Probing Natural Language Inference Models through Semantic Fragments

Kyle Richardson† and Hai Hu‡ and Lawrence S. Moss† and Ashish Sabharwal†

†Allen Institute for AI, Seattle, WA, USA
‡Indiana University, Bloomington, IN, USA
†{kyler,ashishs}@allenai.org, ‡{huai, lmoss}@indiana.edu

Abstract

Do state-of-the-art models for language understanding already have, or can they easily learn, abilities such as boolean coordination, quantification, conditionals, comparatives, and monotonicity reasoning (i.e., reasoning about word substitutions in sentential contexts)? While such phenomena are involved in natural language inference (NLI) and go beyond basic linguistic understanding, it is unclear the extent to which they are captured in existing NLI benchmarks and effectively learned by models. To investigate this, we propose the use of semantic fragments—systematically generated datasets that each target a different semantic phenomenon—for probing, and efficiently improving, such capabilities of linguistic models. This approach to creating challenge datasets allows direct control over the semantic diversity and complexity of the targeted linguistic phenomena, and results in a more precise characterization of a model’s linguistic behavior. Our experiments, using a library of 8 such semantic fragments, reveal two remarkable findings: (a) State-of-the-art models, including BERT, that are pre-trained on existing NLI benchmark datasets perform poorly on these new fragments, even though the phenomena probed here are central to the NLI task; (b) On the other hand, with only a few minutes of additional fine-tuning—with a carefully selected learning rate and a novel variation of “inoculation”—a BERT-based model can master all of these logic and monotonicity fragments while retaining its performance on established NLI benchmarks.

Introduction

Natural language inference (NLI) is the task of detecting inferential relationships between natural language descriptions. For example, given the pair of sentences All dogs chased some cat and All small dogs chased a cat shown in Figure 1, the goal for an NLI model is to determine that the second sentence, known as the hypothesis sentence, follows from the meaning of the first sentence (the premise sentence). Such a task is known to involve a wide range of reasoning and knowledge phenomena, including knowledge that goes beyond basic linguistic understanding (e.g., elementary logic). As one example of such knowledge, the inference in Figure 1 involves monotonicity reasoning (i.e., reasoning about word substitutions in context); here the position of dogs in the premise occurs in a downward monotone context (marked as ↓), meaning that it can be special-

(Linguistically interesting issue, e.g., monotonicity)

**premise:** All dogs chased some cat
**hypothesis:** (entails) All small dogs chased a cat

Construct

Formal Specification: Fragment with Idealized NLI examples

Generate

challenge dataset (NLI pairs)

Empirical Questions

1. Is this fragment learnable using existing NLI architectures?
2. How do pre-trained NLI models perform on this fragment?
3. Can models be fine-tuned/re-trained to master this fragment?

Figure 1: An illustration of our proposed method for studying NLI model behavior through semantic fragments.
definitely answer questions such as: are models able to effectively learn and extrapolate complex knowledge and reasoning abilities when trained on benchmark tasks?

To date, studies using challenge datasets have largely been limited by the simple types of inferences that they included (e.g., lexical and negation inferences). They fail to cover more complex reasoning phenomena related to logic, and primarily use adversarially generated corpus data, which sometimes makes it difficult to identify exactly the particular semantic phenomena being tested for. There is also a focus on datasets that are easily able to be constructed and/or verified using crowd-sourcing techniques. Adequately evaluating a model’s competence on a given reasoning phenomena, however, often requires datasets that are hard even for humans, but that are nonetheless based on sound formal principles (e.g., reasoning about monotonicity where, in contrast to the simple example in Figure 1, several nested downward monotone contexts are involved to test the model’s capacity for compositionality, cf. Lake and Baroni (2017)).

In contrast to existing work on challenge datasets, we propose using semantic fragments—synthetically generated challenge datasets, of the sort used in linguistics, to study NLI model behavior. Semantic fragments provide the ability to systematically control the semantic complexity of each new challenge dataset by bringing to bear the expert knowledge encapsulated in formal theories of reasoning, making it possible to more precisely identify model performance and competence on a given linguistic phenomenon. While our idea of using fragments is broadly applicable to any linguistic or reasoning phenomena, we look at eight types of fragments that cover several fundamental aspects of reasoning in NLI, namely, monotonicity reasoning using two newly constructed challenge datasets as well as six other fragments that probe into rudimentary logic using new versions of the data from Salvatore, Finger, and Hirata Jr (2019).

As illustrated in Figure 1, our proposed method works in the following way: starting with a particular linguistic fragment of interest, we create a formal specification (or a formal rule system with certain guarantees of correctness) of that fragment, with which we then automatically generate a new idealized challenge dataset, and ask the following three empirical questions. 1) Is this particular fragment learnable from scratch using existing NLI architectures (if so, are the resulting models useful)? 2) How well do large state-of-the-art pre-trained NLI models (i.e., models trained on all known NLI data such as SNLI/MNLI) do on this task? 3) Can existing models be quickly re-trained or re-purposed to be robust on these fragments (if so, does mastering a given linguistic fragment affect performance on the original task)?

