Supplementary materials for 'Posing Fair Generalization Tasks for Natural Language Inference'

A Some Unfair Idealized Test Scenarios

In Section 4.4, we explained that a fair NLI dataset must expose every joint projectivity signature to each semantic relation. We will now demonstrate that the difficult generalization tasks posed by Bowman (2013) and Veldhoen and Zuidema (2018) are unfair in our sense. Both use the dataset provided in Bowman (2013), which contains examples with premises and hypotheses that contain a single quantifier.

Bowman (2013) propose the generalization tasks SUBCLASS-OUT and PAIR-OUT. For a particular joint projectivity signature and semantic relation input, the SUBCLASS-OUT generalization task holds out all examples that expose that joint projectivity signature to that semantic relation for testing. For a particular joint projectivity signature, the PAIR-OUT generalization task holds out all examples containing that joint projectivity signature for testing. Both of these tasks directly violate our standard of fairness by not exposing all joint projectivity signatures to all semantic relation inputs.

There is a certain class of examples where the premise and hypothesis are equivalent due to De Morgan's laws for quantifiers. Veldhoen and Zuidema (2018) propose a generalization task that holds out this class of examples for testing. However, these are the only examples in which the joint projectivity signature for *all* and *no* and the joint projectivity signature for *not all* and *some* are exposed to the semantic relations $\equiv$ and $^$. This directly violates our standard of fairness by not exposing all joint projectivity signatures to all semantic relation inputs.

B Joint Projectivity

The projectivity signature of a semantic function f is $P_f : \mathcal{B} \to \mathcal{B}$ where, if the relation between A and B is R, the relation between $f(A)$ and $f(B)$ is $P_f(R)$. The natural logic of MacCartney and Manning (2009) only makes use of projectivity signatures, and is unable to derive De Morgan's laws for quantifiers. For example, it would label the relation between *some dog eats* and *every dog does not eat* as independence $\#$ when the true semantic relation between these expressions is contradiction, $^$. This demonstrates the need for a joint projectivity signature that directly captures such relations.

We provide a small extension to the natural logic theory of MacCartney and Manning (2009) by introducing joint projectivity signatures, which allow for new inferences involving quantifiers. The joint projectivity signature of a pair of semantic functions f and g is $P_{f/g} : \mathcal{B} \to \mathcal{B}$ where, if the relation between A and B is R, the relation between $f(A)$ and $g(B)$ is $P_{f/g}(R)$. It follows that the joint projectivity between a semantic function and itself is equivalent to the projectivity of that semantic function.

The joint projectivity signatures between *every* and *some* are provided in Figure 5 along with the joint projectivity signatures between ε and *not*. We mark where the natural logic of MacCartney and Manning (2009) is extended. The remaining joint projectivity signatures between the quantifiers *every*, *some*, *no*, and *not every* can be determined by composing the joint projectivity signatures of *every* and *some* with the joint projectivity signatures between *not* and ϵ, where we parse *no* as *not some*.

C Fair Data Generation Algorithm

We now present Algorithm 3, which generates fair training data of varying difficulties by restricting the way an intermediate output is realized during training based on the outputs realized by sibling nodes. The *ratio* parameter determines the difficulty of the generalization task; the higher the ratio, the more permissive the training set and the easier the task. This algorithm uses a handful of helper functions. The function $\texttt{children}(a)$ returns the left-to-right ordered children of node a and the function $\texttt{cartesian\_product}(L)$ returns the Cartesian product of a tuple of sets L. The function $\texttt{sibling\_space}(a, c_k)$ returns the set $\mathit{Dom}(c_1) \times \cdots \times \mathit{Dom}(c_{k-1}) \times \mathit{Dom}(c_{k+1}) \times \mathit{Dom}(c_m)$ where $c_1, \ldots, c_m$ are the children of a. The function $\texttt{random\_even\_split}(S, D, \mathit{ratio})$ partitions a set S into P_1 and P_2 where $\frac{|P_1|}{|S|} = \mathit{ratio}$ and returns a function $F : D \to \wp(S)$ ($\wp$ is the powerset function) that satisfies the follow properties: the elements in the range of F are non-empty, P_1 is a subset of every element in the range of F, the union of every element in the range of F is S, and the elements of P_2 are randomly and evenly distributed among the

