most frequent tokens in this corpus; less frequent tokens are mapped to a special `<unk>` token. We refer to the full training corpus type with the name Full, and to the LMs trained on this corpus as Full LMs. In addition to Full, we use multiple other corpora which are derived from Full by means of filtering. This will allow us to draw conclusions about specific generalization abilities and reliance on collocational cues of LMs; filtered corpora will

---

[2] See however Zwarts, 1995; Giannakidou, 1998; Barker, 2018 for different takes on NPI acceptability generalizations.

[3] An NPI occurs in the syntactic scope of a licensor if the licensor *c-commands* the NPI. An NPI licensor c-commands an NPI if the NPI is the licensor's sister node or one of its sister's descendants in a constituent tree (Reinhart, 1976).

Environment Class	Abbrev.	DM example	UM example
Adverbs	ADV	A lady **rarely** *ever* ...	*A lady sometimes *ever*...
Conditionals	COND	**If** the dancers see *any* ...	*While the dancers see *any*...
Determiner Negation	D-NEG	**No** teacher says that the students had practiced *at all*.	*Some teacher says that the students had practiced *at all*.
Sentential Negation	S-NEG	The dancer was **not** saying that the guy had profited *yet*.	*The dancer was really saying that the guy had profited *yet*.
Only	ONLY	**Only** the boys had *ever* ...	*Even the boys had *ever* ...
Quantifiers	QNT	**Every** senator who had *ever* ...	*Some senator who had *ever* ...
Embedded Questions	QUES	The patients wonder **whether** the lady admires *any* ...	*The patients say that the lady admires *any*...
Simple Questions	SMP-Q	Did the boy *ever* listen**?**	*The boy did ever listen.
Superlatives	SUP	A lady buys the **oldest** dish that the adult had *ever* ...	*A lady buys the old dish that the adult had *ever* ...

Table 1: The nine environment classes of Warstadt et al. (2019), with an example of a minimal DM/UM pair for each class taken from the corpus.

be introduced in the relevant sections.

Model Architecture In our studies, we focus on recurrent language models. More specifically, following Gulordava et al. (2018), we consider two-layer LSTM language models, with an embedding and hidden size of 650. All training runs across our experiments follow the same regime, identical to the regime described by Gulordava et al. (2018): 40 epochs of training with SGD, with a plateau scheduler and an initial learning rate of 20, a batch size of 64, BPTT length of 35, and dropout of 0.1.[4]

Evaluation Corpus To assess monotonicity and NPI licensing knowledge of LMs in our experiments, we leverage the NPI corpus of Warstadt et al. (2019), which consists of a large amount of grammatical and ungrammatical sentences with NPIs. This corpus is divided into 9 distinct **environment classes**, allowing for fine-grained analysis of NPI licensing. Importantly, these nine environment classes come in two versions: a DM version—in which NPIs are grammatically acceptable, and a minimally different UM version—in which they are not. We provide an overview with examples of DM and UM versions of all environment classes in Table 1. The full size of the corpus is 106.000 distinct DM sentences, and the division of environment classes is split roughly uniformly.

5   Experiments and Results

In this section we describe the experimental pipeline in more detail. A graphical overview of our experiments is depicted in Figures 1 and 4. Each experiment description is directly followed by an analysis of its results.

---

[4]Models are trained on a *GeForce 1080 Ti* GPU, take around 40 hours to train, and consist of 71M parameters.

5.1   Experiment 1: Do LMs represent monotonicity information?

In our first experiment, we test whether LMs trained on our Full corpus possess a notion of monotonicity. We train five different LMs and test how well they represent monotonicity properties of different environments by training linear **diagnostic classifiers** (DCs, Hupkes et al., 2018) on top of the hidden states of the LM. To create a corpus of monotonicity sentences for training and testing the DCs, we leverage the corpus of Warstadt et al. (2019), now selecting all DM and UM sentences to build up a balanced corpus of these categories. The nine environment classes in that corpus hence provide a broad spectrum of DM environments and their minimally different UM counterparts.

For training and testing the DCs, we consider the hidden states at the position directly before an NPI occurs (see Figure 1). The reason we train the DCs at this position is because only at this point we are sure that the monotonicity information should surface and be encoded *linearly*. This is due to the fact that the decoder of the LM that transforms a hidden state into a probability distribution is linear as well: if the probability of some token depends on a linguistic feature, this feature must hence be encoded linearly. The DCs are implemented using the `diagNNose` library of Jumelet (2020), and trained using 10-fold cross-validation, Adam optimization (Kingma and Ba, 2015), a learning rate of 10^{-2} and L1 regularization with $\lambda = 0.005$.

