# Adversarially Regularising Neural NLI Models to Integrate Logical Background Knowledge

**Pasquale Minervini**
University College London
p.minervini@cs.ucl.ac.uk

**Sebastian Riedel**
University College London
s.riedel@cs.ucl.ac.uk

## Abstract

Adversarial examples are inputs to machine learning models designed to cause the model to make a mistake. They are useful for understanding the shortcomings of machine learning models, interpreting their results, and for regularisation. In NLP, however, most example generation strategies produce input text by using known, pre-specified semantic transformations, requiring significant manual effort and in-depth understanding of the problem and domain. In this paper, we investigate the problem of automatically generating adversarial examples that violate a set of given First-Order Logic constraints in Natural Language Inference (NLI). We reduce the problem of identifying such adversarial examples to a combinatorial optimisation problem, by maximising a quantity measuring the degree of violation of such constraints and by using a language model for generating linguistically-plausible examples. Furthermore, we propose a method for adversarially regularising neural NLI models for incorporating background knowledge. Our results show that, while the proposed method does not always improve results on the SNLI and MultiNLI datasets, it significantly and consistently increases the predictive accuracy on adversarially-crafted datasets – up to a 79.6% relative improvement – while drastically reducing the number of background knowledge violations. Furthermore, we show that adversarial examples *transfer* among model architectures, and that the proposed adversarial training procedure improves the robustness of NLI models to adversarial examples.

## 1 Introduction

An open problem in Artificial Intelligence is quantifying the extent to which algorithms exhibit intelligent behaviour (Levesque, 2014). In Machine Learning, a standard procedure consists in estimating the *generalisation error*, i.e. the prediction error over an independent test sample (Hastie et al., 2001). However, machine learning models can succeed simply by recognising patterns that happen to be predictive on instances in the test sample, while ignoring deeper phenomena (Rimell and Clark, 2009; Paperno et al., 2016).

Adversarial examples are inputs to machine learning models designed to cause the model to make a mistake (Szegedy et al., 2014; Goodfellow et al., 2014). In Natural Language Processing (NLP) and Machine Reading, generating adversarial examples can be really useful for understanding the shortcomings of NLP models (Jia and Liang, 2017; Kannan and Vinyals, 2017) and for regularisation (Minervini et al., 2017).

In this paper, we focus on the problem of generating adversarial examples for Natural Language Inference (NLI) models in order to gain insights about the inner workings of such systems, and regularising them. NLI, also referred to as Recognising Textual Entailment (Fyodorov et al., 2000; Condoravdi et al., 2003; Dagan et al., 2005), is a central problem in language understanding (Katz, 1972; Bos and Markert, 2005; van Benthem, 2008; MacCartney and Manning, 2009), and thus it is especially well suited to serve as a benchmark task for research in machine reading. In NLI, a model is presented with two sentences, a *premise* $p$ and a *hypothesis* $h$, and the goal is to determine whether $p$ semantically entails $h$.

The problem of acquiring large amounts of labelled data for NLI was addressed with the creation of the SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2017) datasets. In these processes, annotators were presented with a *premise* $p$ drawn from a corpus, and were required to generate three new sentences (*hypotheses*) based on $p$, according to the following criteria: *a)* Entailment – $h$ is definitely true given $p$ ($p$

*Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018)*, pages 65–74
Brussels, Belgium, October 31 - November 1, 2018. ©2018 Association for Computational Linguistics

entails $h$); *b)* Contradiction – $h$ is definitely not true given $p$ ($p$ contradicts $h$); and *c)* Neutral – $h$ might be true given $p$. Given a premise-hypothesis sentence pair $(p, h)$, a NLI model is asked to classify the relationship between $p$ and $h$ – i.e. either *entailment*, *contradiction*, or *neutral*. Solving NLI requires to fully capture the sentence meaning by handling complex linguistic phenomena like lexical entailment, quantification, co-reference, tense, belief, modality, and lexical and syntactic ambiguities (Williams et al., 2017).

In this work, we use adversarial examples for: *a)* identifying cases where models violate existing background knowledge, expressed in the form of *logic rules*, and *b)* training models that are *robust* to such violations.

The underlying idea in our work is that NLI models should adhere to a set of structural constraints that are intrinsic to the human reasoning process. For instance, *contradiction* is inherently *symmetric*: if a sentence $p$ contradicts a sentence $h$, then $h$ contradicts $p$ as well. Similarly, entailment is both *reflexive* and *transitive*. It is reflexive since a sentence $a$ is always entailed by (i.e. is true given) $a$. It is also transitive, since if $a$ is entailed by $b$, and $b$ is entailed by $c$, then $a$ is entailed by $c$ as well.

**Example 1** (Inconsistency)**.** Consider three sentences $a$, $b$ and $c$ each describing a situation, such as: *a)* “The girl plays”, *b)* “The girl plays with a ball”, and *c)* “The girl plays with a red ball”. Note that if $a$ is entailed by $b$, and $b$ is entailed by $c$, then also $a$ is entailed by $c$. If a NLI model detects that $b$ entails $a$, $c$ entails $b$, but $c$ does not entail $a$, we know that it is making an error (since its results are inconsistent), even though we may not be aware of the sentences $a$, $b$, and $c$ and the true semantic relationships holding between them. $\triangle$

Our adversarial examples are different from those used in other fields such as computer vision, where they typically consist in small, semantically invariant perturbations that result in drastic changes in the model predictions. In this paper, we propose a method for generating adversarial examples that cause a model to violate pre-existing background knowledge (Section 4), based on reducing the generation problem to a combinatorial optimisation problem. Furthermore, we outline a method for incorporating such background knowledge into models by means of an *adversarial training* procedure (Section 5).

