Neural Natural Language Inference Models Partially Embed Theories of Lexical Entailment and Negation

Atticus Geiger
Stanford University
atticusg@stanford.edu

Kyle Richardson
Allen Institute for AI
kyler@allenai.org

Christopher Potts
Stanford University
cgpotts@stanford.edu

Abstract

We address whether neural models for Natural Language Inference (NLI) can learn the compositional interactions between lexical entailment and negation, using four methods: the *behavioral* evaluation methods of (1) challenge test sets and (2) systematic generalization tasks, and the *structural* evaluation methods of (3) probes and (4) interventions. To facilitate this holistic evaluation, we present Monotonicity NLI (MoNLI), a new naturalistic dataset focused on lexical entailment and negation. In our behavioral evaluations, we find that models trained on general-purpose NLI datasets fail systematically on MoNLI examples containing negation, but that MoNLI fine-tuning addresses this failure. In our structural evaluations, we look for evidence that our top-performing BERT-based model has learned to implement the monotonicity algorithm behind MoNLI. Probes yield evidence consistent with this conclusion, and our intervention experiments bolster this, showing that the causal dynamics of the model mirror the causal dynamics of this algorithm on subsets of MoNLI. This suggests that the BERT model at least partially embeds a theory of lexical entailment and negation at an algorithmic level.

1 Introduction

Natural Language Inference (NLI) keys into fundamental aspects of how people reason with language. Although NLI is generally cast in informal terms that embrace the indeterminacy of such reasoning, the task nonetheless manifests a number of very predictable reasoning patterns. For example, systematic manipulations of the lexical meanings (Glockner et al., 2018), syntactic constructions (Nie et al., 2019a), and contextual assumptions (Pavlick and Callison-Burch, 2016) have systematic effects on the correct labels. These patterns present crisp, motivated learning targets that we can leverage to

not only evaluate the ability of NLI models to learn robust solutions, but also to analyze the internal dynamics of successful models.

In this paper, our learning target concerns the role of *monotonicity* in NLI (MacCartney, 2009; Icard and Moss, 2013). Specifically, we would like to determine whether models can learn to represent lexical relations and accurately model that negation reverses entailment relations (e.g., *dance* entails *move*, but *not move* entails *not dance*). This property of negation is *downward monotonicity*.

In service of pursuing this question, we present Monotonicity NLI (MoNLI), a new naturalistic NLI dataset for training and assessing systems on these semantic notions (Section 3). MoNLI extends SNLI (Bowman et al., 2015) to provide comprehensive coverage of examples that depend on lexical reasoning with and without negation. Using MoNLI, we conduct both behavioral and structural evaluations, seeking to provide a detailed picture of the solutions that top-performing models learn. We evaluate Enhanced Sequential Inference Models (Chen et al., 2016) and BERT-based models (Devlin et al., 2019), along with standard baselines.

Previous work evaluating the ability of neural models to learn monotonicity has focused on challenge test sets and systematic generalization tasks (Yanaka et al., 2019b,a; Geiger et al., 2019; Richardson et al., 2019). These behavioral evaluations ask whether models achieve a desired input–output behavior. We employ these methods as well, but we also ask whether models achieve an *algorithmic-level* learning target, in the terms of Marr (1982). Monotonicity reasoning can be cast as an algorithm that solves MoNLI perfectly. Do neural models implement this algorithm?

We first report on two behavioral evaluations (Section 5). When MoNLI is used as a challenge test set, we find that models trained on SNLI and/or MNLI (Williams et al., 2018) fail to reason with lex-

ical entailments when negation is involved. However, we trace these failures to gaps in the training data. In response, we pose a systematic generalization task in which we expose models to MoNLI examples through fine-tuning while still requiring them to generalize to entirely new pairs of lexical items in negated linguistic contexts at test time. All our models solve the task, which suggests that they have learned general theories of lexical entailment and negation.

We then report on structural evaluations (Section 6), seeking to determine whether our top-performing BERT-based models implement the target monotonicity algorithm. In probing experiments, we find evidence consistent with this result, but it's not conclusive, since probes alone cannot reveal a model's causal dynamics. However, our intervention experiments provide evidence that BERT does mirror the causal dynamics of the monotonicity algorithm, at least on large subsets of MoNLI. We conclude that this model at least partially embeds a theory of lexical entailment and negation at an algorithmic level, in addition to fully achieving the correct input–output behavior on MoNLI.