	Projectivity for first argument							Projectivity for second argument						
	≡	⊏	⊐	^	|	⌣	#	≡	⊏	⊐	^	|	⌣	#
some/every	⊐	⊐	⊐	#	#	#	#	⊐	#	⊐	^	|	⌣	#
every/some	⊏	⊏	⊏	#	#	#	#	⊏	⊏	#	^	|	⌣	#
every/every	≡	⊐	⊏	|	#	|	#	≡	⊏	⊐	|	|	#	#
some/some	≡	⊏	⊐	⌣	#	⌣	#	≡	⊏	⊐	⌣	#	⌣	#
not/ε	^	⌣	|	≡	⊐	⊏	#	-	-	-	-	-	-	-
ε/not	^	⌣	|	≡	⊏	⊐	#	-	-	-	-	-	-	-
not/not	≡	⊐	⊏	^	⌣	|	#	-	-	-	-	-	-	-
ε/ε	≡	⊏	⊐	^	|	⌣	#	-	-	-	-	-	-	-

Figure 5: The four joint projectivity signatures between quantifiers *every* and *some* and the four joint projectivity signatures between ε and *not*. Boxed relations would not be computed in the natural logic of MacCartney and Manning (2009).

elements in the range of F.

By our definition, a dataset is fair if it exposes every function to all possible local inputs. This algorithm recursively constructs a fair training dataset. When *ratio* is set to 0 the dataset constructed is minimal and when *ratio* is set to 1 the dataset is the entire space of examples. When this algorithm is called on the root node, it returns a function mapping outputs to sets of inputs and the training dataset is the union of these sets.

When this algorithm is called on an intermediate node it constructs sets of partial inputs that expose this node's function to all possible local inputs. Then these partial inputs are recursively passed to the parent node, where the process is repeated. This ensures that the function of every node is exposed to every local input by the generated training data. According to the value of *ratio*, we constrain how local inputs are realized based on the values of their siblings. For example, the fair training dataset in Table 1 restricts how the truth value of the right argument to $\Rightarrow$ is realized base on the truth value of the left argument.

We experimentally verify that our training dataset is fair, though it is clearly guaranteed to be from the data generation algorithm.

Data: A composition tree (*T*,*Dom*, *Func*) and *ratio* a number between 0 and 1 inclusive.
Result: A training dataset $\mathcal{D}$ that is fair with respect to our baseline model

function `generate_inputs` *(T,Dom, Func, a, ratio)*
  if $a \in N^T_{leaf}$ **then**
    equivalence_classes ← Dict()
    for i ∈ Dom(a) **do**
      equivalence_classes[i] ← $\{i\}$
    end
    return *equivalence_classes*
  else
    $c_1, \ldots c_m \leftarrow$ `children`(a)
    equivalence_classes ← Dict()
    $\mathcal{C} \leftarrow$ *Dom*$(c_1) \times$ *Dom*$(c_2) \cdots \times$ *Dom*(c_m)
    for $(i_1, i_2, \ldots, i_m) \in \mathcal{C}$ **do**
      new_class ← *List*()
      for $k \leftarrow 1 \ldots m$ **do**
        partial_inputs ← `generate_inputs` $(T,$ *Dom*, *Func*, $c_k)$
        split ← `random_even_split` (*partial_inputs*$[i_k]$, `sibling_space` (*a*,*k*), *ratio*)
        new_class.append(*split*$[i_1, \ldots, i_{k-1}, i_{k+1}, \ldots, i_m]$)
      end
      equivalence_classes[*Func*$(a)(i_1, \ldots, i_m)$] ←
        equivalence_classes[*Func*$(a)(i_1, \ldots, i_m)$] ∪ `cartesian_product` (*new_class*)
    end
    return *equivalence_classes*
  end

Algorithm 3: This model generates a training dataset that is fair with respect to our baseline model. The output of `generate_inputs`(*T*, *Dom*, *Func*, *a*, *ratio*) is a function mapping elements of *Dom*(a) to sets of partial inputs that realize the element.