We train our monotonicity DCs in two separate ways. First, we divide the entire monotonicity corpus into a 90/10 train/test split, sampled *uniformly* across the different environment classes. This allows us to examine whether DM and UM environments are linearly separable in a way that is applicable to all environment classes. We refer to this classifier as the All-ENV DC.
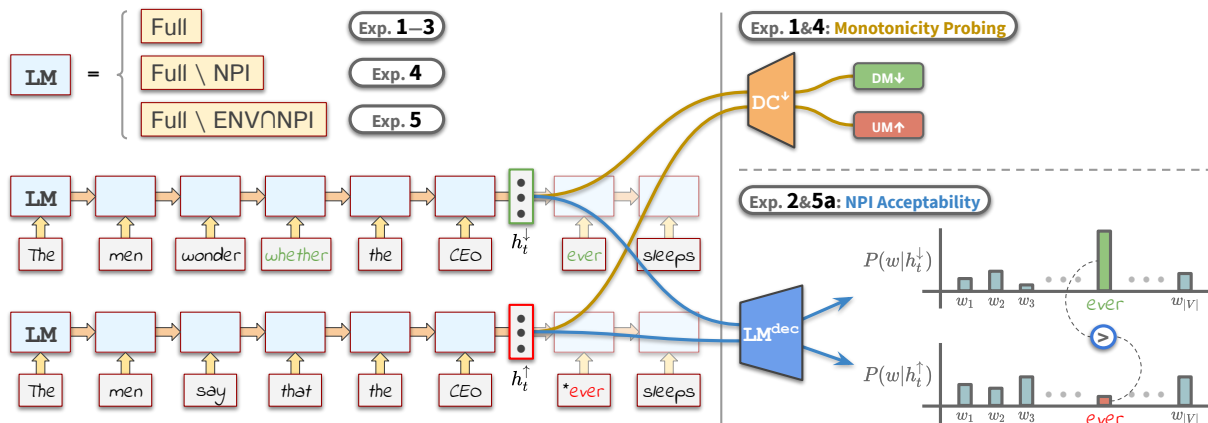
Second, we move to a more fine-grained type

Figure 1: The pipeline of our experimental setup. We start by computing the hidden states $h_t^\downarrow$ (within a DM environment ahead of the NPI) and $h_t^\uparrow$ (within a UM environment). These hidden states are then used for training the monotonicity DC (Exp. 1 & 4), and to compare $P_{\text{LM}}(\text{NPI}|h_t^\downarrow) > P_{\text{LM}}(\text{NPI}|h_t^\uparrow)$ (Exp. 2 & 5a). The task of Experiments 3 and 5a can be found in Figure 4. Experiments 4 and 5 consist of the same tasks as the first three experiments, but differ in the language model that is used.

of analysis. High performance of the All-ENV DC namely does not provide evidence that monotonicity is encoded the same way for each environment: the set of salient hidden units used by the All-ENV DC for classifying monotonicity within the *Adverbs* environment, for example, could be disjoint from the set of units used for the *Only* environment. To investigate this, we train a DC on the hidden states of all-but-one environment class, and test its performance on the excluded class. This provides a measure to what extent the monotonicity representation of DM and UM environments derived from all other environment classes *generalizes* to the held-out class, demonstrating stronger evidence that the model represent monotonicity in the same way across different environments.

Results The results of our first experiment are shown in the top row of Figure 2. The first column contains the average accuracy for the All-ENV DC, and it can be seen that the diagnostic classifier succeeds in this task with high accuracy (97%). This indicates that the uniform split over all environment classes is linearly separable.

Next, we consider the held-out evaluation procedure for each of the nine environment classes. It can be seen that the monotonicity signal generalizes well to five classes (*adverbs, determiner negation, only, sentential negation, and embedded questions*), all with an accuracy above 90%. The other four classes yield a higher standard deviation, indicating that these classes are encoded less consistently across initialization seeds. The accuracy

for all held-out DCs is lower compared to the All-ENV DC results, indicating that the All-ENV DC relied partly on information unrelated to a shared notion of monotonicity. The fact that the accuracy of these DCs is still so high, however, indicates that there is a substantial overlap between the way that monotonicity is encoded within the different environments.