Our results (Section 8) show that, even though the proposed adversarial training procedure does not sensibly improve accuracy on SNLI and MultiNLI, it yields significant relative improvement in accuracy (up to 79.6%) on adversarial datasets. Furthermore, we show that adversarial examples *transfer* across models, and that the proposed method allows training significantly more robust NLI models.

## 2 Background

**Neural NLI Models.** In NLI, in particular on the Stanford Natural Language Inference (SNLI) (Bowman et al., 2015) and MultiNLI (Williams et al., 2017) datasets, neural NLI models – end-to-end differentiable models that can be trained via gradient-based optimisation – proved to be very successful, achieving state-of-the-art results (Rocktäschel et al., 2016; Parikh et al., 2016; Chen et al., 2017).

Let $\mathcal{S}$ denote the set of all possible sentences, and let $a = (a_1, \ldots, a_{\ell_a}) \in \mathcal{S}$ and $b = (b_1, \ldots, b_{\ell_b}) \in \mathcal{S}$ denote two input sentences – representing the premise and the hypothesis – of length $\ell_a$ and $\ell_b$, respectively. In neural NLI models, all words $a_i$ and $b_j$ are typically represented by $k$-dimensional *embedding vectors* $\mathbf{a}_i, \mathbf{b}_j \in \mathbb{R}^k$. As such, the sentences $a$ and $b$ can be encoded by the sentence *embedding matrices* $\mathbf{a} \in \mathbb{R}^{k \times \ell_a}$ and $\mathbf{b} \in \mathbb{R}^{k \times \ell_b}$, where the columns $\mathbf{a}_i$ and $\mathbf{b}_j$ respectively denote the embeddings of words $a_i$ and $b_j$.

Given two sentences $a, b \in \mathcal{S}$, the goal of a NLI model is to identify the semantic relation between $a$ and $b$, which can be either *entailment*, *contradiction*, or *neutral*. For this reason, given an instance, neural NLI models compute the following conditional probability distribution over all three classes:

$$p_\Theta(\cdot \mid a, b) = \quad \mathrm{softmax}(\mathrm{score}_\Theta(\mathbf{a}, \mathbf{b})) \qquad (1)$$

where $\mathrm{score}_\Theta : \mathbb{R}^{k \times \ell_a} \times \mathbb{R}^{k \times \ell_b} \to \mathbb{R}^3$ is a model-dependent *scoring function* with parameters $\Theta$, and $\mathrm{softmax}(\mathbf{x})_i = \exp\{x_i\} / \sum_j \exp\{x_j\}$ denotes the softmax function.

Several scoring functions have been proposed in the literature, such as the conditional Bidirectional LSTM (cBiLSTM) (Rocktäschel et al., 2016), the Decomposable Attention Model (DAM) (Parikh et al., 2016), and the Enhanced LSTM model (ESIM) (Chen et al., 2017). One desirable quality of the scoring function $\mathrm{score}_\Theta$ is that it should be *differentiable* with respect to the model parameters $\Theta$, which allows the neural NLI model to be trained from data via back-propagation.

**Model Training.** Let $\mathcal{D} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ represent a NLI dataset, where $x_i$ denotes the $i$-th premise-hypothesis sentence pair, and $y_i \in \{1, \ldots, K\}$ their relationship, where $K \in \mathbb{N}$ is the number of possible relationships – in the case of NLI, $K = 3$. The model is trained by minimising a *cross-entropy loss* $\mathcal{J}_{\mathcal{D}}$ on $\mathcal{D}$:

$$\mathcal{J}_{\mathcal{D}}(\mathcal{D}, \Theta) = -\sum_{i=1}^{m} \sum_{k=1}^{K} \mathbb{1}\{y_i = k\} \log(\hat{y}_{i,k}) \quad (2)$$

where $\hat{y}_{i,k} = p_{\Theta}(y_i = k \mid x_i)$ denotes the probability of class $k$ on the instance $x_i$ inferred by the neural NLI model as in Eq. (1).

In the following, we analyse the behaviour of neural NLI models by means of *adversarial examples* – inputs to machine learning models designed to cause the model to commit mistakes. In computer vision models, adversarial examples are created by adding a very small amount of noise to the input (Szegedy et al., 2014; Goodfellow et al., 2014): these perturbations do not change the semantics of the images, but they can drastically change the predictions of computer vision models. In our setting, we define an adversary whose goal is finding *sets* of NLI instances where the model fails to be consistent with available background knowledge, encoded in the form of First-Order Logic (FOL) rules. In the following sections, we define the corresponding optimisation problem, and propose an efficient solution.

## 3 Background Knowledge

For analysing the behaviour of NLI models, we verify whether they agree with the provided background knowledge, encoded by a set of FOL rules. Note that the three NLI classes – *entailment*, *contradiction*, and *neutrality* – can be seen as *binary logic predicates*, and we can define FOL formulas for describing the formal relationships that hold between them.