We emphasize the quickly part in the last question; given the multitude of possible fragments and linguistic phenomena that can be formulated and that we expect a wide-coverage NLI model to cover, we believe that models should be able to efficiently learn and adapt to new phenomena as they are encountered without having to learn entirely from scratch. In this paper we look specifically at the question: are there particular linguistic fragments (relative to other fragments) that are hard for these pre-trained models to adapt to or that confuse the model on its original task?

On these eight fragments, we find that while existing NLI architectures can effectively learn these particular linguistic phenomena, pre-trained NLI models do not perform well. This, as in other studies (Glockner, Shwartz, and Goldberg 2018), reveals weaknesses in the ability of these models to generalize. While most studies into linguistic probing end the story there, we take the additional step to see if attempts to continue the learning and re-fine-tune these models on fragments (using a novel and cheap inoculation (Liu, Schwartz, and Smith 2019) strategy) can improve performance. Interestingly, we show that this yields mixed results depending on the particular linguistic phenomena and model being considered. For some fragments (e.g., comparatives), re-training some models comes at the cost of degrading performance on the original tasks, whereas for other phenomena (e.g., monotonicity) the learning is more stable, even across different models. These findings, and our technique of obtaining them, make it possible to identify the degree to which a given linguistic phenomenon stresses a benchmark NLI model, and suggest a new methodology for quickly making models more robust.

Related Work

The use of semantic fragments has a long tradition in logical semantics, starting with the seminal work of Montague (1973), as well as earlier work on NLI (Cooper et al. 1996). We follow Pratt-Hartmann (2004) in defining a semantic fragment more precisely as a subset of a language equipped with semantics which translate sentences in a formal system such as first-order logic. In contrast to work on empirical NLI, such linguistic work often emphasizes the complex cases of each phenomena in order measure competence (see Chomsky (1965) for a discussion about competence vs. performance). For our fragments that test basic logic, the target formal system includes basic boolean algebra, quantification, set comparisons and counting (see Figure 2), and builds on the datasets from Salvatore, Finger, and Hirata Jr (2019). For our second set of fragments that focus on monotonicity reasoning, the target formal system is based on the monotonicity calculus of van Benthem (1986) (see review by Icard and Moss (2014)). To construct these datasets, we build on recent work on automatic polarity projection (Hu and Moss 2018; Hu, Chen, and Moss 2019; Hu et al. 2019).

Our work follows other attempts to learn neural models from fragments and small subsets of language, which includes work on syntactic probing (McCoy, Pavlick, and Linzen 2019; Goldberg 2019), probing basic reasoning (Winston et al. 2015; Geiger et al. 2018; 2019) and probing other tasks (Lake and Baroni 2017; Chrupala and Alishahi 2019; Warstadt et al. 2019). Geiger et al. (2018) is the closest work to ours. However, they intentionally focus on artificial fragments that deviate from ordinary language, whereas our fragments (despite being automatically constructed and sometimes a bit pedantic) aim to test naturalistic subsets of English. In a similar spirit, there have been other attempts to collect datasets that target different types of inference phenomena (White et al. 2017; Poliak et al. 2018a), which have been limited in linguistic complexity. Other attempts to
study complex phenomena such as monotonicity reasoning in NLI models has been limited to training data augmentation (Yanaka et al. 2019b), whereas we create several new challenge test sets to directly evaluate NLI performance on each phenomenon (see Yanaka et al. (2019a) for closely related work that appeared concurrently with our work).

Unlike existing work on building NLI challenge datasets (Glockner, Shwartz, and Goldberg 2018; Naik et al. 2018), we focus on the trade-off between mastering a particular linguistic fragment or phenomena independent of other tasks and data (i.e., Question 1 from Figure 1), while also maintaining performance on other NLI benchmark tasks (i.e., related to Question 3 in Figure 1). To study this, we introduce a novel variation of the inoculation through fine-tuning methodology of Liu, Schwartz, and Smith (2019), which emphasizes maximizing the model’s aggregate score over multiple tasks (as opposed to only on challenge tasks). Since our new challenge datasets focus narrowly on particular linguistic phenomena, we take this in the direction of seeing more precisely the extent to which a particular linguistic fragment stresses an existing NLI model. In addition to the task-specific NLI models looked at in Liu, Schwartz, and Smith (2019), we inoculate with the state-of-the-art pre-trained BERT model, using the fine-tuning approach of Devlin et al. (2019), which itself is based on the transformer architecture of Vaswani et al. (2017).

### Some Semantic Fragments

As shown in Figure 1, given a particular semantic fragment or linguistic phenomenon that we want to study, our starting point is a formal specification of that fragment (e.g., in the form of a set of templates/formal grammar that encapsulate expert knowledge of that phenomenon), which we can sample in order to obtain a new challenge set. In this section, we describe the construction of the particular fragments we investigate in this paper, which are illustrated in Figure 2. While these particular fragments seem to capture many of the core phenomena involved in NLI, we emphasize that any arbitrary linguistic fragment of interest could be constructed and subjected to the sets of experiments we describe in the next section.

