

NatLogAttack: A Framework for Attacking Natural Language Inference Models with Natural Logic

Zi'ou Zheng & Xiaodan Zhu

Department of Electrical and Computer Engineering & Ingenuity Labs Research Institute
Queen's University
{ziou.zheng,xiaodan.zhu}@queensu.ca

Abstract

Reasoning has been a central topic in artificial intelligence from the beginning. The recent progress made on distributed representation and neural networks continues to improve the state-of-the-art performance of natural language inference. However, it remains an open question whether the models perform real reasoning to reach their conclusions or rely on spurious correlations. Adversarial attacks have proven to be an important tool to help evaluate the Achilles' heel of the victim models. In this study, we explore the fundamental problem of developing attack models based on logic formalism. We propose NatLogAttack to perform systematic attacks centring around *natural logic*, a classical logic formalism that is traceable back to Aristotle's syllogism and has been closely developed for natural language inference. The proposed framework renders both label-preserving and label-flipping attacks. We show that compared to the existing attack models, NatLogAttack generates better adversarial examples with fewer visits to the victim models. The victim models are found to be more vulnerable under the label-flipping setting. NatLogAttack provides a tool to probe the existing and future NLI models' capacity from a key viewpoint and we hope more logic-based attacks will be further explored for understanding the desired property of reasoning.¹

1 Introduction

While deep neural networks have achieved the state-of-the-art performance on a wide range of tasks, the models are often vulnerable and easily deceived by imposing perturbations to the original input (Goodfellow et al., 2014; Kurakin et al., 2018), which seriously hurts the accountability of the systems. In depth, this pertains to model robustness, capacity, and the development of models with more advanced intelligence.

¹The code of NatLogAttack is available at <https://github.com/orianna-zzo/NatLogAttack>.

Natural language inference (NLI), also known as textual entailment (Dagan et al., 2005; Iftene and Balahur-Dobrescu, 2007; MacCartney, 2009; Bowman et al., 2015), is a fundamental problem that models the inferential relationships between a premise and hypothesis sentence. The models built on *distributed* representation have significantly improved the performance on different benchmarks (Bowman et al., 2015; Chen et al., 2017; Williams et al., 2018; Chen et al., 2018; Devlin et al., 2019; Liu et al., 2019; Zhang et al., 2020; Pilault et al., 2021). However, it is still highly desirable to conduct research to probe if the models possess the desired reasoning ability rather than rely on spurious correlation to reach their conclusions (Glockner et al., 2018; Poliak et al., 2018; Belinkov et al., 2019; McCoy et al., 2019; Richardson et al., 2020).

Adversarial attacks have proven to be an important tool to reveal the Achilles' heel of victim models. Specifically for natural language inference, the logic relations are easily broken if an attack model does not properly generate the adversarial examples following the logic relations and related semantics. Therefore, unlike other textual attack tasks such as those relying on semantic similarity and relatedness, it is more challenging to create effective attacks here.

In this study, we explore the basic problem of developing adversarial attacks based on logic formalism, with the aim to probe victim models for the desired reasoning capability. Specifically, we propose NatLogAttack, in which the adversarial attacks are generated based on *natural logic* (Lakoff, 1970; Van Benthem, 1995; MacCartney, 2009; Icard, 2012; Angeli et al., 2016; Hu and Moss, 2018; Chen et al., 2021), a classical logic formalism with a long history that has been closely developed with natural language inference. From a general perspective, natural language inference provides an appropriate setup for probing the development of *distributed representation* and the

models based on that. A robust solution for the task requires manipulation of discrete operations and adversarial attacks can help understand whether and how the required symbols and inference steps emerge from the data and the learned distributed representation. Our work has also been inspired by recent research on exploring the complementary strengths of neural networks and symbolic models (Garcez et al., 2015; Yang et al., 2017; Rocktäschel and Riedel, 2017; Evans and Grefenstette, 2018; Weber et al., 2019; De Raedt et al., 2019; Mao et al., 2019; Feng et al., 2020, 2022).

Our research contributes to the development of logic-based adversarial attacks for natural language understanding. Specifically, we propose a novel attack framework, NatLogAttack, based on natural logic for natural language inference. Our experiments with both human and automatic evaluation show that the proposed model outperforms the state-of-the-art attack methods. Compared to the existing attack models, NatLogAttack generates better adversarial examples with fewer visits to the victim models. In addition to the commonly used attack setting where the labels of generated examples remain the same as the original pairs, we also propose to construct label-flipping attacks. The victim models are found to be more vulnerable in this setup and NatLogAttack succeeds in deceiving them with much smaller numbers of queries. NatLogAttack provides a systematic approach to probing the existing and future NLI models' capacity from a basic viewpoint that has a traceable history, by combining it with the recent development of attacking models. The proposed framework is constrained by the natural logic formalism and we hope more logic-based attacks will be further explored for understanding the desired property of natural language reasoning.

2 Related Work

Adversarial Attacks in NLP. White-box attacks leverage the architecture and parameters of victim models to craft adversarial examples (Liang et al., 2018; Wallace et al., 2019; Ebrahimi et al., 2018). Black-box models, however, have no such knowledge. Pioneering blind models (Jia and Liang, 2017), for example, create adversarial examples by adding distracting sentences to the input. More recently, score-based (e.g., Zhang et al. (2019); Jin et al. (2020)) and decision-based attack models (Zhao et al., 2018) also query the prediction

scores or the final decisions of victim models.