In the following, we denote the predicates associated with entailment, contradiction, and neutrality as $\text{ent}$, $\text{con}$, and $\text{neu}$, respectively. By doing so, we can represent semantic relationships between sentences via logic atoms. For instance, given three sentences $s_1, s_2, s_3 \in \mathcal{S}$, we can represent the fact that $s_1$ entails $s_2$ and $s_2$ contradicts $s_3$ by using the logic atoms $\text{ent}(s_1, s_2)$ and $\text{con}(s_2, s_3)$.

| | NLI Rules |
|---|---|
| $\mathbf{R_1}$ | $\top \Rightarrow \text{ent}(X_1, X_1)$ |
| $\mathbf{R_2}$ | $\text{con}(X_1, X_2) \Rightarrow \text{con}(X_2, X_1)$ |
| $\mathbf{R_3}$ | $\text{ent}(X_1, X_2) \Rightarrow \neg\text{con}(X_2, X_1)$ |
| $\mathbf{R_4}$ | $\text{neu}(X_1, X_2) \Rightarrow \neg\text{con}(X_2, X_1)$ |
| $\mathbf{R_5}$ | $\text{ent}(X_1, X_2) \wedge \text{ent}(X_2, X_3) \Rightarrow \text{ent}(X_1, X_3)$ |

Table 1: First-Order Logic rules defining desired properties of NLI models: $X_i$ are universally quantified variables, and operators $\wedge$, $\neg$, and $\top$ denote logic conjunction, negation, and tautology.

Let $X_1, \ldots, X_n$ be a set of universally quantified variables. We define our background knowledge as a set of FOL rules, each having the following $\text{body} \Rightarrow \text{head}$ form:

$$\text{body}(X_1, \ldots, X_n) \Rightarrow \text{head}(X_1, \ldots, X_n), \quad (3)$$

where $\text{body}$ and $\text{head}$ represent the *premise* and the *conclusion* of the rule – if $\text{body}$ holds, $\text{head}$ holds as well. In the following, we consider the rules $\mathbf{R_1}, \ldots, \mathbf{R_5}$ outlined in Table 1. Rule $\mathbf{R_1}$ enforces the constraint that entailment is reflexive; rule $\mathbf{R_2}$ that contradiction should always be symmetric (if $s_1$ contradicts $s_2$, then $s_2$ contradicts $s_1$ as well); rule $\mathbf{R_5}$ that entailment is transitive; while rules $\mathbf{R_3}$ and $\mathbf{R_4}$ describe the formal relationships between the *entailment*, *neutral*, and *contradiction* relations.

In Section 4 we propose a method to automatically generate sets of sentences that violate the rules outlined in Table 1 – effectively generating *adversarial examples*. Then, in Section 5 we show how we can leverage such adversarial examples by generating them on-the-fly during training and using them for regularising the model parameters, in an *adversarial training* regime.

## 4 Generating Adversarial Examples

In this section, we propose a method for efficiently *generating* adversarial examples for NLI models – i.e. examples that make the model violate the background knowledge outlined in Section 3.

### 4.1 Inconsistency Loss

We cast the problem of generating adversarial examples as an optimisation problem. In particular, we propose a continuous *inconsistency loss* that measures the *degree* to which a set of sentences causes a model to violate a rule.

**Example 2** (Inconsistency Loss)**.** Consider the rule $\mathbf{R_2}$ in Table 1, i.e. $\text{con}(X_1, X_2) \Rightarrow \text{con}(X_2, X_1)$. Let $s_1, s_2 \in \mathcal{S}$ be two sentences: this rule is violated if, according to the model, a sentence $s_1$ contradicts $s_2$, but $s_2$ does not contradict $s_1$. However, if we just use the final decision made by the neural NLI model, we can simply check whether the rule is violated by two given sentences, without any information on the *degree* of such a violation.

Intuitively, for the rule being *maximally violated*, the conditional probability associated to $\text{con}(s_1, s_2)$ should be *very high* ($\approx 1$), while the one associated to $\text{con}(s_2, s_1)$ should be *very low* ($\approx 0$). We can measure the extent to which the rule is violated – which we refer to as *inconsistency loss* $\mathcal{J}_\mathcal{I}$ – by checking whether the probability of the body of the rule is higher than the probability of its head:

$$\begin{aligned}&\mathcal{J}_\mathcal{I}(S = \{X_1 \mapsto s_1, X_2 \mapsto s_2\}) \\ &\qquad = [p_\Theta(\text{con} \mid s_1, s_2) - p_\Theta(\text{con} \mid s_2, s_1)]_+\end{aligned}$$

where $S$ is a *substitution set* that maps the variables $X_1$ and $X_2$ in $\mathbf{R_2}$ to the sentences $s_1$ and $s_2$, $[x]_+ = \max(0, x)$, and $p_\Theta(\text{con} \mid s_i, s_j)$ is the (conditional) probability that $s_i$ contradicts $s_j$ according to the neural NLI model. Note that, in accordance with the logic implication, the inconsistency loss reaches its global minimum when the probability of the body is close to zero – i.e. the *premise* is false – and when the probabilities of both the body and the head are close to one – i.e. the *premise* and the *conclusion* are both true. △

We now generalise the intuition in Ex. 2 to any FOL rule. Let $r = (\text{body} \Rightarrow \text{head})$ denote an arbitrary FOL rule in the form described in Eq. (3), and let $\text{vars}(r) = \{X_1, \ldots, X_n\}$ denote the set of universally quantified variables in the rule $r$.

Furthermore, let $S = \{X_1 \mapsto s_1, \ldots, X_n \mapsto s_n\}$ denote a *substitution set*, i.e. a mapping from variables in $\text{vars}(r)$ to sentences $s_1, \ldots, s_n \in \mathcal{S}$. The inconsistency loss associated with the rule $r$ on the substitution set $S$ can be defined as:

$$\mathcal{J}_\mathcal{I}(S) = [p(S; \text{body}) - p(S; \text{head})]_+ \qquad (4)$$

where $p(S; \text{body})$ and $p(S; \text{head})$ denote the probability of body and head of the rule, after replacing the variables in $r$ with the corresponding sentences in $S$. The motivation for the loss in Eq. (4) is that logic implications can be understood as "whenever the body is true, the head has to be true as well". In terms of NLI models, this translates as "the probability of the head should at least be as large as the probability of the body".