#### The Logic Fragments

The first set of fragments probe into problems involving rudimentary logical reasoning. Using a fixed vocabulary of people and place names, individual fragments cover boolean coordination (boolean reasoning about conjunction and), simple negation, quantification and quantifier scope (quantifier), comparative relations, set counting, and conditional phenomena all related to a small set of traveling and height relations.

These fragments (with the exception of the conditional fragment, which was built specially for this study) were first built using the set of verb-argument templates first described in Salvatore, Finger, and Hirata Jr (2019). Since their original rules were meant for 2-way NLI classification (i.e., ENTAILMENT and CONTRADICTION), we repurposed their rule sets to handle 3-way classification, and added other inference rules, which resulted in some of the simplified templates shown in Figure 3. For each fragment, we uniformly generated 3,000 training examples and reserved 1,000 examples for testing. As in Salvatore, Finger, and Hirata Jr (2019), the people and place names for testing are drawn from an entirely disjoint set from training. We also reserve 1,000 for development. While we were capable of generating more data, we follow Weston et al. (2015) in limiting the size of our training sets to 3,000 since our goal is to learn from as little data as possible, and found 3,000 training examples to be sufficient for most fragments and models.

As detailed in Figure 2, these new fragments vary in complexity, with the negation fragment (which is limited to verbal negation) being the least complex in terms of linguistic phenomena. We also note that all other fragments include basic negation and boolean operators, which we found to help preserve the naturalness of the examples in each fragment. As shown in last column of Figure 2, some of our frag-
The Monotonicity Fragments  

The second set of fragments cover monotonicity reasoning, as first discussed in the introduction. This fragment can be described using a regular grammar with polarity facts according to the monotonicity calculus, such as the following: *every* is downward monotone/entailing in its first argument but *upward* monotone/entailing in the second, denoted by the ↑ and ↓ arrows in the example sentence *every*↑↑↑small↓↓↓dog↓↓↓ran↓↓↓. We have manually encoded monotonicity information for 14 sets of quantifiers (*every*, *some*, *no*, *most*, at least 3, at most 4, etc.) and negators (*not*, *without*) and generated sentences using a simple regular grammar and a small lexicon of about 100 words. We then use the system described by Hu and Moss (2018)¹ to automatically assign arrows to every token (see Figure 4, note that = means that the inference is *neither* monotonically up or down in general). Because we manually encoded the monotonicity information of each token in the lexicon and built sentences via a controlled set of grammar rules, the resulting arrows assigned by Hu and Moss (2018) can be proved to be correct.

Once we have the sentences with arrows, we use the algorithm of Hu, Chen, and Moss (2019) to generate pairs of sentences with ENTAIL, NEUTRAL or CONTRADICTORY relations, as exemplified in Figure 4. Specifically, we first define a knowledge base that stores the relations of the lexical items in our lexicon, e.g., *poodle* ≤ *dog* ≤ *mammal* ≤ *animal*; also, *waltz* ≤ *dance* ≤ *move*; and *every* ≤ *most* ≤ *some* = *a*. For nouns, ≤ can be understood as the subset-superset relation. For higher-type objects like the determiners above, see Icard and Moss (2013) for discussion. Then to generate entailments, we perform substitution (shown in Figure 4 in blue). That is, we substitute upward entailment tokens or constituents with something “greater than or equal to” (≥) them, or downward entailment ones with something “less than or equal to” them. To generate neutrals, substitution goes the reverse way. For example, all↑↑↓dogs↓↓↓danced↓↓↓ENTAIL all poodles danced, while all↑↑↓dogs↓↓↓danced↓↓↓NEUTRAL all mammals danced. This is due to the facts which we have seen: *poodle* ≤ *dog* ≤ *mammal*. Simple rules such as “replace some/many/every in subjects by no” or “negate the main verb” are applied to generate contradictions.

Using this basic machinery, we generated two separate challenge datasets, one with limited complexity (e.g., each example is limited to 1 relative clause and uses an inventory of 5 quantifiers), which we refer to throughout as monotonicity (simple), and one with more overall quantifiers and substitutions, or monotonicity (hard) (up to 3 relative clauses and a larger inventory of 14 unique quantifiers). Both are defined over the same set of lexical items (see Figure 2).

Experimental Setup and Methodology  

To address the questions in Figure 1, we experiment with two task-specific NLI models from the literature, the ESIM model of Chen et al. (2017) and the decomposable-attention (Decomp-Attn) model of Parikh et al. (2016) as implemented in the AllenNLP toolkit (Gardner et al. 2018), and the pre-trained BERT architecture of Devlin et al. (2019).²

1https://github.com/huhailinguist/ccg2mono

2We use the BERT-base uncased model in all experiments, as implemented in HuggingFace: https://github.com/huggingface/
When evaluating whether fragments can be learned from scratch (Question 1), we simply train models on these fragments directly using standard training protocols. To evaluate pre-trained NLI models on individual fragments (Question 2), we train BERT models on combinations of the SNLI and MNLI datasets from GLUE (Wang et al., 2018), and use pre-trained ESIM and Decomp-Attn models trained on MNLI following Liu, Schwartz, and Smith (2019).