2 Related work

Monotonicity Our empirical focus is entailment and negation. This is one (highly prevalent) aspect of monotonicity reasoning, which governs many aspects of lexical and constructional meaning in natural language (Sánchez-Valencia, 1991; van Benthem, 2008). There is an extensive literature on monotonicity logics (Moss, 2009; Icard, 2012; Icard and Moss, 2013; Icard et al., 2017). Within NLP, MacCartney and Manning (2008, 2009) apply very rich monotonicity algebras to NLI problems, Hu et al. (2019a,b) create NLI models that use polarity-marked parse trees, and Yanaka et al. (2019a,b) and Geiger et al. (2019) investigate the ability of neural models to understand natural logic reasoning. While we consider only a small fragment of these approaches, the methods we develop should apply to more complex systems as well.

Challenge Test Sets Challenge[1] test sets are supplementary evaluation resources that test the ability of a model to generalize to examples outside the dis-

tribution of the data it was trained, developed, and (standardly) tested on. These tests probe the generalization capabilities of state-of-the-art models with respect to the tasks they have been trained on, by focusing on difficult or underrepresented examples in a model's training set (Jia and Liang, 2017; Naik et al., 2018; Glockner et al., 2018; Richardson et al., 2019; Talmor et al., 2019).

Systematic Generalization Tasks Fodor and Pylyshyn (1988) offer *systematicity* as a hallmark of human cognition. Systematicity says that certain behaviors are intrinsically connected to others by compositional structures. For example, understanding *the puppy loves Sandy* is intrinsically connected to understanding *Sandy loves the puppy*. For Fodor and Pylyshyn, these observations trace to the mind's ability to recombine known parts and rules. There are often strong intuitions that certain generalization tasks are only solved by models with systematic structures. These tasks are referred to as *systematic generalization tasks* (Lake and Baroni, 2018; Hupkes et al., 2019; Yanaka et al., 2020; Bahdanau et al., 2018; Geiger et al., 2019; Goodwin et al., 2020).

Probing Probes are supervised learning models trained to extract information from representations created by another model. They are a primary tool in the analysis of neural network models (Peters et al. 2018; Tenney et al. 2019; Clark et al. 2019; for a full review, see Belinkov and Glass 2019). In aggregate, this work has provided nuanced insights into the internal representations of these models, as well as their capacity to directly support learning diverse NLP tasks via fine-tuning (Hewitt and Liang, 2019). However, probes are only able to reveal how representations correlate with information. They cannot determine if that information plays a causal role in model predictions (Belinkov and Glass, 2019; Vig et al., 2020).

Interventions Intervention studies go beyond probing to make changes to the internal states of a network, with the goal of observing how those changes affect system outputs. Giulianelli et al. (2018) use probing results to make informed interventions during LSTM language model predictions to preserve information about the grammatical subject's number, and this led to improved performance in subject–verb agreement. Vig et al. (2020) use interventions to characterize how gender bias is represented in the internal causal structure

---

[1]Though *adversarial* and *challenge* are sometimes used synonymously, we opt for the term *challenge*, because our dataset was designed with the intention of evaluating whether a model learned a particular phenomenon, as opposed to breaking any particular model (cf. Nie et al. 2019b).

of a model, and find that a small number of synergistic neurons mediate gender bias. They also find that the effect of these neurons is roughly linearly separable from the effect of the remainder of the model, a remarkable finding considering the highly non-linear nature of neural networks.

3 Monotonicity NLI dataset

We created the MoNLI corpus to investigate the ability of NLI models to learn the compositional interactions between lexical entailment and negation. MoNLI contains 2,678 NLI examples in the usual format for NLI datasets like SNLI. In each example, the hypothesis is the result of substituting a single word w_p in the premise for a hypernym or hyponym w_h. We refer to w_h and w_p as the *substituted words* in an example. In 1,202 of these examples, the substitution is performed under the scope of the downward monotone operator *not*. Downward monotone operators reverse entailment relations: *dance* entails *move*, but *not move* entails *not dance*. We refer to these examples collectively as NMoNLI. In the remaining 1,476 examples, this substitution is performed under the scope of no downward monotone operator. We refer to these examples collectively as PMoNLI.

MoNLI was generated according to the following procedure. First, randomly select a premise or hypothesis sentence s from the SNLI training dataset. Second, select a noun in s, and, using WordNet (Fellbaum, 1998), select all hypernyms and hyponyms of the noun subject to two conditions: (1) the hypernym or hyponym appears in the SNLI training data, and (2) substituting the hypernym or hyponym results in a grammatical, coherent sentence s'. Finally, for each substitution, generate two examples for the corpus – one where the original sentence is the premise and the edited sentence is the hypothesis, and one example with those roles reversed. Each of these example pairs has one example with the label **entailment** and one example with the label **neutral**, resulting in a dataset perfectly balanced between the two labels.