For calculating the inconsistency loss in Eq. (4), we need to specify how to calculate the probability of $\text{head}$ and $\text{body}$. The probability of a single ground atom is given by querying the neural NLI model, as in Eq. (1). The head contains a single atom, while the body can be a conjunction of multiple atoms. Similarly to Minervini et al. (2017), we use the Gödel t-norm, a continuous generalisation of the conjunction operator in logic (Gupta and Qi, 1991), for computing the probability of the body of a clause:

$$p_\Theta(a_1 \wedge a_2) = \min\{p_\Theta(a_1), p_\Theta(a_2)\}$$

where $a_1$ and $a_2$ are two clause atoms.

In this work, we cast the problem of generating adversarial examples as an optimisation problem: we search for the substitution set $S = \{X_1 \mapsto s_1, \ldots, X_n \mapsto s_n\}$ that maximises the inconsistency loss in Eq. (4), thus (maximally) violating the available background knowledge.

### 4.2 Constraining via Language Modelling

Maximising the inconsistency loss in Eq. (4) may not be sufficient for generating meaningful adversarial examples: they can lead neural NLI models to violate available background knowledge, but they may not be well-formed and meaningful.

For such a reason, in addition to maximising the inconsistency loss, we also constrain the *perplexity* of generated sentences by using a neural language model (Bengio et al., 2000). In this work, we use a LSTM (Hochreiter and Schmidhuber, 1997) neural language model $p_\mathcal{L}(w_1, \ldots, w_t)$ for generating low-perplexity adversarial examples.

### 4.3 Searching in a Discrete Space

As mentioned earlier in this section, we cast the problem of automatically generating adversarial examples – i.e. examples that cause NLI models to violate available background knowledge – as an optimisation problem. Specifically, we look for substitutions sets $S = \{X_1 \mapsto s_1, \ldots, X_n \mapsto s_n\}$ that jointly: *a*) maximise the *inconsistency loss* described in Eq. (4), and *b*) are composed by sentences with a low perplexity, as defined by the neural language model in Section 4.2.

The search objective can be formalised by the following optimisation problem:

$$\begin{aligned} \underset{S}{\text{maximise}} \quad & \mathcal{J}_{\mathcal{I}}(S) \\ \text{subject to} \quad & \log p_{\mathcal{L}}(S) \leq \tau \end{aligned} \tag{5}$$

where $\log p_{\mathcal{L}}(S)$ denotes the log-probability of the sentences in the substitution set $S$, and $\tau$ is a threshold on the perplexity of generated sentences.

For generating low-perplexity adversarial examples, we take inspiration from Guu et al. (2017) and generate the sentences by editing prototypes extracted from a corpus. Specifically, for searching substitution sets whose sentences jointly have a high probability and are highly adversarial, as measured the inconsistency loss in Eq. (4), we use the following procedure, also described in Appendix A.4: *a)* we first sample sentences close to the data manifold (i.e. with a low perplexity), by either sampling from the training set or from the language model; *b)* we then make small variations to the sentences – analogous to adversarial images, which consist in small perturbations of training examples – so to optimise the objective in Eq. (5).

When editing prototypes, we consider the following perturbations: *a)* change one word in one of the input sentences; *b)* remove one parse subtree from one of the input sentences; *c)* insert one parse sub-tree from one sentence in the corpus in the parse tree of one of the input sentences.

Note that the generation process can easily lead to ungrammatical or implausible sentences; however, these will be likely to have a high perplexity according to the language model (Section 4.2), and thus they will be ruled out by the search algorithm.

## 5 Adversarial Regularisation

We now show one can use the adversarial examples to regularise the training process. We propose training NLI models by jointly: *a)* minimising the data loss (Eq. (2)), and *b)* minimising the inconsistency loss (Eq. (4)) on a set of generated adversarial examples (substitution sets).

More formally, for training, we jointly minimise the cross-entropy loss defined on the data $\mathcal{J}_{\mathcal{D}}(\Theta)$ and the inconsistency loss on a set of generated adversarial examples $\max_S \mathcal{J}_{\mathcal{I}}(S; \Theta)$, resulting in the following optimisation problem:

$$\begin{aligned} \underset{\Theta}{\text{minimise}} \quad & \mathcal{J}_{\mathcal{D}}(\mathcal{D}, \Theta) + \lambda \max_{S} \mathcal{J}_{\mathcal{I}}(S; \Theta) \\ \text{subject to} \quad & \log p_{\mathcal{L}}(S) \leq \tau \end{aligned} \tag{6}$$

where $\lambda \in \mathbb{R}_+$ is a hyperparameter specifying the trade-off between the data loss $\mathcal{J}_{\mathcal{D}}$ (Eq. (2)), and the inconsistency loss $\mathcal{J}_{\mathcal{I}}$ (Eq. (4)), measured on the generated substitution set $S$.

In Eq. (6), the regularisation term $\max_S \mathcal{J}_{\mathcal{I}}(S; \Theta)$ has the task of *generating* the adversarial substitution sets by maximising the inconsistency loss. Furthermore, the constraint $\log p_{\mathcal{L}}(S) \leq \tau$ ensures that the perplexity of generated sentences is lower than a threshold $\tau$. For this work, we used the $\max$ aggregation function. However, other functions can be used as well, such as the sum or mean of multiple inconsistency losses.