To evaluate whether a pre-trained NLI model can be re-trained to improve on a fragment (Question 3), we employ the recent inoculation by fine-tuning method (Liu, Schwartz, and Smith 2019). The idea is to re-fine-tune (i.e., continue training) the models above using k pieces of fragment training data, where k ranges from 50 to 3,000 (i.e., a very small subset of the fragment dataset to the full training set; see horizontal axes in Figures 5, 6, and 7). The intuition is that by doing this, we see the extent to which this additional data makes the model more robust to handle each fragment, or stresses it, resulting in performance loss on its original benchmark. In contrast to re-training models from scratch with the original data augmented with our fragment data, fine-tuning on only the new data is substantially faster, requiring in many cases only a few minutes. This is consistent with our requirement discussed previously that training existing models to be robust on new fragments should be quick, given the multitude of fragments that we expect to encounter over time. For example, in coming up with new linguistic fragments, we might find newer fragments that are not represented in the model; it would be prohibitive to re-train the model each time entirely from scratch with its original data (e.g., the 900k+ examples in SNLI+MNLI) augmented with the new fragment.

Our approach to inoculation, which we call lossless inoculation, differs from Liu, Schwartz, and Smith (2019) in explicitly optimizing the aggregate score of each model on both its original and new task. More formally, let k denote the number of examples of fragment data used for fine-tuning. Ideally, we would like to be able to fine-tune each pre-trained NLI model architecture a (e.g., BERT) to learn a new fragment perfectly with a minimal k, while—importantly—not losing performance on the original task that the model was trained for (e.g., SNLI or MNLI). Given that fine-tuning is sensitive to hyper-parameters, we use the following methodology: For each k we fine-tune J variations of a model architecture, denoted $M_j^{a,k}$ for $j \in \{1, \ldots, J\}$, each characterized by a different set of hyper-parameters. We then identify a model $M_j^{a,k}$ with the best aggregated performance based on its score $S_{frag}(M_j^{a,k})$ on the fragment dataset and $S_{orig}(M_j^{a,k})$ on the original dataset. For simplicity, we use the average of these two scores as the aggregated score. Thus, we have:

$$M_j^{a,k} = \arg\max_{M \in \{M_1^{a,k}, \ldots, M_J^{a,k}\}} \text{AVG}(S_{frag}(M), S_{orig}(M))$$

By keeping the hyper-parameter space consistent among all fragments, the point is to observe how certain fragments behave relative to one another.

### Additional Baselines
To ensure that the challenge datasets that are generated from our fragments are not trivially solvable and subject to annotation artifacts, we implemented variants of the Hypothesis-Only baselines from Polli et al. (2018b), as shown at the top of Table 1. This involves training a single-layered biLSTM encoder for the hypothesis side of the input, which generates a representation for the input using max-pooling over the hidden states, as originally done in Conneau et al. (2017). We used the same model to train a Premise-Only model that instead uses the premise text, as well as an encoder that looks at both the premise and hypothesis (Premise+Hyp.) separated by an artificial token (for more baselines, see Salvatore, Finger, and Hirata Jr (2019)).

### Results and Findings
We discuss the different questions posed in Figure 1.

### Answering Questions 1 and 2.
Table 1 shows the performance of baseline models and pre-trained NLI models on our different fragments. In all cases, the baseline models did poorly on our datasets, showing the inherent difficulty of our challenge sets. In the second case, we see clearly that state-of-the-art models do not perform well on our fragments, poorly on our datasets, showing the inherent difficulty of our challenge sets. In the second case, we see clearly that state-of-the-art models do not perform well on our fragments.

---

1We found all models to be sensitive to learning rate, and performed comprehensive hyper-parameters searches to consider different learning rates, # iterations and (for BERT) random seeds.

2Other ways of aggregating the two scores can be substituted. E.g., one could maximize $S_{frag}(M_j^{a,k})$ while requiring that $S_{orig}(M_j^{a,k})$ is not much worse relative to when the model’s hyper-parameters are optimized directly for the original dataset.
Table 1: Baseline models and model performance (accuracy %) on NLI benchmarks and challenge test sets (before and after re-training), including the Breaking NLI challenge set from Glockner, Shwartz, and Goldberg (2018). The arrows ↓ in the last section show the average drop in accuracy on MNLI benchmark after re-training with the fragments.

Figure 5: Dev. results on training NLI models from scratch on the different fragments and architectures.

consistent with findings on other challenge datasets. One result to note is the high accuracy of BERT-based pre-trained models on the Breaking NLI challenge set of Glockner, Shwartz, and Goldberg (2018), which previously proved to be a difficult benchmark for NLI models. This result, we believe, highlights the need for more challenging NLI benchmarks, such as our new datasets.

Figure 5 shows the results of training NLI models from scratch (i.e., without NLI pre-training on other benchmarks) on the different fragments. In nearly all cases, it is possible to train a model to master a fragment (with counting being the hardest fragment to learn). In other studies on learning fragments (Geiger et al. 2018; Salvatore, Finger, and Hirata Jr 2019), this is the main result reported, however, we also show that the resulting models perform below random chance on benchmark tasks, meaning that these models are not by themselves very useful for general NLI. This even holds for results on the GLUE diagnostic test (Wang et al. 2018), which was hand-created and designed to model many of the logical phenomena captured in our fragments.