For example, suppose we select the SNLI sentence (A) and we identify the noun *plants* for substitution. Then we enter *plants* into WordNet and find that *flowers* is a hyponym of *plants*, so we substitute *flowers* for *plants* to create the edited sentence (B):

(A) The three children are not holding **plants**.

$$\Downarrow$$

(B) The three children are not holding **flowers**.

This leads to two new MoNLI examples:

| (A) | **entailment** | (B) |
| (B) | **neutral** | (A) |

These two examples would belong to NMoNLI, due to *not* scoping over the substitution site. If *not* were removed from both of these sentences, then their labels would be swapped and both examples would belong to PMoNLI.

MoNLI was generated by the authors by hand; examples judged to be unnatural were removed, and any grammatical or spelling errors in the original SNLI sentence were corrected.

This data generation process is similar to that of Glockner et al. (2018), except they focus on the lexical relations of exclusion and synonymy, while we focus on entailment relations. This difference prevents their dataset from capturing monotonicity reasoning, which involves entailment relations, but not exclusion or synonymy.

4 Models

We evaluated four models on MoNLI:

CBOW The continuous bag of words baseline from Williams et al. (2018).

BiLSTM The bidirectional LSTM baseline from Williams et al. (2018).

ESIM The Enhanced Sequential Inference Model (Chen et al., 2016) is a hybrid TreeLSTM-based and biLSTM-based model that uses an inter-sentence attention mechanism to align words across sentences.

BERT A Transformer model trained to do masked language modeling and next-sentence prediction (Devlin et al., 2019). We rely on uncased BERT-base parameters from Hugging Face `transformers` (Wolf et al., 2019).

The first two models serve as baselines, while the other two models achieve comparable, near state-of-the-art scores on SNLI.

5 Behavioral Evaluations

5.1 MoNLI as a Challenge Test Set

We first use MoNLI as a challenge test dataset, i.e., models trained only on SNLI are expected to generalize to MoNLI. MoNLI can be considered a

Model	Input pretraining	NLI train data	No MoNLI fine-tuning			With NMoNLI fine-tuning	
			SNLI	PMoNLI	NMoNLI	SNLI	NMoNLI
CBOW	GloVe	SNLI train	78.9	64.6	22.9	65.9	95.5
BiLSTM	GloVe	SNLI train	81.6	73.2	37.9	74.6	93.5
ESIM	GloVe	SNLI train	87.9	86.6	39.4	56.9	96.2
ESIM	GloVe		–	–	–	–	98.0
ESIM			–	–	–	–	35.5
BERT	BERT	SNLI train	90.8	94.4	2.2	90.5	90.0
BERT	BERT		–	–	–	–	96.7
BERT			–	–	–	–	62.3

Table 1: The results of our behavioral analysis. The columns labeled *No MoNLI fine-tuning* display the challenge test set results (Section 5.1), and the columns labeled *With MoNLI fine-tuning* display systematic generalization task results (Section 5.2). The numbers are accuracy values; all the datasets have balanced label distributions. Dashes mark experiments that would involve untrained NLI parameters due to training/fine-tuning set-up.

challenge test dataset that evaluates an NLI model's ability to perform simple inferences founded in lexical entailments and monotonicity. As discussed in Section 3, it is not especially adversarial, in that we sampled sentences from the SNLI training set and only substituted in hypernyms and hyponyms that occur in the SNLI training set. This keeps MoNLI as close as possible to the distribution of SNLI. Thus, if a model fails on MoNLI, we can be confident that this failure stems from a lack of knowledge about monotonicity and lexical entailment relations, rather than some other confounding factor like syntactic structures or vocabulary items that were unseen in training.

5.1.1 Results

The results are in Table 1 under the heading 'No MoNLI fine-tuning', and they are stark. The four models achieve comparably high accuracies on SNLI and PMoNLI, the examples where no downward monotone operators scope over the substitution site. However, they are well below chance accuracy on NMoNLI, the examples where *not* scopes over the substitution site. BERT is more extreme than the other models, achieving a higher accuracy on PMoNLI than SNLI and almost zero accuracy on NMoNLI. High performance on PMoNLI shows that models have knowledge of the lexical relations between the substituted words, but low performance on NMoNLI shows the models have no knowledge of the downward monotone nature of *not*. In fact, the below chance accuracy on NMoNLI indicates that these models are somewhat reliably (incredibly reliably in BERT's case) predicting the wrong label on these examples, suggesting that they treat NMoNLI examples the same as PMoNLI examples.