For minimising the regularised loss in Eq. (6), we alternate between two optimisation processes – generating the adversarial examples (Eq. (5)) and minimising the regularised loss (Eq. (6)). The algorithm is outlined in Appendix A.4: at each iteration, after generating a set of adversarial examples $S$, it computes the gradient of the regularised loss in Eq. (6), and updates the model parameters via a gradient descent step.

| | |
|---|---|
| **Premise** | A man in a suit walks through a train station. |
| **Hypothesis** | Two boys ride skateboard. |
| **Type** | **Contradiction** |
| **Premise** | Two boys ride skateboard. |
| **Hypothesis** | A man in a suit walks through a train station. |
| **Type** | **Contradiction** |
| **Premise** | Two people are surfing in the ocean. |
| **Hypothesis** | There are people outside. |
| **Type** | **Entailment** |
| **Premise** | There are people outside. |
| **Hypothesis** | Two people are surfing in the ocean. |
| **Type** | **Neutral** |

Table 2: Sample sentences from an Adversarial NLI Dataset generated using the DAM model, by maximising the inconsistency loss $\mathcal{J}_{\mathcal{I}}$.

## 6 Creating Adversarial NLI Datasets

We crafted a series of datasets for assessing the robustness of the proposed regularisation method to adversarial examples. Starting from the SNLI test set, we proceeded as follows. We selected the $k$ instances in the SNLI test set that maximise the inconsistency loss in Eq. (4) with respect to the rules in $\mathbf{R_1}$, $\mathbf{R_2}$, $\mathbf{R_3}$, and $\mathbf{R_4}$ in Table 1. We refer to the generated datasets as $\mathcal{A}^k_{\text{m}}$, where $m$ identifies the model used for selecting the sentence pairs, and $k$ denotes number of examples in the dataset.

For generating each of the $\mathcal{A}^k_{\mathrm{m}}$ datasets, we proceeded as follows. Let $\mathcal{D} = \{(x_1, y_i), \ldots, (x_n, y_n)\}$ be a NLI dataset (such as SNLI), where each instance $x_i = (p_i, h_i)$ is a premise-hypothesis sentence pair, and $y_i$ denotes the relationship holding between $p_i$ and $h_i$. For each instance $x_i = (p_i, h_i)$, we consider two substitution sets: $S_i = \{X_1 \mapsto p_i, X_2 \mapsto h_i\}$ and $S'_i = \{X_1 \mapsto h_i, X_2 \mapsto p_i\}$, each corresponding to a mapping from variables to sentences.

We compute the *inconsistency score* associated to each instance $x_i$ in the dataset $\mathcal{D}$ as $\mathcal{J}_{\mathcal{I}}(S_i) + \mathcal{J}_{\mathcal{I}}(S'_i)$. Note that the inconsistency score only depends on the premise $p_i$ and hypothesis $h_i$ in each instance $x_i$, and it does not depend on its label $y_i$.

After computing the inconsistency scores for all sentence pairs in $\mathcal{D}$ using a model $m$, we select the $k$ instances with the highest inconsistency score, we create two instances $x_i = (p_i, h_i)$ and $\hat{x}_i = (h_i, p_i)$, and add both $(x_i, y_i)$ and $(\hat{x}_i, \hat{y}_i)$ to the dataset $\mathcal{A}^k_{\mathrm{m}}$. Note that, while $y_i$ is already known from the dataset $\mathcal{D}$, $\hat{y}_i$ is unknown. For this reason, we find $\hat{y}_i$ by manual annotation.

## 7 Related Work

Adversarial examples are receiving a considerable attention in NLP; their usage, however, is considerably limited by the fact that semantically invariant input perturbations in NLP are difficult to identify (Buck et al., 2017).

Jia and Liang (2017) analyse the robustness of extractive question answering models on examples obtained by adding adversarially generated distracting text to SQuAD (Rajpurkar et al., 2016) dataset instances. Belinkov and Bisk (2017) also notice that character-level Machine Translation are overly sensitive to random character manipulations, such as typos. Hosseini et al. (2017) show that simple character-level modifications can drastically change the toxicity score of a text. Iyyer et al. (2018) proposes using paraphrasing for generating adversarial examples. Our model is fundamentally different in two ways: *a)* it does not need labelled data for generating adversarial examples – the inconsistency loss can be maximised by just making an NLI model produce inconsistent results, and *b)* it incorporates adversarial examples during the training process, with the aim of training more robust NLI models.

Adversarial examples are also used for assessing the robustness of computer vision models (Szegedy et al., 2014; Goodfellow et al., 2014; Nguyen et al., 2015), where they are created by adding a small amount of noise to the inputs that does not change the semantics of the images, but drastically changes the model predictions.

|  | **Model** | **Original** |  | **Regularised** |  |
|---|---|---|---|---|---|
|  |  | Valid. | Test | Valid. | Test |
| **MultiNLI** | cBiLSTM | 61.52 | 63.95 | **66.98** | **66.68** |
|  | DAM | 72.78 | 73.28 | **73.57** | **73.51** |
|  | ESIM | 73.66 | 75.22 | **75.72** | **75.80** |
| **SNLI** | cBiLSTM | 81.41 | 80.99 | **82.27** | **81.12** |
|  | DAM | 86.96 | 86.29 | **87.08** | **86.43** |
|  | ESIM | 87.83 | 87.25 | **87.98** | **87.55** |

Table 3: Accuracy on the SNLI and MultiNLI datasets with different neural NLI models *before* (left) and *after* (right) adversarial regularisation.