5.2 Experiment 2: Do LMs predict the licensing conditions of NPIs?

In the next experiment we investigate the NPI acceptability judgments of the Full LMs on the corpus of Warstadt et al. (2019). This is done by comparing the probability of an NPI conditioned on the model's representation of a DM environment ($h_t^\downarrow$) and a UM environment ($h_t^\uparrow$), where success is defined as follows:

$$P_{\text{LM}}(\text{NPI}|h_t^\downarrow) > P_{\text{LM}}(\text{NPI}|h_t^\uparrow)$$

This is a common evaluation procedure in the interpretability literature (Linzen et al., 2016), and has earlier been applied in the domain of NPI licensing by Jumelet and Hupkes (2018) and Warstadt et al. (2020). Our approach is similar to the Cloze Test of Warstadt et al. (2019), but their setup used (bi-directional) masked LMs, making it possible to directly compare the probabilities of the NPI licensor, instead of comparing the NPI probabilities. Note that we purposefully do not base NPI acceptability on comparing full sentence probabilities: in our view this type of comparison can be distracted by token probabilities not related to the NPI itself.

Monotonicity probing accuracy

	All-ENV	ADV	COND	D-NEG	S-NEG	ONLY	QNT	QUES	SMP-Q	SUP
Full (exp. 1)	$0.97_{\pm 0.01}$	$0.93_{\pm 0.02}$	$0.81_{\pm 0.16}$	$0.97_{\pm 0.01}$	$0.91_{\pm 0.02}$	$0.94_{\pm 0.03}$	$0.78_{\pm 0.05}$	$0.93_{\pm 0.06}$	$0.80_{\pm 0.08}$	$0.66_{\pm 0.04}$
Full \ NPI (exp. 4)	$0.95_{\pm 0.01}$	$0.88_{\pm 0.02}$	$0.89_{\pm 0.11}$	$0.94_{\pm 0.01}$	$0.86_{\pm 0.04}$	$0.95_{\pm 0.02}$	$0.77_{\pm 0.03}$	$0.89_{\pm 0.05}$	$0.78_{\pm 0.05}$	$0.57_{\pm 0.03}$

DC evaluated on held-out environment class

Figure 2: Accuracy and standard deviation on the monotonicity diagnostic classification task, averaged over 5 seeds for each model type. The All-ENV column denotes train/test split procedure sampled uniformly over all environment class; other columns denote accuracy on one environment class that has been excluded during training.

NPI acceptability accuracy

	ADV	COND	D-NEG	S-NEG	ONLY	QNT	QUES	SMP-Q	SUP
Full (exp. 2)	$0.88_{\pm 0.03}$	$0.80_{\pm 0.03}$	$0.93_{\pm 0.03}$	$0.82_{\pm 0.02}$	$0.84_{\pm 0.03}$	$0.79_{\pm 0.02}$	$0.85_{\pm 0.02}$	$0.72_{\pm 0.01}$	$0.76_{\pm 0.02}$
Full \ ENV∩NPI (exp. 5a)	$0.81_{\pm 0.00}$	$0.65_{\pm 0.04}$	$0.89_{\pm 0.04}$	$0.69_{\pm 0.06}$	$0.74_{\pm 0.05}$	$0.69_{\pm 0.01}$	$0.83_{\pm 0.05}$	$0.68_{\pm 0.00}$	$0.72_{\pm 0.01}$

Environment class

Figure 3: Accuracy on the NPI acceptability task—based on whether the NPI was assigned a higher probability in the DM environment than in its UM counterpart.

We split this procedure out for each of the nine environment classes. The example sentence of the Simple Questions environment in Table 1, for example, is evaluated as follows:

$$P_{\text{LM}}(ever|\text{Did the boy}) > P_{\text{LM}}(ever|\text{The boy did})$$

Using the full sentence probabilities for this comparison would require taking probabilities into account such as $P_{\text{LM}}(the|Did)$ and $P_{\text{LM}}(boy|The)$, that have no relation to NPI licensing at all.