We note that in the monotonicity examples, we included results on a development set (in dashed green) that was built by systematically paraphrasing all the nouns and verbs in the fragment to be disjoint from training. Even in this case, when lexical variation is introduced, the BERT model is robust (see Rozen et al. (2019) for a more systematic study of this type of generalization using BERT for NLI in different settings).

Answering Question 3. Figures 6 and 7 show the results of the re-training study. They compare the performance of a retrained model on the challenge tasks (dashed lines) as well as on its original benchmark tasks (solid lines). We discuss here results from the two illustrative fragments depicted in Figure 6. All 4 models can master Monotonicity Reasoning while retaining accuracy on their original benchmarks. However, non-BERT models lose substantial accuracy on their original benchmark when trying to learn comparatives, suggesting that comparatives are generally harder for models to learn. In Figure 7, we show the results for all other fragments, which show varied, though largely stable, trends depending on the particular linguistic phenomena.

At the bottom of Table 1, we show the resulting accuracies on the challenge sets and MNLI benchmark for each model after re-training (using the optimal model $M^{o,k}_a$, as described previously). In the case of $BERT^*_{SNLI+MNLI+frag}$, we see that despite performing poorly on these new chal-

3For MNLI, we report results on the mismatched dev. set.
Figure 6: Inoculation results for two illustrative semantic fragments, Monotonicity Reasoning (left) and Comparatives (right), for 4 NLI models shown in different colors. Horizontal axis: number of fine-tuning challenge set examples used. Each point represents the model $M^*_k$ trained using hyperparameters that maximize the accuracy averaged across the model’s original benchmark dataset (solid line) and challenge dataset (dashed line).

Figure 7: Inoculation results for 6 semantic fragments not included in Figure 6, using the same setup.

lenge dataset before re-training, it can learn to master these fragments with minimal losses to performance on its original task (i.e., it only loses on average about 1.3% accuracy of the original MNLI dev set). In other words, it is possible teach BERT (given its inherent capacity) a new fragment quickly through re-training without affecting its original performance, assuming however that time is spent on carefully finding the optimal model.\footnote{We note that models without optimal aggregate performance are often prone to catastrophic forgetting.} For the other models, there is more of a trade-off; Decomp-Attn on average never quite masters the logic fragments (but does master the Monotonicity Fragments), and incurs an average 6.7% loss on MNLI after re-training. In the case of comparatives, the inability of the model to master this fragment likely reveals a certain architectural limitation of the model given that it is not sensitive to word-order. Given such losses, perhaps in such cases a more sophisticated re-training scheme is needed in order to optimally learn particular fragments.

### Discussion and Conclusion

We explored the use of semantic fragments—systematically controlled subsets of language—to probe into NLI models and benchmarks. Our investigation considered 8 particular fragments and new challenge datasets that center around basic logic and monotonicity reasoning. In answering the questions first introduced in Figure 1, we found that while existing NLI architectures are able to learn these fragments from scratch, the resulting models are of limited interest. Further, pre-trained models perform poorly on these new datasets (even relative to other available challenge benchmarks), revealing the weaknesses of these models. Interestingly, how-
ever, we show that many models can be quickly re-tuned (e.g., often in a matter of minutes) to master these different fragments using a novel variant of the inoculation through fine-tuning strategy (Liu, Schwartz, and Smith 2019) that we introduce called lossless inoculation.

Our results suggest the following methodology for improving models: Given a particular linguistic hole in an NLI model, one can plug this hole by simply generating synthetic data and using it to re-train a model. This methodology comes with some caveats, however: Depending on the model and particular linguistic phenomena, there may be some trade-offs with the model’s original performance, which should first be looked at empirically and compared against other linguistic phenomena. Our work is one small step in trying to gather an inventory of NLI phenomena and look rigorously at model performance, which follows earlier work on NLI (see Zaenen, Karttunen, and Crouch (2005)).

Can we find more difficult fragments? Despite differences across various fragments, we largely found NLI models to be robust when tackling new linguistic phenomena and easy to quickly re-purpose (especially with BERT). This generally positive result begs the question: Are there more challenging fragments and linguistic phenomena that we should be studying?

The ubiquity of logical and monotonicity reasoning provides a justification for our particular fragments, and we take it as a positive sign that models are able to solve these tasks. As we emphasize throughout, however, our general approach is amenable to any linguistic phenomena, and future work may focus on developing more complicated fragments that capture a wider range of linguistic phenomena and inference. This could include, for example, efforts to extend to fragments in a way that moves beyond elementary logic to systematically target the types of commonsense reasoning known to be common in existing NLI tasks (LoBue and Yates 2011). We believe that semantic fragments are a promising way to introspect model performance generally, and can also be used to forge interdisciplinary collaboration between neural NLP research and traditional linguistics.