5.1.2 Discussion

While these models trained on SNLI do not know that *not* is downward monotone in these examples, this is not conclusive evidence that they are unable to learn this semantic property. This ability might not be necessary for success on SNLI, where only 38 examples have negation in both the premise and hypothesis. A natural next step is to train on MNLI, where the coverage with regard to negation is better: about 18K examples ($\approx$4%) have negation in the premise and hypothesis. We tried this, by combining MNLI with SNLI, and the results were almost exactly the same. However, even the MNLI examples might not manifest the kind of monotonicity reasoning that we are targeting. Our next experiments help to resolve this issue.

5.2 A Systematic Generalization Task

Our three models trained on SNLI have knowledge of the lexical relations between substituted words, but do not know that the presence of *not* reverses the relationship between the word-level relation and the sentence-level relation. We now conduct a behavioral evaluation to determine whether models are able to learn a general theory of lexical entailment and negation when exposed to a limited subset of NMoNLI during training.

In designing systematic generalization tasks, we seek to constrain the training data in ways that prevent unsystematic models from succeeding. Defining disjoint train/test splits is enough to foil truly unsystematic models (e.g., simple look-up tables). However, building on much previous work (Lake and Baroni, 2018; Hupkes et al., 2019; Yanaka et al., 2020; Bahdanau et al., 2018; Goodwin et al., 2020; Geiger et al., 2019), we contend that a randomly constructed disjoint train/test split only diag-

noses the most basic level of systematicity. More difficult systematic generalization tasks will only be solved by models exhibiting more complex compositional structures. Specifically, we want our systematic generalization task to be solved only by models that compute lexical entailment relations that may be reversed by negation. A learning model that memorizes labels based on substituted word pairs and whether negation is present would succeed on a disjoint train and test set as long as all pairs of substituted words appear during training, and this model does not compute the lexical relation between word pairs.

As such, we propose a generalization task where NMoNLI is partitioned into train and test sets such that the substituted words in the train set and the substituted words in the test sets are disjoint.[2] The specific train/test split we used is described in Appendix A.1. Ideally, a model trained on SNLI that is further trained on NMoNLI will still maintain strong performance on SNLI. We use inoculation by fine-tuning (Liu et al., 2019) to evaluate models on this ability. We report on the inoculated model with the highest average performance on SNLI test and NMoNLI test (full details of the inoculation process are in Appendix A.2).

The models are evaluated on examples where they know the relation between the substituted words, as evidenced by high performance on PMoNLI, but have not seen those substituted words in the presence of negation during training. However, they have seen other substituted words with the same relation in the presence of negation during training, making this task *hard*, but *fair* (Geiger et al., 2019). To solve this harder generalization task, we believe a model must learn to reverse the lexical relation *in general*; the identity of the substituted words must be abstracted away.

5.2.1 Results and Discussion

We present our results in Table 1, under the heading 'With NMoNLI fine-tuning'. All of our models solve this generalization task. However, only BERT does so while maintaining high performance on SNLI. We also report ablation studies on our two non-baseline models, evaluating their performance on our systematic generalization task without training on SNLI and without any pretraining at all. We find that both models still succeed with no pre-

---

[2]We use only NMoNLI in our systematic generalization task because models trained on SNLI already achieve high performance on PMoNLI.

INFER(*MoNLIexample*)
1   *lexrel* ← GET-LEX-REL(*MoNLIexample*)
2   **if** CONTAINS-NOT(*MoNLIexample*)
3   **return** REVERSE(*lexrel*)
4   **return** *lexrel*

Figure 1: An algorithm able to solve the MoNLI dataset that provides a theoretically motivated learning target for neural models at an algorithmic level of analysis (Marr, 1982). INFER takes in an example from MoNLI and outputs the relation between the premise and hypothesis. It uses three predefined functions. GET-LEX-REL returns the relation (one of $\{\sqsupset, \sqsubset\}$) between the substituted words in the premise and hypothesis. CONTAINS-NOT returns true iff negation is present. REVERSE maps $\sqsubset$ to $\sqsupset$ and vice-versa.

training on SNLI, but fail with no pretraining whatsoever. This suggests that BERT pretraining and GloVe vectors both provide sufficient information about lexical relations for the models to succeed. BERT's ability to get slightly above chance performance with no pretraining indicates the presence of some statistical artifacts in our dataset (Gururangan et al., 2018).

In sum, our models were able to solve our systematic generalization task, which we believe to be evidence that they learn to compute the lexical relations between substituted words. However, we also believe this evidence is weak, as there is no formal relationship between a model solving a generalization task and that model having any particular systematic internal structures. This evaluation is fundamentally behavioral, only concerning model inputs and outputs. We believe that a structural evaluation is necessary to conclusively evaluate systematicity.