Results We present the results for this experiment in the top row of Figure 3. The Full models demonstrate a considerable ability at predicting NPI acceptability, with the least performing class (SMP-Q, *Simple Questions*) yielding an accuracy that is still well above chance (0.72). Compared to earlier investigations on the ability of LSTM LMs in NPI licensing, our results indicate that these models are able to obtain a more sophisticated understanding of NPIs than previously thought: both Marvin and Linzen (2018) and Hu et al. (2020) report LSTM performance below chance on NPI acceptability tasks. This might in part be due to the different evaluation procedure we used (conditional vs. full-sentence probability comparison).

5.3 Experiment 3: Is the LM's knowledge of DM environments and of NPI licensing related?

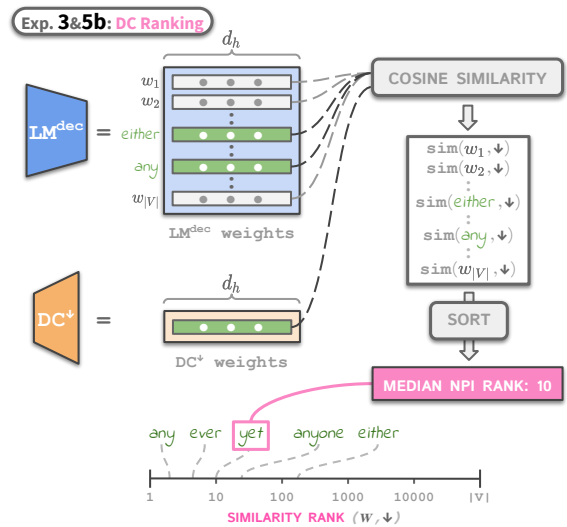We have now established that our models encode a signal related to monotonicity, and are successful

Figure 4: The *DC Ranking* experiment, in which we investigate whether the monotonicity DC and the LM decoder base their predictions on similar cues, by computing and ranking the cosine similarities between the DC weights and the decoder weights of each token.

at predicting NPI acceptability. In our third experiment, we assess to what extent the parameters used by the LM to predict NPIs (i.e. the LM's decoder embeddings for NPIs) *overlap* with the information the DCs use to predict the monotonicity properties of a particular environment class. For this we have devised a novel **DC ranking** method, that ranks the LM's decoder weights for all tokens based on their similarity with the DC weights.

We present a schematic overview of the method

4963

LM decoder & monotonicity DC cosine similarity − Median NPI rank

	All-ENV	ADV	COND	D-NEG	S-NEG	ONLY	QNT	QUES	SMP-Q	SUP
Full (exp. 3)	9	14	1121	16	28	449	206	82	14124	1248
Full \ ENV∩NPI (exp. 5b)	10	35	6634	622	1152	1418	1509	297	28670	3288

Environment class monotonicity DC trained on

Figure 5: Results on the median NPI rank task. A low median rank indicates that the monotonicity DC uses the same representational information as the NPI decoder.

in Figure 4. The LM's decoder weight matrix can be interpreted as a collection of vectors corresponding to each token in the model's vocabulary. The monotonicity DC is a binary classifier, so its weights are represented by a single vector. The LM's decoder vectors are of the same dimensionality as the weight vector of the monotonicity DC, which allows us to compute the similarity between each decoder vector and the monotonicity DC. For each of the 50.000 tokens in the LM's vocabulary, we calculate the cosine similarity between the decoder weights corresponding to that token and the DC's weights. We then sort these similarity scores, which results in a ranking of tokens that are most similar to the DC.

As we are interested in finding the connection of the monotonicity DC and the LM's NPI processing in general, we compute the **median rank** over a set of 11 NPIs.[5] A low median NPI rank indicates that the LM uses the same cues for NPI prediction as the monotonicity DC, demonstrating a clear connection between NPI licensing and monotonicity.

Contrary to Experiment 1, we no longer make use of the hold-one-out training procedure, that gave insights to what extent a general monotonicity signal generalizes to a held-out environment class. Instead, we train a separate diagnostic classifier for each environment class using a train/test split made up of DM and UM environments within that class. This results in a classifier that represents the class-specific decision boundary between minimal pairs of DM and UM items and allows us to investigate to what extent these decision boundaries align with the weights of the LM decoder. Next to the environment-specific DCs we also report the DC ranking outcome for the All-ENV DC that has been trained on all environments.

Results The results of this experiment are presented in the top row of Figure 5. The first column (All-ENV) contains the result for the DC trained on all environment classes, and the median NPI rank of 9 demonstrates that the monotonicity DC aligns very closely with the NPI decoder weights of the LMs. This median rank should be interpreted within the context of the model vocabulary size: it can range upwards to 50.000, so a rank that is close to 0 signifies a tight connection between the probing task and the tokens of interest.