Benchmark NLI annotations and judgements are often imperfect and error-prone (cf. Kalouli, de Paiva, and Real (2017), Pavlick and Kwiatkowski (2019)), partly due to the loose way in which the task is traditionally defined (Dagan, Glickman, and Magnini 2005). For models trained on benchmarks such as SNLI, understanding model performance not only requires probing how each target model works, but also probing the particular flavor of NLI that is captured in each benchmark. We believe that our variant of inoculation and overall framework can also be used to more systematically look at these issues, as well as help identify annotation errors and artifacts.

What are Models Actually Learning? One open question concerns the extent to which models trained on narrow fragments can generalize beyond them. Newer analysis methods that attempt to correlate neural activation patterns and target symbolic patterns (Chrupała and Alishahi 2019) might help determine the extent to which models are truly generalizing, and provide insights into alternative ways of training more robust and generalizable models.

A key feature of our lossless inoculation strategy, which differs from the original proposal of Liu, Schwartz, and Smith (2019), is that each time we teach the model something new, we explicitly take into account how much loss this same model has on its original task, and balance the two scores accordingly. The fact that models such as BERT can effectively learn new tasks with minimal loss on their original tasks gives some indication that, even if the models are not generalizing too far beyond the provided challenge tasks, one way to increase generalization is by continuously feeding models new challenge tasks. This type of continuous or never-ending learning scenario is one promising area for future work that one may pursue by looking at more robust methods for model inoculation and fine-tuning.

Acknowledgements

We thank the anonymous reviewers for their helpful feedback, as well as our colleagues, especially Peter Clark, Vered Shwartz, and Reut Tsarfaty. Part of this work is supported by grant #586136 from the Simons Foundation. Hai Hu is supported by the China Scholarship Council.

References

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A Large Annotated Corpus for Learning Natural Language Inference. In EMNLP.

Chen, Q.; Zhu, X.; Ling, Z.-H.; Wei, S.; Jiang, H.; and Inkpen, D. 2017. Enhanced LSTM for Natural Language Inference. In ACL.

Chomsky, N. 1965. Aspects of the Theory of Syntax. MIT press.

Chrupała, G., and Alishahi, A. 2019. Correlating Neural and Symbolic Representations of Language. In ACL.

Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; and Bordes, A. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In EMNLP.

Cooper, R.; Crouch, D.; Van Eijck, J.; Fox, C.; Van Genabith, J.; Jaspars, J.; Kamp, H.; Milward, D.; Pinkal, M.; Poesio, M.; et al. 1996. Using the Framework. Technical report, LRE 62-051 D-16, The FraCaS Consortium.

Dagan, I.; Glickman, O.; and Magnini, B. 2005. The PASCAL Recognising Textual Entailment Challenge. In Machine Learning Challenges Workshop.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL.

Gardner, M.; Grus, J.; Neumann, M.; Tafjord, O.; Dasigi, P.; Liu, N.; Peters, M.; Schmitz, M.; and Zettlemoyer, L. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In Workshop for NLP Open Source Software (NLP-OSS).
Geiger, A.; Cases, I.; Karttunen, L.; and Potts, C. 2018. Stress-Testing Neural Models of Natural Language Inference with Multiply-Quantified Sentences. arXiv:1810.13033.

Geiger, A.; Cases, I.; Karttunen, L.; and Potts, C. 2019. Positing Fair Generalization Tasks for Natural Language Inference. In EMNLP.

Glockner, M.; Shwartz, V.; and Goldberg, Y. 2018. Breaking NLI Systems with Sentences that Require Simple Lexical Inferences. In ACL.

Goldberg, Y. 2019. Assessing BERT’s Syntactic Abilities. arXiv:1901.05287.

Gururangan, S.; Swayamdipta, S.; Levy, O.; Schwartz, R.; Bowman, S.; and Smith, N. A. 2018. Annotation Artifacts in Natural Language Inference Data. In NAACL.

Hu, H., and Moss, L. S. 2018. Polarity Computations in Flexible Categorial Grammar. In *SEM.

Hu, H.; Chen, Q.; Richardson, K.; Mukherjee, A.; Moss, L. S.; and Kuebler, S. 2019. Monalog: a Lightweight System for Natural Language Inference Based on Monotonicity. arXiv:1910.08772.

Hu, H.; Chen, Q.; and Moss, L. S. 2019. Natural Language Inference with Monotonicity. In IWCS.

Icard, T. F., and Moss, L. S. 2013. A Complete Calculus of Inference with Monotonicity. In IWCS.

Icard, T. F., and Moss, L. S. 2014. Recent Progress on Monotonicity. Linguistic Issues in Language Technology 9(7):167–194.

Kalouli, A.-L.; de Paiva, V.; and Real, L. 2017. Correcting Contradictions. In the Computing Natural Language Inference Workshop.

Lake, B. M., and Baroni, M. 2017. Generalization Without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks. In ICML.

Liu, N. F.; Schwartz, R.; and Smith, N. A. 2019. Inoculation by Fine-Tuning: A Method for Analyzing Challenge Datasets. In NAACL.

LoBue, P., and Yates, A. 2011. Types of Common-sense Knowledge Needed for Recognizing Textual Entailment. In ACL.