6 Structural Evaluations

In our behavioral evaluations, the learning target was to mimic the input–output behavior defined by MoNLI. Assessing this learning target is straightforward. We now report on structural evaluations to try to determine whether a neural model has particular internal dynamics. For this, we rely on very recent probing and intervention methodologies that are not yet well understood and must be tailored to the model being analyzed. As such, we choose to focus on a single model, namely, the BERT model from Section 5 fine-tuned on NMoNLI. We chose BERT because it achieved exceptional results on
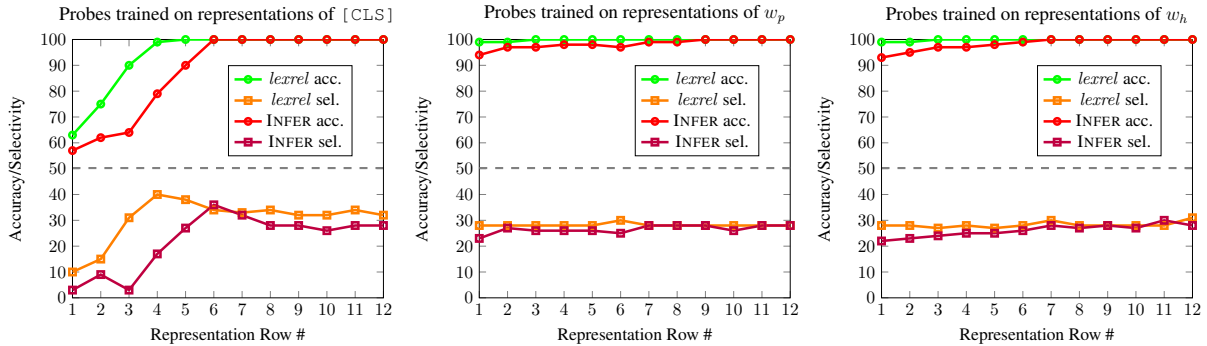
Figure 2: Results where classifier probes are trained on BERT representations to predict the value of *lexrel* and the output of INFER (Figure 1). Selectivity is probe accuracy minus control probe accuracy (Hewitt and Liang, 2019). The grey dotted line provides a soft ceiling for selectivity values, because we expect control probes trained on a binary task to at least achieve chance accuracy.

NMoNLI after fine-tuning without experiencing a significant drop on SNLI.

Figure 1 presents the simple algorithm INFER, which is our learning target. It takes in a MoNLI example and stores the lexical entailment relation between the substituted words in the variable *lexrel*. If negation is present, the reverse of *lexrel* is returned; if there is no negation, *lexrel* itself is returned. This is simply an algorithmic description of the MoNLI construction method. The most important piece is the intermediate variable *lexrel*. Intuitively, if our BERT model implements this algorithm, there will be some representation in BERT that *stores lexrel* and BERT will *use* that representation for a final prediction. Probes can give us an idea of where information is stored, and interventions help us see how that information is used.

Before we can go looking for where BERT stores and uses *lexrel*, we must limit ourselves to a tractable number of model internal representations. When our BERT model processes an example from MoNLI, it is tokenized as

$$e = \langle [\texttt{CLS}], p, [\texttt{SEP}], h, [\texttt{SEP}] \rangle$$

and 12 rows of vector representations are created, so each token is associated with 12 vectors. We localize our efforts to the representations created for [CLS] and the tokens for the substituted words in the premise and hypothesis, w_p and w_h (as described in Section 3). This narrows our search to 36 possible vector locations where BERT could be storing the variable *lexrel* for use in final output prediction. We denote these 36 locations with $\text{BERT}^r_{w_p}$, $\text{BERT}^r_{w_h}$, and $\text{BERT}^r_{[\texttt{CLS}]}$ where r is a row $(1 \leqslant r \leqslant 12)$.

6.1 Probes

We follow Hupkes et al. (2018) in using probing evidence to determine whether a neural model stores the same information as a symbolic algorithm. They used probes to predict variable values used in an algorithm from the hidden states of sequential recurrent networks trained to perform basic arithmetic. We do something similar, probing the 36 vector locations defined by $\text{BERT}^r_{w_p}$, $\text{BERT}^r_{w_h}$, and $\text{BERT}^r_{[\texttt{CLS}]}$ for the value of the variable *lexrel* and the output of INFER.

Hewitt and Liang (2019) argue that accuracy is a poor metric for probes and that the ideal probe will highly *selective*, that is, it will have high accuracy on a linguistic task but low accuracy on a control task where inputs are given random labels. In this setting, our linguistic tasks are predicting the value of *lexrel* and the output of INFER from a model-internal vector created by BERT for some MoNLI example. Our control task is identical, except labels are randomly assigned to inputs. Hewitt and Liang demonstrate that small, linear probes result in high selectivity. Following this guidance, we used a linear classifier with 4 hidden units that was trained and evaluated on all of MoNLI.