| **Model** | **Rule** | $\lvert\mathbf{B}\rvert$ | $\lvert\mathbf{B} \wedge \neg\mathbf{H}\rvert$ | **Violations** (%) |
|---|---|---|---|---|
| cBiLSTM | $\mathbf{R_1}$ | 1,098,734 | 261,064 | 23.76 % |
|  | $\mathbf{R_2}$ | 174,902 | 80,748 | 46.17 % |
|  | $\mathbf{R_3}$ | 197,697 | 24,294 | 12.29 % |
|  | $\mathbf{R_4}$ | 176,768 | 33,435 | 18.91 % |
| DAM | $\mathbf{R_1}$ | 1,098,734 | 956 | 00.09 % |
|  | $\mathbf{R_2}$ | 171,728 | 28,680 | 16.70 % |
|  | $\mathbf{R_3}$ | 196,042 | 11,599 | 05.92 % |
|  | $\mathbf{R_4}$ | 181,597 | 29,635 | 16.32 % |
| ESIM | $\mathbf{R_1}$ | 1,098,734 | 10,985 | 01.00 % |
|  | $\mathbf{R_2}$ | 177,950 | 17,518 | 09.84 % |
|  | $\mathbf{R_3}$ | 200,852 | 6,482 | 03.23 % |
|  | $\mathbf{R_4}$ | 170,565 | 17,190 | 10.08 % |

Table 4: Violations (%) of rules $\mathbf{R_1}, \mathbf{R_2}, \mathbf{R_3}, \mathbf{R_4}$ from Table 1 on the SNLI training set, yield by cBiLSTM, DAM, and ESIM.

## 8 Experiments

We trained DAM, ESIM and cBiLSTM on the SNLI corpus using the hyperparameters provided in the respective papers. The results provided by such models on the SNLI and MultiNLI validation and tests sets are provided in Table 3. In the case of MultiNLI, the validation set was obtained by removing 10,000 instances from the training set (originally composed by 392,702 instances), and the test set consists in the *matched* validation set.

**Background Knowledge Violations.** As a first experiment, we count the how likely our model is to violate rules $\mathbf{R_1}, \mathbf{R_2}, \mathbf{R_3}, \mathbf{R_4}$ in Table 1.

In Table 4 we report the number sentence pairs in the SNLI training set where DAM, ESIM and cBiLSTM violate $\mathbf{R_1}, \mathbf{R_2}, \mathbf{R_3}, \mathbf{R_4}$. In the $|\mathbf{B}|$ column we report the number of times the body

| Model \ Dataset | $\mathcal{A}^{100}_{\text{DAM}}$ | $\mathcal{A}^{500}_{\text{DAM}}$ | $\mathcal{A}^{1000}_{\text{DAM}}$ | $\mathcal{A}^{100}_{\text{ESIM}}$ | $\mathcal{A}^{500}_{\text{ESIM}}$ | $\mathcal{A}^{1000}_{\text{ESIM}}$ | $\mathcal{A}^{100}_{\text{cBiLSTM}}$ | $\mathcal{A}^{500}_{\text{cBiLSTM}}$ | $\mathcal{A}^{1000}_{\text{cBiLSTM}}$ |
|---|---|---|---|---|---|---|---|---|---|
| DAM$^{\mathcal{AR}}$ | **83.33** | **79.15** | **79.37** | **71.35** | **72.19** | **70.05** | **93.00** | **88.99** | **86.00** |
| DAM | 47.40 | 47.93 | 51.66 | 55.73 | 60.94 | 60.88 | 81.50 | 77.37 | 75.28 |
| ESIM$^{\mathcal{AR}}$ | **89.06** | **86.00** | **85.08** | **78.12** | **76.04** | **73.32** | **96.50** | **91.92** | **88.52** |
| ESIM | 72.40 | 74.59 | 76.92 | 52.08 | 58.65 | 60.78 | 87.00 | 84.34 | 82.05 |
| cBiLSTM$^{\mathcal{AR}}$ | **85.42** | **80.39** | **78.74** | **73.96** | **70.52** | **65.39** | **92.50** | **88.38** | **83.62** |
| cBiLSTM | 56.25 | 59.96 | 61.75 | 47.92 | 53.23 | 53.73 | 51.50 | 52.83 | 53.24 |

Table 5: Accuracy of unregularised and regularised neural NLI models DAM, cBiLSTM, and ESIM, and their adversarially regularised versions DAM$^{\mathcal{AR}}$, cBiLSTM$^{\mathcal{AR}}$, and ESIM$^{\mathcal{AR}}$, on adversarial datasets $\mathcal{A}^{k}_{\text{m}}$.

of the rule holds, according to the model. In the $|\mathbf{B} \wedge \neg \mathbf{H}|$ column we report the number of times where the body of the rule holds, but the head does not – which is clearly a violation of available rules.

We can see that, in the case of rule $\mathbf{R_1}$ (reflexivity of entailment), DAM and ESIM make a relatively low number of violations – namely 0.09 and 1.00 %, respectively. However, in the case of cBiLSTM, we can see that, each sentence $s \in \mathcal{S}$ in the SNLI training set, with a 23.76 % chance, $s$ does not entail itself – which violates our background knowledge.

With respect to $\mathbf{R_2}$ (symmetry of contradiction), we see that none of the models is completely consistent with the available background knowledge. Given a sentence pair $s_1, s_2 \in \mathcal{S}$ from the SNLI training set, if – according to the model – $s_1$ contradicts $s_2$, a significant number of times (between 9.84% and 46.17%) the same model also infers that $s_2$ *does not* contradict $s_1$. This phenomenon happens 16.70 % of times with DAM, 9.84 % of times with ESIM, and 46.17 % with cBiLSTM: this indicates that all considered models are prone to violating $\mathbf{R_2}$ in their predictions, with ESIM being the more robust.