McCoy, R. T.; Pavlick, E.; and Linzen, T. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In ACL.

Montague, R. 1973. The proper treatment of quantification in ordinary English. In Approaches to natural language. Springer.

Naik, A.; Ravichander, A.; Sadeh, N.; Rose, C.; and Neubig, G. 2018. Stress Test Evaluation for Natural Language Inference. In COLING.

Parikh, A. P.; Täckström, O.; Das, D.; and Uszkoreit, J. 2016. A Decomposable Attention Model for Natural Language Inference. In EMNLP.

Pavlick, E., and Kwiatkowski, T. 2019. Inherent Disagreements in Human Textual Inferences. TACL 7:677–694.

Poliak, A.; Haldar, A.; Rudinger, R.; Hu, J. E.; Pavlick, E.; White, A. S.; and Van Durme, B. 2018a. Collecting Diverse Natural Language Inference Problems for Sentence Representation Evaluation. In EMNLP.

Poliak, A.; Naradowsky, J.; Haldar, A.; Rudinger, R.; and Van Durme, B. 2018b. Hypothesis Only Baselines in Natural Language Inference. In *SEM.

Pratt-Hartmann, I. 2004. Fragments of Language. Journal of Logic, Language and Information 13(2):207–223.

Rozen, O.; Shwartz, V.; Aharoni, R.; and Dagan, I. 2019. Analyzing Generalization in Natural Language Inference via Controlled Variance in Adversarial Datasets. In CoNLL.

Salvatore, F.; Finger, M.; and Hirata Jr, R. 2019. Using Syntactical and Logical Forms to Evaluate Textual Inference Competence. arXiv:1905.05704.

Steedman, M. 2000. The Syntactic Process. MIT Press.

van Benthem, J. 1986. Essays in Logical Semantics, volume 29 of Studies in Linguistics and Philosophy. Dordrecht: D. Reidel Publishing Co.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In NIPS, 5998–6008.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2018. Glue: A Multi-task Benchmark and Analysis Platform for Natural Language Understanding. In EMNLP Workshop BlackboxNLP.

Warstadt, A.; Cao, Y.; Grosu, I.; Peng, W.; Blixa, H.; Nie, Y.; Alsop, A.; Bordia, S.; Liu, H.; Parrish, A.; et al. 2019. Investigating BERT’S Knowledge of Language: Five Analysis Methods with NPIs. In EMNLP.

Weston, J.; Bordes, A.; Chopra, S.; Rush, A. M.; van Merriënboer, B.; Joulin, A.; and Mikolov, T. 2015. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. arXiv:1502.05698.

White, A. S.; Rastogi, P.; Duh, K.; and Van Durme, B. 2017. Inference is Everything: Recasting Semantic Resources into a Unified Evaluation Framework. In IJCNLP.

Williams, A.; Nangia, N.; and Bowman, S. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In NAACL.

Yanaka, H.; Mineshima, K.; Bekki, D.; Inui, K.; Sekine, S.; Abzianidze, L.; and Bos, J. 2019a. Can Neural Networks Understand Monotonicity Reasoning? In ACL Workshop BlackboxNLP.

Yanaka, H.; Mineshima, K.; Bekki, D.; Inui, K.; Sekine, S.; Abzianidze, L.; and Bos, J. 2019b. HELP: A Dataset for Identifying Shortcomings of Neural Models in Monotonicity Reasoning. In *SEM.

Zaenen, A.; Karttunen, L.; and Crouch, R. 2005. Local Textual Inference: Can it be Defined or Circumscribed? In the ACL workshop on empirical modeling of semantic equivalence and entailment.
Appendix

We provide more details of how the monotonicity fragment. To generate this fragment, we first defined a grammar with a hand coded lexicon. Then we use the tool described in Hu and Moss (2018) to obtain polarities (arrows) on each constituent of the sentences. Finally we implement the substitution algorithm on the polarized sentences described in Hu, Chen, and Moss (2019). These steps are detailed below.

Grammar and Lexicon All premises in the monotonicity fragment are generated using the grammar and lexicon detailed in Figure 8 included in this Appendix.

Polarization We use the tool from Hu and Moss (2018) to generate a generated sentence. For example, if the input is every dog slept, their tool will output every↑ dog↓ slept↑, represented as a CCG tree (Steedman 2000).

Substitution As described in the paper and Figure 4 in the paper, we follow Hu, Chen, and Moss (2019) to generate sentence pairs from an input premise. In simple terms, the main search tree (Figure 4) is based on inferences, at each node polarized and contradictions are also generated (see function substitute() below).

Algorithm 1 shows the pseudocode for this procedure. Assuming an input sentence, the model first polarizes this sentence (line 43 using the POLARIZE function) using the algorithm of Hu and Moss (2018) as described above. Three sets of sentences are created (line 2): entailments of sentence (infer), contradictions (contr), and neutral inferences (neutr). Inferred sentences from infer are then selected (starting line 3) and new inferences are generated using the SUBSTITUTE function called on line 8.