Our probing results are summarized in Figure 2. Probes were able to achieve high accuracy and high selectivity predicting the output of INFER at every location other than the locations $\text{BERT}^k_{[\texttt{CLS}]}$ where $1 \leq k \leq 4$, and high accuracy and high selectivity predicting the value of *lexrel* at every location other than $\text{BERT}^1_{[\texttt{CLS}]}$ and $\text{BERT}^2_{[\texttt{CLS}]}$.

This qualitative picture is compatible with a story where BERT stores the value of *lexrel* at any location other than $\text{BERT}^1_{[\texttt{CLS}]}$ or $\text{BERT}^2_{[\texttt{CLS}]}$ and then uses this information to compute a final output

prediction at any location other than the locations $\text{BERT}^k_{\text{[CLS]}}$ where $1 \leq k \leq 4$. The fact that probes trained on the vectors at locations $\text{BERT}^3_{\text{[CLS]}}$ or $\text{BERT}^4_{\text{[CLS]}}$ have high accuracy and selectivity predicting the value of *lexrel*, but moderate accuracy and low selectivity predicting the output of INFER may suggest a more specific story where these two locations store the value of the variable *lexrel* before this information is used to compute the final output.

We emphasize that, while the probing results are compatible with these stories, they only provide conclusive evidence about how representations correlate with the value of *lexrel* and the output of INFER. They cannot determine whether this information plays a causal role in model predictions (Belinkov and Glass, 2019; Vig et al., 2020).

6.2 Interventions

Probes give us a picture of where information is stored by our BERT model, but they cannot determine whether that information is used to make final predictions. Interventions can help us address this deeper question. As discussed above, our algorithmic-level learning target is for BERT to mimic the dynamics of the algorithm INFER in Figure 1. Icard (2017) provided the insight that algorithms like INFER can be explicitly understood as causal models (Pearl, 2001). This means that the causal role of *lexrel*, the lone variable in INFER, can be characterized with counterfactual claims about how altering the value of the variable would cause output behavior to change.

Suppose INFER is run on a MoNLI example i. Let $lexrel(i) \in \{\sqsupseteq, \sqsubseteq\}$ be the value that *lexrel* takes on, and let $\text{INFER}(i) \in \{\sqsupseteq, \sqsubseteq\}$ be the output. Then INFER can be see as providing the following counterfactual characterization of *lexrel*: if the value of *lexrel* were changed from $lexrel(i)$ to $lexrel(j)$, where j is a second MoNLI example, then $\text{INFER}(i)$ would change to

$$\text{INFER}_{lexrel(i) \to lexrel(j)}(i) =$$
$$\begin{cases} \text{INFER}(i) & lexrel(i) = lexrel(j) \\ \text{REVERSE}(\text{INFER}(i)) & lexrel(i) \neq lexrel(j) \end{cases}$$

In other words, if *lexrel* were to take on the opposite value, then the output would also take on the opposite value.

Our analytic tool for evaluating whether such causal dynamics are present in BERT is the *interchange intervention*. Figure 3 provides a high-level
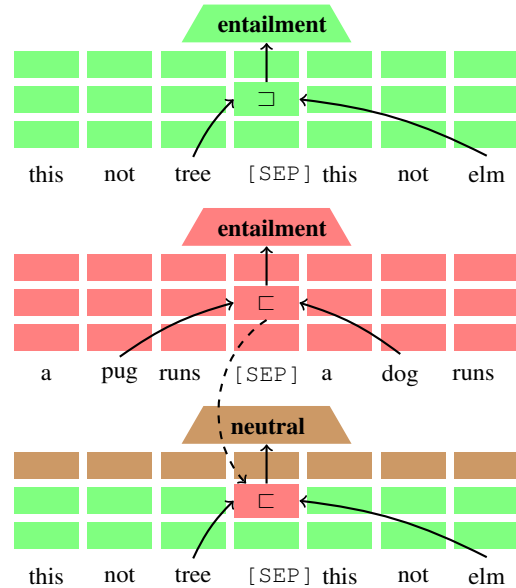
Figure 3: An illustrative **interchange intervention**: The solid arrows represent a hypothesis about where the model stores and uses information about lexical entailment. The dotted arrow is an interchange intervention, where the green vector (top) we think stores reverse entailment, trees $\sqsupseteq$ elms, is interchanged with the red vector (middle) we think stores forward entailment, pugs $\sqsubseteq$ dogs, leading to a modified network (bottom). If our hypothesis is correct, then the output should change from **entailment** to **neutral**, because the negation in the green example reverses the relationship between lexical entailment and sentence-level entailment. If this label reversal is not observed, crucial entailment information must lie elsewhere in the network.