Moving on to the environment-specific results, it can be seen that the results vary considerably between the environment classes. The worst scoring class is again that of Simple Questions. This makes sense, as the licensing conditions for question constructions do not depend on the presence of a specific licensing token such as *not*, but on the overall structure of the whole sentence. The other environment classes lead to scores far closer to 0, indicating that for these classes monotonicity classification is closely aligned to NPI processing.

Interestingly, the median rank of the All-ENV DC is lower than the ranks of all other DCs. This shows that the model has aligned its representation of NPIs to an aggregate of the monotonicity representations in the different environment classes. This allows the model to flexibly deal with NPIs in a wide range of licensing environments.

5.4 Experiment 4: Are NPIs important for learning monotonicity information?

With Experiment 3 we established that NPI processing and monotonicity are related in our LMs. Now, we investigate to what extent their representations are entangled during training. More specifically, we investigate if the signal from the presence of NPIs is indispensable for the LM to develop a notion of monotonicity, or if instead the success in categorizing monotonicity environments can be learned independently of NPIs.

We address this question by testing whether LMs

---

[5]We consider the following 11 single-token NPI expressions: *dared, any, anybody, anymore, anyone, anything, anywhere, ever, nor, whatsoever,* and *yet*. These are all the single-token NPIs taken from a list of NPIs that is described in §5.4.

can still classify the monotonicity properties of environments when they are completely deprived of NPIs during training. To do so, we train new language models on a modified corpus that does not contain any NPIs at all. To arrive at this corpus, we remove all sentences that contain at least one NPI expression from the Full corpus. We identify these expressions based on a comprehensive list of NPI expressions in English collected by Hoeksema (2012) and the list of NPIs in English compiled by Israel (2011). From this list, we manually removed expressions that have both NPI and non-NPI uses (e.g. *a thing, a bit*). The 40 NPI expressions that resulted from this procedure can be found in Appendix A. We train 5 models on this corpus and refer to them by the name Full\NPI.

In this experiment, we run the monotonicity probing procedure of Experiment 1 on the Full\NPI models. We posit that if the notion of monotonicity can be learned independently of NPIs, there should be no significant drop in performance compared to the results of the Full LMs.

Results   We report the results of this experiment in the bottom row of Figure 2. Again it can be noted that the All-ENV DC, trained and tested uniformly over all environment classes, obtains a high accuracy on the task (0.95). Furthermore, none of the held-out environment DCs lead to significant drops in performance compared to the Full LMs. Based on this we conclude that even in the absence of NPI cues, LMs are still able to build up a shared robust notion of monotonicity.

5.5   Experiment 5: How robust is the connection between monotonicity and NPI processing?

This research aims to uncover whether LMs possess a robust connection between monotonicity and NPI licensing. Our findings indicate that this connection is present in our models. A major confound that has not yet been addressed, however, is the extent to which our models rely on collocational cues when judging the acceptability of an NPI. To test this, we examine whether an LM's connection between NPIs and monotonicity *generalizes* to novel environment classes in which NPIs have never been encountered during the training phase of the LM.

We have created nine modified corpora in which sentences with NPIs within a specific environment have been removed. For these different corpora, we again consider the nine NPI-licensing environments of Warstadt et al. (2019). For each environment class we create a new corpus by removing all sentences from the Full corpus in which an NPI expression from Appendix A is preceded by an expression belonging to that class, somewhere earlier in the sentence.[6] Note that we only remove the sentences in which the environment actually licenses an NPI; sentences in which the environment occurs without an NPI are retained. So for the *adverbs* environment, for example, we remove sentences like "*Mary rarely ate any cookies*" but not "*Mary rarely ate cookies*". For each of these nine corpora we train 3 new LMs. Models trained on these corpora are referred to by the name Full\ENV∩NPI.

We run the NPI acceptability task of Experiment 2 and the DC ranking method of Experiment 3 on the nine types of Full\ENV∩NPI models. A model with a robust connection between monotonicity and NPI processing should be able to learn for NPIs in the held-out environment that (i) the environment belongs to the class of environments in which NPIs are licensed, and that (ii) determining NPI acceptance should be done based on representational cues that are similar for monotonicity prediction.

