

A Appendices

A.1 Train–Test Split for Systematic Generalization Task

In our systematic generalization task, NMoNLI is partitioned into train, dev, and test sets such that the substituted words in the train set and the substituted words in the dev and test sets are disjoint. The specific train/test split we used is described in Table 2.

NMoNLI Train		NMoNLI Test	
person	198	dog	88
instrument	100	building	64
food	94	ball	28
machine	60	car	12
woman	58	mammal	4
music	52	animal	4
tree	52		
boat	46		
fruit	42		
produce	40		
fish	40		
plant	38		
jewelry	36		
anything	34		
hat	20		
man	20		
horse	16		
gun	12		
adult	10		
shirt	8		
shoe	6		
store	6		
cake	4		
individual	4		
clothe	2		
weapon	2		
creature	2		

Table 2: The hyponyms that occur in the train-test split of NMoNLI described in Section 5.2. The number next to each hyponym corresponds to the number of examples that hyponym occurs in.

A.2 Further Details of Inoculation

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. In this method, a pretrained model is

further fine-tuned on different small amounts of adversarial data while performance on the original dataset and the adversarial dataset is tracked. For each amount of adversarial data, a hyperparameter search is run and the model with the highest average performance on the original dataset and adversarial dataset is selected. Optimizing for the average accuracy is what Richardson et al. (2019) refer to as *lossless inoculation*, and we perform the same hyperparameter searches that they do. The results of our inoculation experiments are shown in Figure 4. The results in Table 1 under the heading ‘With NMoNLI fine-tuning’ are from the inoculated model with the highest average performance on SNLI test and NMoNLI test.

A.3 Further Details of Interventions

We say that that BERT mimics the causal dynamics of INFER if there is a map L from MoNLI examples to model-internal vectors in BERT such that the model internal-vectors satisfy the counterfactual claims ascribed to the variable $lexrel$. Intuitively, L is a hypothesis about where BERT stores the value of $lexrel$ for different examples. Our analytic tool for evaluating a map L is the *interchange intervention*:

Consider inputs i and j and some map from inputs to model-internal vectors L . Suppose that, when BERT is making a prediction for i , the vector $L(i)$ is replaced with the vector $L(j)$ resulting in output y . We say that y is the result of an interchange intervention from i to j under map L and denote this output as $BERT_{L(i) \rightarrow L(j)}(i)$.

In essence, $BERT_{L(i) \rightarrow L(j)}(i)$ characterizes the output behavior that results from an experiment where model-internal vectors are interchanged. Recall that $INFER_{lexrel(i) \rightarrow lexrel(j)}(i)$ describes what output is provided by INFER if variables are interchanged. Thus, we can say that BERT *implements* the algorithm INFER over a set of examples S if, for all $i, j \in S$, the following equality holds:

$$INFER_{lexrel(i) \rightarrow lexrel(j)}(i) = BERT_{L(i) \rightarrow L(j)}(i)$$

This amounts to observing that the variables in the algorithm and the vectors in the model satisfy the same counterfactual claims.

In the case when S has only two elements i and j , we write $\mathcal{X}(i, j)$. For some map L , if $\mathcal{X}(i, j)$ holds for every pair of inputs i and j in MoNLI, then BERT mimics the causal dynamics of INFER on the entirety of MoNLI.

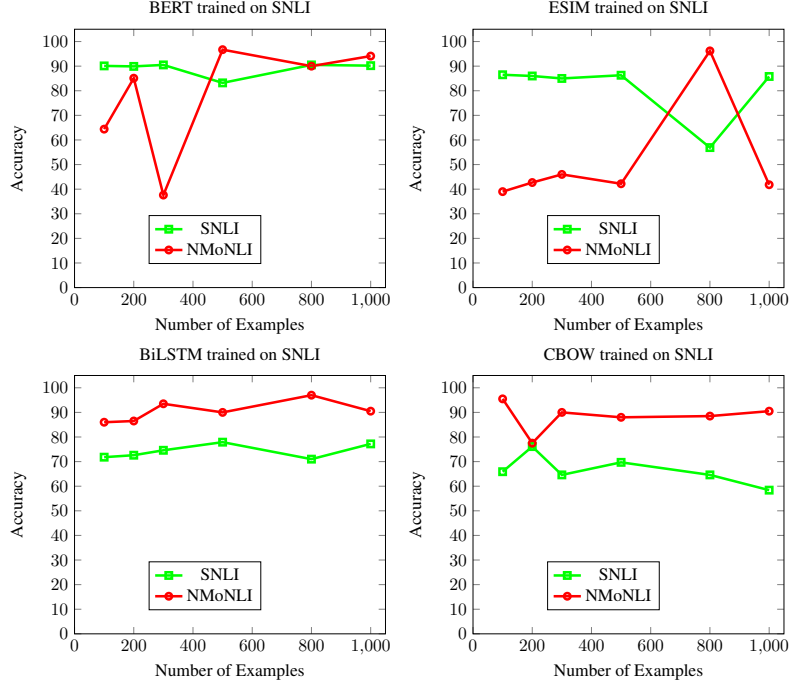


Figure 4: Inoculation results for our four models performing our systematic generalization task.

There are a multitude of possible maps L , and MoNLI has $\approx 2,000$ examples, so 7 million interchange interventions must be conducted to verify that BERT mimics the causal dynamics of INFER under some map. As such, we must make some assumptions to narrow down our space of possible maps.

When our BERT model processes an example from MoNLI, it is tokenized as

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

and 12 rows of vector representations are created, so each token is associated with 12 vectors. In order to efficiently find an appropriate map L , we localize our efforts to the representations created for $[\text{CLS}]$ and the tokens for the substituted words in the premise and hypothesis, w_p and w_h . We additionally assume that every example is mapped to a vector at the same location. This narrows our search to 36 possible maps from inputs in MoNLI to model-internal vectors. For row r , we call these $\text{BERT}_{w_p}^r$, $\text{BERT}_{w_h}^r$, and $\text{BERT}_{[\text{CLS}]}^r$.