picture of how these experiments work, and the following definition seeks to make this more precise and general:

Interchange Intervention Let L be one of the 36 locations defined by $\text{BERT}^r_{w_p}$, $\text{BERT}^r_{w_h}$, and $\text{BERT}^r_{\text{[CLS]}}$. When BERT is making a prediction for i, suppose that the vector created at location L on input i is replaced with the vector created at location L on input j and this results in the output y. We say that y is the result of an interchange intervention from i to j at location L and denote this output as $\text{BERT}_{L(i) \to L(j)}(i)$.

In essence, $\text{BERT}_{L(i) \to L(j)}(i)$ characterizes the output behavior that results from an experiment where model-internal vectors are interchanged at location L. Recall that $\text{INFER}_{lexrel(i) \to lexrel(j)}(i)$ describes what output is provided by INFER if variables are interchanged. If for some subset of MoNLI S, we believe that BERT is both storing the value of *lexrel* at some location L and using

that information to make a final prediction, then for all $i, j \in S$ the following should hold:

$$\text{INFER}_{lexrel(i) \rightarrow lexrel(j)}(i) = \text{BERT}_{L(i) \rightarrow L(j)}(i)$$

This amounts to observing that the variables in the algorithm and the vectors in the model satisfy the same counterfactual claims. When a vector representing forward entailment is interchanged with a different vector representing forward entailment, model output behavior should be unchanged. If a vector representing forward entailment is interchanged with a different vector representing reverse entailment, then the model output should be reversed.

Results Due to computational constraints, we randomly conducted interchange experiments at our 36 different locations and chose the location with the most promise, namely, $\text{BERT}^3_{w_h}$. (Appendix A.3 covers our selection methodology in detail.) We conducted $\approx$7 million interchange experiments at this location, one experiment for every pair of examples in MoNLI. Using a simple greedy algorithm, we discovered several large subsets of MoNLI where BERT mimics the causal dynamics of INFER. (The greedy algorithm is described in Appendix A.3.) These subsets have size 98, 63, 47, and 37, and for each of these subsets there are many pairs of examples with interchange experiments that had a causal impact on the final model prediction. To put these results in context, if interchange experiments had a random effect on model output, then the expected number of subsets larger than 20 with this property would be less than 10^{-8}.

Discussion These results show that the values assigned by the algorithm INFER to the variable *lexrel* and the vectors created by BERT at the location $\text{BERT}^3_{w_h}$ exhibit the same causal dynamics on four large subsets of MoNLI. In Appendix A.3 we show a visualization of the subset with 98 examples. These pairs contain only 13 of the 69 distinct hyponyms in MoNLI, which makes it clear that this subset of MoNLI is not a random sample, but rather reflects a coherent semantic space. From this we conclude that, in addition to capturing the input–output behavior described by MoNLI, our BERT model at least partially embeds a theory of lexical entailment and negation at an algorithmic level of analysis.

Importantly, these results do not show that BERT fails to mimic the causal dynamics of INFER on larger subsets of MoNLI. First, we only conducted interchange experiments for every pair of examples in MoNLI at the location $\text{BERT}^3_{w_h}$. Second, we did not consider the possibility that BERT stores and uses the value of *lexrel* at different locations, depending on which input is provided. Third, analyzing vector representations may be too coarse-grained; perhaps experiments will need to be done on individual vector units. Finally, we used a greedy algorithm to discover the four subsets of MoNLI. We did not exhaustively analyze BERT to find the largest subset of MoNLI on which it mimics the causal dynamics of INFER; such an analysis is likely computationally impossible. What we did do is perform an efficient analysis that was able to find several large subsets of MoNLI on which the desired causal dynamics are present.

7 Conclusion

To operationalize our research question of whether neural NLI models can learn the compositional interactions between lexical entailment and negation, we constructed two learning targets for neural NLI models: (1) learn the input–output behavior described by MoNLI and (2) acquire the internal dynamics of the algorithm INFER. We evaluated the first learning target with two behavioral evaluation methods, using challenge datasets to show that state-of-the-art models trained on general-purpose NLI datasets fail to exhibit the correct behavior when negation is present and then following up with a systematic generalization task that showed our models are able to learn the correct input–output behavior when fine-tuned on a limited, but sufficient, subset of NMoNLI. We evaluated the second learning target with two structural evaluation methods, using probes to investigate where information about the variable *lexrel* from INFER might be stored in a BERT model and using interventions to show that on some subsets of MoNLI our BERT model exhibits the same causal dynamics as the algorithm INFER.