Results   We report the results of this experiment next to the previous results of the Full model. First, we consider the NPI acceptability task, which is reported in the bottom row of Figure 3. Note that each cell in this row now corresponds to a specific model type: the ADV result, for instance, corresponds to the accuracy of the Full\ENV∩NPI models in which sentences with NPIs within adverbial environments have been removed. Our results show that the performance drops slightly for all environment classes, which can be attributed to a model's dependence on collocational cues. However, the models are still able to adequately generalize from the other environments, in which NPIs still are encountered, to the held-out environment. This demonstrates the semantic generalization capacities of the LM: it infers that the held-out environment in which NPIs have never been encountered shares some relevant properties with the other eight environment classes in which NPIs still occur.

The results for the DC ranking experiment are shown in the bottom row of Figure 5. Similar to the NPI acceptability results, the performance of the Full\ENV∩NPI models has dropped slightly compared to the Full models. However, if the models

---

[6] For Simple Questions we remove a sentence if an NPI occurs in a sentence that ends with a question mark.

would no longer pick up on the connection between monotonicity in the held-out environment and NPI licensing at all, these median ranks should drop to chance, i.e. around the halfway mark of the vocabulary size (25.000). It can be seen that this is only the case for the Simple Questions environment, that was already performing poorly for the Full models. Based on this we conclude that although models depend partly on collocational cues for their connection between monotonicity and NPIs, they are still able to encode a robust connection that generalizes to novel DM environments.

6 Conclusion

Based on a series of experiments, we have established the following: (1) LMs categorize environments into DM and UM; (2) LMs are overall successful with NPI licensing; (3) LMs employ similar representational cues when processing NPIs and predicting monotonicity; (4) their categories of DM and UM environments can be learned independently of NPI occurrence; and (5) their connection between monotonicity and NPI processing is robust and not solely dependent on co-occurrence heuristics. This demonstrates that LMs have quite sophisticated knowledge of NPI licensing, which may be similar to that of humans and constitutes a vital step towards better understanding the linguistic generalization capacities of LMs.

These results raise the question: what do LMs learn about the DM and UM environments which they succeed in finding? Do they actually learn the inferential properties of those environments, or do they rely on some other property that DM environments have in common to categorize them as such? A direction for future work would be to develop methods to probe the inferential capacities of LMs and explore how they align with the DM and UM categories they construct.

Another direction for future work would be to incorporate the recent advancements on probing-based interpretability methods in our experimental pipeline (Hewitt and Liang, 2019; Voita and Titov, 2020). Our DC Ranking method aligns the performance of a probe with that of the language model itself, which is related to the approaches of Saphra and Lopez (2019), Elazar et al. (2021), and Lovering et al. (2021). Placing our methodology more firmly in this body of work will allow for stronger conclusions to be drawn regarding the semantic knowledge of current language models.

References

Afra Alishahi, Grzegorz Chrupała, and Tal Linzen. 2019. Analyzing and interpreting neural networks for NLP: A report on the first BlackboxNLP workshop. *Natural Language Engineering*, 25(4):543–557.

Chris Barker. 2018. Negative polarity as scope marking. *Linguistics and Philosophy*, 41(5):483–510.

Yonatan Belinkov and James Glass. 2019. Analysis Methods in Neural Language Processing: A Survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Emmanuel Chemla, Vincent Homer, and Daniel Rothschild. 2011. Modularity and intuitions in formal semantics: The case of polarity items. *Linguistics and Philosophy*, 34(6):537–570.

Gennaro Chierchia. 2006. Broaden your views: Implicatures of domain widening and the "logicality" of language. *Linguistic inquiry*, 37(4):535–590.

Gennaro Chierchia. 2013. *Logic in Grammar: Polarity, Free Choice, and Intervention*. Oxford Studies in Semantics and Pragmatics 2. OUP Oxford.

Milica Denić, Vincent Homer, Daniel Rothschild, and Emmanuel Chemla. 2021. The influence of polarity items on inferential judgments. *To appear in Cognition*.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

David Dowty. 1994. The role of negative polarity and concord marking in natural language reasoning. In *Semantics and Linguistic Theory*, volume 4, pages 114–144.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Trans. Assoc. Comput. Linguistics*, 9:160–175.

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.

Gilles Fauconnier. 1975. Polarity and the scale principle. *Chicago Linguistics Society*, 11:188–199.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Anastasia Giannakidou. 1998. *Polarity sensitivity as (non) veridical dependency*, volume 23. John Benjamins Publishing.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.