In Appendix A.2 we report several examples of such violations in the SNLI training set. We select those that maximise the inconsistency loss described in Eq. (4), violating rules $\mathbf{R_2}$ and $\mathbf{R_3}$. We can notice that the presence of inconsistencies is often correlated with the length of the sentences. The model tends to detect entailment relationships between longer (i.e., possibly more specific) and shorter (i.e., possibly more general) sentences.

## 8.1 Generation of Adversarial Examples

In the following, we analyse the automatic generation of sets of adversarial examples that make the model violate the existing background knowledge. We search in the space of sentences by applying perturbations to sampled sentence pairs, using a language model for guiding the search process. The generation procedure is described in Section 4.

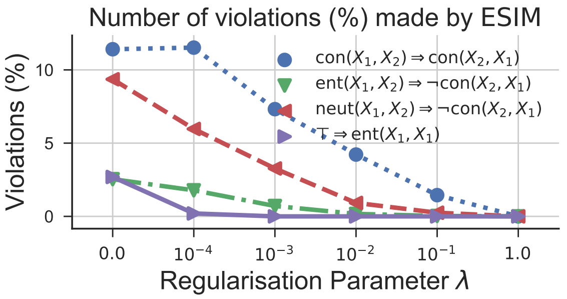

Figure 1: Number of violations (%) to rules in Table 1 made by ESIM on the SNLI test set.

The procedure was especially effective in generating adversarial examples – a sample is shown in Table 6. We can notice that, even though DAM and ESIM achieve results close to human level performance on SNLI, they are likely to fail when faced with linguistic phenomena such as negation, hyponymy, and antonymy. Gururangan et al. (2018) recently showed that NLI datasets tend to suffer from annotation artefacts and limited linguistic variations: this allows NLI models to achieve nearly-human performance by capturing repetitive patterns and idiosyncrasies in a dataset, without being able of effectively capturing textual entailment. This is visible, for instance, in example 5 of Table 6, where the model fails to capture the hyponymy relation between “male” and “man”, incorrectly predicting an entailment in place of a neutral relationship. Furthermore, it is clear that models lack commonsense knowledge, such as the relation between “pushing” and “carrying” (example 1), and being outside and swimming (example 2). Generating such adversarial

|  |  | Adversarial Example | Prediction | Inconsistency |
|---|---|---|---|---|
| 1 | $s_1$ | A man in uniform is pushing a medical bed. | $\boxed{s_1 \xrightarrow{0.72} s_2}$ | .01 ⇝ .92 |
|  | $s_2$ | a man is ~~pushing~~ carrying something. | $s_2 \xrightarrow{0.93} s_1$ |  |
| 1 | $s_1$ | A dog swims in the water | $s_1 \xrightarrow{0.78} s_2$ | .00 ⇝ .99 |
|  | $s_2$ | A dog is ~~swimming~~ outside. | $\boxed{s_2 \xrightarrow{0.99} s_1}$ |  |
| 2 | $s_1$ | A young man is sledding down a snow covered hill on a green sled. | $s_1 \xrightarrow{0.98} s_2$ | .00 ⇝ .97 |
|  | $s_1$ | A man is ~~sledding~~ down to meet his daughter. | $\boxed{s_2 \xrightarrow{1.00} s_1}$ |  |
| 3 | $s_1$ | A woman sleeps on the ground. A boy and girl play in a pool. | $s_1 \xrightarrow{0.94} s_2$ | .00 ⇝ .82 |
|  | $s_2$ | Two kids are happily playing in a swimming pool. | $\boxed{s_2 \xrightarrow{0.85} s_1}$ |  |
| 4 | $s_1$ | The school is having a special event in order to show the american culture on how other cultures are dealt with in parties. | $s_1 \xrightarrow{0.96} s_2$ | .01 ⇝ .63 |
|  | $s_2$ | A ~~school~~ dog is hosting an event. | $\boxed{s_2 \xrightarrow{0.66} s_1}$ |  |
| 5 | $s_1$ | A boy is drinking out of a water fountain shaped like a woman. | $s_1 \xrightarrow{0.96} s_2$ | .00 ⇝ .94 |
|  | $s_2$ | A male is getting a drink of water. | $\boxed{s_2 \xrightarrow{0.93} s_3}$ |  |
|  | $s_3$ | A ~~male~~ man is getting a drink of water. | $s_1 \xrightarrow{0.97} s_3$ |  |

Table 6: Inconsistent results produced by DAM on automatically generated adversarial examples. The notation ~~segment one~~ segment two denotes that the corruption process removes "segment one" and introduced "segment two" in the sentence, and $s_1 \xrightarrow{p} s_2$ indicates that DAM classifies the relation between $s_1$ and $s_2$ as *contradiction*, with probability $p$. We use different colours for representing the contradiction, entailment and neutral classes. Examples 1, 2, 3, and 4 violate the rule $\mathbf{R_2}$, while example 5 violates the rule $\mathbf{R_5}$. .00 ⇝ .99 indicates that the corruption process increases the inconsistency loss from .00 to .99, and the $\boxed{\text{red boxes}}$ are used for indicating mistakes made by the model on the adversarial examples.

examples provides us with useful insights on the inner workings of neural NLI models, that can be leveraged for improving the robustness of state-of-the-art models.

## 8.2 Adversarial Regularisation

We evaluated whether our approach for integrating logical background knowledge via adversarial training (Section 5) is effective at reducing the number of background knowledge violations, without reducing the predictive accuracy of the model. We started with pre-trained DAM, ESIM, and cBiLSTM models, trained using the hyperparameters published in their respective papers.