We believe that our holistic evaluation, leveraging both behavioral and structural methods, provides a multifaceted picture of how neural NLI models treat lexical entailment and negation. While our interchange intervention methodology is not yet formally grounded, there is great promise in the idea of investigating whether a neural model mirrors the causal dynamics of an algorithm.

References

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. 2018. Systematic generalization: What is required and can it be learned? In *In Proceedings of the 6th International Conference on Learning Representations*, Beijing.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Johan van Benthem. 2008. A brief history of natural logic. In *Logic, Navya-Nyaya and Applications: Homage to Bimal Matilal*.

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.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, and Hui Jiang. 2016. Enhancing and combining sequential and tree LSTM for natural language inference. *CoRR*, abs/1609.06038.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

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.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.

Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71.

Atticus Geiger, Ignacio Cases, Lauri Karttunen, and Christopher Potts. 2019. Posing fair generalization tasks for natural language inference. 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 4485–4495, Hong Kong, China. Association for Computational Linguistics.

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.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Emily Goodwin, Koustuv Sinha, and Timothy J. O'Donnell. 2020. Probing linguistic systematicity.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, 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, Hong Kong, China. Association for Computational Linguistics.

Hai Hu, Qi Chen, and Larry Moss. 2019a. Natural language inference with monotonicity. In *Proceedings of the 13th International Conference on Computational Semantics - Short Papers*, pages 8–15, Gothenburg, Sweden. Association for Computational Linguistics.

Hai Hu, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S. Moss, and Sandra Kübler. 2019b. MonaLog: A lightweight system for natural language inference based on monotonicity. *ArXiv*, abs/1910.08772.

Dieuwke Hupkes, Sanne Bouwmeester, and Raquel Fernández. 2018. Analysing the potential of seq-to-seq models for incremental interpretation in task-oriented dialogue. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 165–174, Brussels, Belgium. Association for Computational Linguistics.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2019. Compositionality decomposed: how do neural networks generalise?

Thomas Icard, Lawrence Moss, and William Tune. 2017. A monotonicity calculus and its completeness. In *Proceedings of the 15th Meeting on the Mathematics of Language*, pages 75–87, London, UK. Association for Computational Linguistics.

Thomas F. Icard. 2012. Inclusion and exclusion in natural language. *Studia Logica*, 100(4):705–725.

Thomas F. Icard. 2017. From programs to causal models. In *Proceedings of the 21st Amsterdam Colloquium*, pages 35–44. University of Amsterdam.

Thomas F. Icard and Lawrence S. Moss. 2013. Recent progress on monotonicity. *Linguistic Issues in Language Technology*, 9(7):1–31.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *CoRR*, abs/1707.07328.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2879–2888. PMLR.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019. 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.

Bill MacCartney. 2009. *Natural Language Inference*. Ph.D. thesis, Stanford University.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 521–528, Manchester, UK. Coling 2008 Organizing Committee.

Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *Proceedings of the Eight International Conference on Computational Semantics*, pages 140–156, Tilburg, The Netherlands. Association for Computational Linguistics.

David Marr. 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., New York, NY, USA.

Lawrence S Moss. 2009. Natural logic and semantics. In *Proceedings of the 18th Amsterdam Colloquium: Revised Selected Papers*, pages 71–80, Berlin. University of Amsterdam, Springer.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yixin Nie, Yicheng Wang, and Mohit Bansal. 2019a. Analyzing compositionality-sensitivity of NLI models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6867–6874.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019b. Adversarial NLI: A new benchmark for natural language understanding.

Ellie Pavlick and Chris Callison-Burch. 2016. Most "babies" are "little" and most "problems" are "huge": Compositional entailment in adjective-nouns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2164–2173, Berlin, Germany. Association for Computational Linguistics.

Judea Pearl. 2001. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, page 411–420, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.

Kyle Richardson, Hai Hu, Lawrence S. Moss, and Ashish Sabharwal. 2019. Probing natural language inference models through semantic fragments.

Víctor Sánchez-Valencia. 1991. *Studies in Natural Logic and Categorial Grammar*. Ph.D. thesis, University of Amsterdam.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2019. olmpics – on what language model pre-training captures.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Causal mediation analysis for interpreting neural nlp: The case of gender bias.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. 2020. Do neural models learn systematicity of monotonicity inference in natural language?

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019a. Can neural networks understand monotonicity reasoning? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 31–40, Florence, Italy. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019b. HELP: A dataset for identifying shortcomings of neural models in monotonicity reasoning. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 250–255, Minneapolis, Minnesota. Association for Computational Linguistics.