The SUBSTITUTE function works in the following way: for each polarized sentence s, it enumerates through all constituents (const) in s's parse tree (equal to (2 | s | -1) constituents, since we assume binary derivation rules). For each constituent, the polarity marking (i.e., the arrows ↑, ↓) is used to determine how span substitution should work. For example, if the polarity of the constituent if up (↑, see line 18), then it will generate a span substitution (using the SWAP function) using rules from a knowledge base K that generalize (or have a greater relation with const). The opposite for constituents with ↓ labels.

### Algorithm 1 Generating Inferences

**Input:** a sentence sentence, knowledge base K, depth d, starting depth 0.

**Output:** Lists of inferences.

1. function SEARCH(ps,d)
2. infer, contr, neutr ← [(0,ps), [], []
3. while len(infer) > 0 do
4.   depth,s ← infer.pop()
5.   depth + = 1
6.   if depth ≤ d then
7.     ▷ Generate new inferences
8.     e,n,c ← SUBSTITUTE(s)
9.     infer.add((depth,e))
10.    neutr.add(n)
11.    contr.add(c)
12.  return infer, contr, neutr
13. function SUBSTITUTE(s)
14.    infer, contr, neutr ← [], [], []
15.    ▷ Iterate sentence constituents
16. for const in s do
17.        ▷ If polarity is up, generalize span
18.        if const.polarity is ↑ then
19.            for repl in K[const].greater do
20.                ns ← SWAP(s, const, repl)
21.                ps ← POLARIZE(ns)
22.                infer.add(ps)
23.            for repl in K[const].less do
24.                ns ← SWAP(s, const, repl)
25.                ps ← POLARIZE(ns)
26.                neutr.add(ps)
27.        ▷ If polarity is down, specialize span
28.        if const.polarity is ↓ then
29.            for repl in K[const].less do
30.                ns ← SWAP(s, const, repl)
31.                ps ← POLARIZE(ns)
32.                infer.add(ps)
33.            for repl in K[const].greater do
34.                ns ← SWAP(s, const, repl)
35.                ps ← POLARIZE(ns)
36.                neutr.add(ps)
37.        ▷ Find negation replacements
38.            for repl in K[const].negate do
39.                ns ← SWAP(s, const, repl)
40.                ps ← POLARIZE(ns)
41.                contr.add(ps)
42.        return infer, contr, neutr
43.  ps ← POLARIZE(sentence)
44. return SEARCH(ps, d)
Grammar:

S → NPanimate VP
NP → NPanimate | NPinanimate
NPanimate → Det (Adjanimate) Nbaranimate
NPinanimate → Det (Adjinanimate) Ninanimate
Nbaranimate → Nanimate | Nanimate SRC | Nanimate ORC | Nanimate PP
SRC → who VP
ORC → who NPanimate Vt
PP → P Adjsmell Nsmell
VP → VPbar | do not VPbar | be (not) Adjpred
VPbar → Vt | Vi NP

(note: SRC = subject relative clause, ORC = object relative clause)

Lexicon:

Det → {every, all, each, a, many, several, few, most, the, some but not all, no, at least n, at most n, exactly n}
Adjanimate → {wooden, hardwood, metal, plastic, iron, steel}
Adjanimate → {old, young, newborn, brown, black, brown or black}
Adjpred → {happy, sad}
Adjsmell → {strong, faint}
Adjsubjective → {good, bad, nice} # only used in building knowledge base
Nanimate → {dog, cat, rabbit, animal, mammal, poodle, beagle, bulldog, bat, horse, stallion, badger, quadruped}
Ninanimate → {table, wagon, chair, door, object, wheel, box, mailbox, wheelbarrow, fence}
Nsmell → {smell, odor, scent}
Vt → {saw, stared-at, inspected, hit, touched, moved-towards, moved-away-from, scratched, sniffed}
Vi → {slept, ran, moved, swam, waltzed, danced}
P → {with, without}

Example Pre-order/Antonym Relations from Knowledge Base:

adjectives
brown ≤ brown or black, black ≤ brown or black
iron ≤ metal, steel ≤ metal, steel ≤ iron, hardwood ≤ wooden

nouns
x ≤ animal for x in {dog, cat, rabbit, mammal, poodle, beagle, bulldog, bat, horse, stallion, badger}
x ≤ mammal for x in {dog, cat, rabbit, mammal, poodle, beagle, bulldog, bat, horse, stallion, badger}
x ≤ dog for x in {poodle, beagle, bulldog}
x ≤ object for x in Nanimate

verbs
x ≤ moved for x in {ran, swam, waltzed, danced}

Determiners
every = all = each ≤ most ≤ some = a, many ≤ several ≤ at least 3 ≤ at least 2 ≤ some = a,
no ≤ at most 1 ≤ at most 2 ≤ ...

Other rules
Adj N ≤ N, N + (SRC | ORC) ≤ N, ...

Antonyms
moved-towards ⊥ moved-away-from, x ⊥ slept for x in {ran, swam, waltzed, danced, moved}
at most 4 ⊥ at least 5, exactly 4 ⊥ exactly 5, every ⊥ some but not all, ...

Figure 8: A specification of the grammar and lexicon used to generate the monotonicity fragments.