After training, each model was then fine-tuned for 10 epochs, by minimising the adversarially regularised loss function introduced in Eq. (6). Table 3 shows results on the SNLI and MultiNLI development and test set, while Fig. 1 shows the number of violations for different values of $\lambda$, where regularised models are much more likely to make predictions that are consistent with the available background knowledge.

We can see that, despite the drastic reduction of background knowledge violations, the improvement may not be significant, supporting the idea that models achieving close-to-human performance on SNLI and MultiNLI may be capturing annotation artefacts and idiosyncrasies in such datasets (Gururangan et al., 2018).

**Evaluation on Adversarial Datasets.** We evaluated the proposed approach on 9 adversarial datasets $\mathcal{A}^k_{\text{m}}$, with $k \in \{100, 500, 1000\}$, generated following the procedure described in Section 6 – results are summarised in Table 5. We can see that the proposed adversarial training method significantly increases the accuracy on the adversarial test sets. For instance, consider $\mathcal{A}^{100}_{\text{DAM}}$: prior to regularising ($\lambda = 0$), DAM achieves a very low accuracy on this dataset – i.e. 47.4%. By increasing the regularisation parameter $\lambda \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, we noticed sensible accuracy increases, yielding relative accuracy improvements up to 75.8% in the case of DAM, and 79.6% in the case of cBiLSTM.

From Table 5 we can notice that adversarial examples *transfer* across different models: an unregularised model is likely to perform poorly also on adversarial datasets generated by using different models, with ESIM being the more robust model to adversarially generated examples. Furthermore, we can see that regularised models are generally more robust to adversarial examples, even when those were generated using different model architectures. For instance we can see that, while cBiLSTM is vulnerable also to adversarial examples generated using DAM and ESIM, its adversarially regularised version cBiLSTM$^{\mathcal{AR}}$ is generally more robust to any sort of adversarial examples.

## 9 Conclusions

In this paper, we investigated the problem of automatically generating adversarial examples that violate a set of given First-Order Logic constraints in NLI. We reduced the problem of identifying such adversarial examples to an optimisation problem, by maximising a continuous relaxation of the violation of such constraints, and by using a language model for generating linguistically-plausible examples. Furthermore, we proposed a method for adversarially regularising neural NLI models for incorporating background knowledge.

Our results showed that the proposed method consistently yields significant increases to the predictive accuracy on adversarially-crafted datasets – up to a 79.6% relative improvement – while drastically reducing the number of background knowledge violations. Furthermore, we showed that adversarial examples transfer across model architectures, and the proposed adversarial training procedure produces generally more robust models. The source code and data for reproducing our results is available online, at https://github.com/uclmr/adversarial-nli/.

### Acknowledgements

We are immensely grateful to Jeff Mitchell, Johannes Welbl, and the whole UCL Machine Reading research group for all useful discussions, inputs, and ideas. This work has been supported by an Allen Distinguished Investigator Award, and a Marie Curie Career Integration Award.

## References

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000*, pages 932–938. MIT Press.

Johan van Benthem. 2008. A brief history of natural logic. In M. Chakraborty, B. Löwe, M. Nath Mitra, and S. Sarukki, editors, *Logic, Navya-Nyaya and Applications: Homage to Bimal Matilal*. College Publications.

Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 628–635. The Association for Computational Linguistics.

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, EMNLP 2015*, pages 632–642. The Association for Computational Linguistics.

Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Andrea Gesmundo, Neil Houlsby, Wojciech Gajewski, and Wei Wang. 2017. Ask the right questions: Active question reformulation with reinforcement learning. *CoRR*, abs/1705.07830.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 1657–1668. Association for Computational Linguistics.

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow. 2003. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 38–45.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005*, volume 3944 of *LNCS*, pages 177–190. Springer.

Yaroslav Fyodorov, Yoad Winter, and Nissim Francez. 2000. A natural logic inference system. In *Proceedings of the of the 2nd Workshop on Inference in Computational Semantics*.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572.

M. M. Gupta and J. Qi. 1991. Theory of t-norms and fuzzy inference methods. *Fuzzy Sets Syst.*, 40(3):431–450.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. *CoRR*, abs/1803.02324.

Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2017. Generating sentences by editing prototypes. *CoRR*, abs/1709.08878.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hossein Hosseini, Baicen Xiao, and Radha Poovendran. 2017. Deceiving google's cloud video intelligence API built for summarizing videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*, pages 1305–1309. IEEE Computer Society.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *CoRR*, abs/1804.06059.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 2011–2021. Association for Computational Linguistics.

Anjuli Kannan and Oriol Vinyals. 2017. Adversarial evaluation of dialogue models. *CoRR*, abs/1701.08198.

J.J. Katz. 1972. *Semantic theory*. Studies in language. Harper & Row.

Hector J. Levesque. 2014. On our best behaviour. *Artif. Intell.*, 212:27–35.

Bill MacCartney and Christopher D Manning. 2009. An extended model of natural logic. In *Proceedings of the of the Eighth International Conference on Computational Semantics*, Tilburg, Netherlands.

Pasquale Minervini, Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2017. Adversarial sets for regularising neural link predictors. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017*. AUAI Press.

Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pages 427–436. IEEE Computer Society.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*. The Association for Computer Linguistics.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In (Su et al., 2016), pages 2249–2255.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In (Su et al., 2016), pages 2383–2392.

Laura Rimell and Stephen Clark. 2009. Porting a lexicalized-grammar parser to the biomedical domain. *Journal of Biomedical Informatics*, 42(5):852–865.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomas Kocisky, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *International Conference on Learning Representations (ICLR)*.

Jian Su et al., editors. 2016. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*. The Association for Computational Linguistics.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *CoRR*, abs/1704.05426.