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.

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.

Jack Hoeksema. 1986. Monotonicity phenomena in natural language. *Linguistic Analysis*, 16(1–2):235–250.

Jack Hoeksema. 2012. On the natural history of negative polarity items. *Linguistic Analysis*, 38(1):3–33.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'Diagnostic Classifiers' Reveal how Recurrent and Recursive Neural Networks Process Hierarchical Structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Thomas Icard III and Lawrence Moss. 2014. Recent progress in monotonicity. *LiLT (Linguistic Issues in Language Technology)*, 9.

Michael Israel. 2011. *The grammar of polarity: Pragmatics, sensitivity, and the logic of scales*, volume 127. Cambridge University Press.

Jaap Jumelet. 2020. diagNNose: A library for neural activation analysis. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 342–350, Online. Association for Computational Linguistics.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do Language Models Understand Anything? On the Ability of LSTMs to Understand Negative Polarity Items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nirit Kadmon and Fred Landman. 1993. Any. *Linguistics and philosophy*, 16(4):353–422.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Manfred Krifka. 1995. The semantics and pragmatics of polarity items. *Linguistic analysis*, 25(3-4):209–257.

William Ladusaw. 1979. *Polarity Sensitivity as Inherent Scope Relations*. Ph.D. thesis, University of Texas Austin.

Utpal Lahiri. 1998. Focus and negative polarity in Hindi. *Natural Language Semantics*, 6:57–123.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in LSTM language models. 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 11–20, Minneapolis, Minnesota. Association for Computational Linguistics.

Tal Linzen, Grzegorz Chrupała, and Afra Alishahi. 2018. Proceedings of the 2018 emnlp workshop blackboxnlp: Analyzing and interpreting neural networks for nlp. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes, editors. 2019. *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Florence, Italy.

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.

Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. 2021. Predicting inductive biases of pre-trained models. In *International Conference of Learning Representations (ICLR)*.

Rebecca Marvin and Tal Linzen. 2018. Targeted Syntactic Evaluation of Language Models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Stroudsburg, PA, USA. Association for Computational Linguistics.

T. Reinhart. 1976. *The Syntactic Domain of Anaphora*. MIT Linguistics Dissertations. Massachusetts Institute of Technology.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with svcca. 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 3257–3267.

Víctor Manuel Sánchez Valencia. 1991. *Studies on natural logic and categorial grammar*. Universiteit van Amsterdam.

Johan Van Benthem. 1995. *Language in Action: categories, lambdas and dynamic logic*. MIT Press.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196.

Alex Warstadt and Samuel R Bowman. 2020. Can neural networks acquire a structural bias from raw linguistic data? In *Proceedings of the 42th annual conference of the Cognitive Science Society*.

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.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Trans. Assoc. Comput. Linguistics*, 8:377–392.

Lucas Weber, Jaap Jumelet, Elia Bruni, and Dieuwke Hupkes. 2021. Language modelling as a multi-task problem. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2049–2060. Association for Computational Linguistics.

Ethan Wilcox, Peng Qian, Richard Futrell, Miguel Ballesteros, and Roger Levy. 2019. Structural Supervision Improves Learning of Non-Local Grammatical Dependencies. In *Proceedings of North American Association for Computational Linguistics (NAACL)*, pages 3302–3312.

Zhaofeng Wu, Hao Peng, and Noah A. Smith. 2020. Infusing finetuning with semantic dependencies. *CoRR*, abs/2012.05395.

Frans Zwarts. 1995. Nonveridical contexts. *Linguistic Analysis*, 25:286–312.

A Filtered NPIs

We here present the full list of NPIs that were used for filtering sentences from the Full corpus, resulting in the Full\NPI corpus. The method for selecting these expressions is described in Section 5.4.

A damn, any, any longer, any old, anybody, anymore, anyone, anything, anything like, anytime soon, anywhere, anywhere near, as yet, at all, avail, by much, can possibly, could possibly, ever, in any, in days, in decades, in minutes, in years, just any, just yet, let alone, much help, nor, or anything, set foot, squat, such thing, that many, that much, that often, the slightest, whatever, whatsoever, yet.

This resulted in a reduction of 75.062 sentences out of the 3.052.726 sentences in the original Full corpus (2.46%).