Since we must make so many assumptions, we may only be able to find a map that shows $\mathcal{X}(i, j)$ holds for all i and j in some subset of MoNLI, but not the entirety of MoNLI. Crucially, though, this subset of MoNLI still must contain both lexical relations \sqsupset and \sqsubset for mimicking the causal dynamics of INFER to not be vacuous. If one lexical relation

is entirely missing from the subset, then none of the interchanges between model vectors will change the output behavior, so there is no guarantee that these vectors play any role in determining output behavior.

As such, we seek the largest subset of MoNLI containing both lexical relations on which BERT implements a modular representation of lexical entailment. To quantify this, we create a graph in which the examples of MoNLI are the nodes and there is an edge between two nodes n_i and n_j if and only if $\mathcal{X}(i, j)$ holds. Cliques in this graph will, in turn, correspond to subsets of MoNLI on which BERT mimics the causal dynamics of INFER. We denote the graph for the map BERT_t^r as \mathcal{G}_t^r for any row r and token $t \in \{[\text{CLS}], w_p, w_h\}$.

To see the intuition behind this graph, it is helpful to consider some logically possible scenarios. First, if no examples interchange under our chosen map BERT_t^r , then our graph for that map, \mathcal{G}_t^r , will have no edges at all and BERT mimics the causal dynamics of INFER on no subset of MoNLI. Second, if all examples interchange under our chosen map BERT_t^r , then our graph for that map, \mathcal{G}_t^r , will be one enormous clique and BERT mimics the causal dynamics of INFER on all of MoNLI.

Even with our assumptions restricting us to the 36 maps defined by $\text{BERT}_{w_p}^r$, $\text{BERT}_{w_h}^r$ and $\text{BERT}_{[\text{CLS}]}^r$, the computational load of performing

(cemetery,location)		(dogs,huskies)		
(house,location)	(den,location)	(dog,husky)	(dog,chiuahua)	
(ghetto,location)	(backyard,location)	(dog,retriever)	(dog,maltese)	(hood,thing)
(jungle,location)	(meadow,location)	(dog,terrier)	(dog,pomeranian)	(nut,thing)
(laboratory,location)	(residence,location)			(capsule,thing)
(slum,location)	(playground,location)	(beetle,insect)		(pouch,thing)
(lab,location)	(studio,location)	(grasshopper,insect)	(bee,insect)	(root,thing)
	(station,location)	(wasp,insect)	(fly,insect)	(nugget,thing)
	(campsite,location)	(butterfly,insect)	(cricket,insect)	(tube,thing)
(town,location)	(farm,location)	(mosquito,insect)		
	(lawn,location)			(box,object)
(saxophone,instrument)	(flute,instrument)			(object,sweater)
		(person,vegetarian)	(person,lunatic)	(hat,object)
(bass,instrument)	(piano,instrument)			(object,jacket)
		(person,republi	(person,trooper)	(toy,object)
(violin,instrument)	(tuba,instrument)	(person,business)		(cane,object)
		(person,navigator)		
(harmonica,instrument)	(person,steward)	(person,goalkeeper)		
	(person,consultant)			
	(person,housekeeper)			
(liquid,whiskey)	(person,sophomore)			
(liquid,margarita)	(person,housekeeper)			
(liquid,alcohol)	(person,physicist)			
	(person,cop)			
	(person,cambodian)			
	(person,detective)			
(woman,granny)	(person,genius)	(person,sergeant)		
	(person,californian)			
(woman,widow)	(person,doctor)	(person,runner)		

Figure 5: A visualization of the largest subset of MoNLI on which we verified BERT mimics the causal dynamics of INFER. This subset contains 98 examples and we display the substituted words in each. The first word in the pair comes from the premise and we cluster word pairs based on hyponyms.

almost 300 million interchange experiments to construct 36 graphs is too high. Under the constraint of resources, we randomly conducted interchange experiments to partially construct each of the 36 graphs and selected the map whose graph exhibited the most clustering, which was BERT_{wh}³.

The problem of finding the largest clique in a graph is NP-complete, so only heuristics are available, but heuristics are fine for the purpose of finding a clique that is large enough. Some edges correspond to interchanges that are causal (the output changes), and some correspond to interchanges that are not causal. To ensure we identify cliques with at least one edge corresponding to a causal interchange, we use the following greedy algorithm: begin with the full graph, and then remove the node with the least number of causal edges until the node with the least number of causal edges has less than α , then remove the node with the least number of edges until only a clique remains. We tested α values between 1 and 10 and chose the best results. We seek only cliques that contain a causal edge, because then the subset of MoNLI corresponding to the clique will have both lexical entailment relations represented.

We ran interchange interventions at the location BERT_{wh}³ to construct a graph which we partitioned

into cliques using our simple, greedy algorithm. We discovered several large disjoint cliques corresponding to subsets of MoNLI. These cliques had size 98, 63, 47, and 37. We show a visualization of the largest subset on MoNLI containing 98 examples in Figure 5.

To put these results in context, consider a graph with the same number of nodes as the original and edges that were assigned randomly with a 50% probability. This baseline tells us the level of modularity that would be expected if interchanging a representation randomized the output of the model for its binary classification task. The expected number of cliques of size k for this graph (2,678 nodes; edge probability of 0.5) is $\binom{n}{k} \times 2^{\binom{k}{2}}$. Thus, for $k > 20$, the expected number of cliques with k nodes is less than 10^{-8} .