In terms of perturbation granularities, character-level attacks modify characters (Ebrahimi et al., 2018) while word-level models rely on word substitutions that can be performed based on word embeddings (Sato et al., 2018), language models (Zhang et al., 2019), or even external knowledge bases (Zang et al., 2020). Sentence-level attack models add perturbation to an entire sentence by performing paraphrasing (Iyyer et al., 2018) or attaching distracting sentences (Jia and Liang, 2017).

Kang et al. (2018) generated natural language inference examples based on entailment label composition functions with the help of lexical knowledge. Minervini and Riedel (2018) utilized a set of first-order-logic constraints to measure the degree of rule violation for natural language inference. The efforts utilized the generated examples for data augmentation. The focus is not on adversarial attack and the adversarial examples' quality, e.g., the attack validity, is not evaluated.

Natural Logic. Natural logic has a long history and has been closely developed with natural language inference (Lakoff, 1970; Van Benthem, 1995; MacCartney, 2009; Icard, 2012; Angeli et al., 2016; Hu and Moss, 2018; Chen et al., 2021). Recently, some efforts have started to consider monotonicity in attacks, including creating test sets to understand NLI models' behaviour (Richardson et al., 2020; Yanaka et al., 2019a,b, 2020; Geiger et al., 2020). The existing work, however, has not performed systematic attacks based on natural logic. The core idea of monotonicity (e.g., downward monotone) and projection has not been systematically considered. The models have not been combined with the state-of-the-art adversarial attack framework and search strategies for the general purpose of adversarial attacks. For example, Richardson et al. (2020) and Yanaka et al. (2020) generate adversarial examples from a small vocabulary and pre-designed sentence structures. The effort of Yanaka et al. (2019b) is limited by only considering one-edit distance between a premise and hypothesis. We aim to explore principled approaches to constructing perturbations based on natural logic, and the control of the quality of attack generation can leverage the continuing advancement of language models. The proposed attack settings, along with the breakdown of attack categories, help reveal the properties of victim models in both label-preserving and label-flipping attacks.

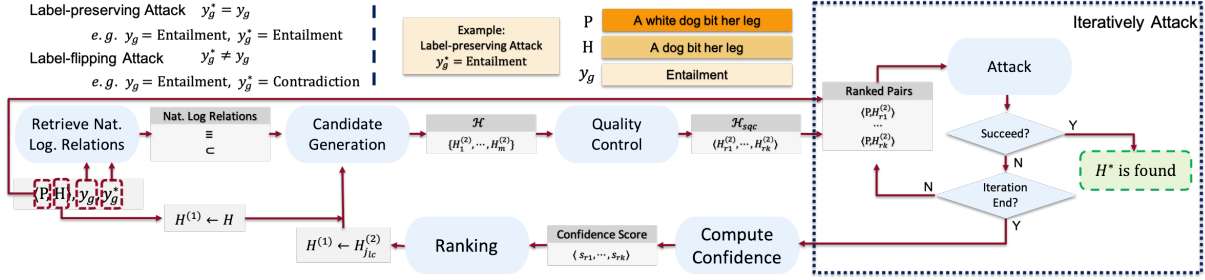


Figure 1: Overview of NatLogAttack generation and attacking process.

3 NatLogAttack: A Natural-logic-based Attack Framework

This section introduces NatLogAttack, a systematic adversarial attack framework centring around natural logic. The overview of NatLogAttack’s generation and attack process is depicted in Figure 1. Below we will introduce the background, attack principles, setups, and each component of the framework.

3.1 Background

The study of natural logic can be traced back to Aristotle’s syllogisms. Rather than performing deduction over an abstract logical form, natural logic models inference in natural language by operating on the structure or surface form of language (Lakoff, 1970; van Benthem, 1988; Valencia, 1991; Van Benthem, 1995; Nairn et al., 2006; MacCartney, 2009; MacCartney and Manning, 2009; Icard, 2012; Angeli and Manning, 2014; Hu and Moss, 2018; Chen and Gao, 2021; Chen et al., 2021). It allows for a wide range of intuitive inferences in a conceptually clean way that we use daily and provides a good framework for attacking inference models—we doubt that a victim model vulnerable to such natural attacks indeed performs reliable reasoning. Our work uses the natural logic variant proposed by MacCartney and Manning (2009) and MacCartney (2009), which extends the prior formalism to model the entailment relations between two spans of texts with seven relations $\mathfrak{B} = \{\equiv, \sqsubset, \supset, \wedge, \mid, \sim, \#\}$, representing *equivalence*, *forward entailment*, *reverse entailment*, *negation*, *alternation*, *cover*, and *independence*, respectively. Through projection based on *monotonicity* in context, local lexical-level entailment relations between a premise and hypothesis can be aggregated to determine the entailment relations at the sentence-pair level. For completeness of this paper, we highlight the key building blocks in Appendix A.

Setups	Label $y_g \rightarrow y_g^*$	Strategy	Nat. Logic Relations
Label-preserving	$E \rightarrow E$	$H \models H^*$	$H \equiv H^*$ or $H \sqsubset H^*$
	$C \rightarrow C$	$H^* \models H$	$H \equiv H^*$ or $H \supset H^*$
	$N \rightarrow N$	$H^* \models H$	$H \equiv H^*$ or $H \sqsupset H^*$
Label-flipping	$E \rightarrow C$	$H \models \neg H^*$	$H \wedge H^*$ or $H \mid H^*$
	$E \rightarrow N$	$H \not\models H^*$ and $H \not\models \neg H^*$	$H \sqsubset H^*$ or $H \sim H^*$
	$C \rightarrow E$	$\neg H^* \models H$	$H \equiv \neg H^*$ or $H \sqsupset H^*$

Table 1: Generation principles of NatLogAttack and natural logic relations between the original hypothesis H and the generated hypothesis H^* , where E, C and N stand for *entailment*, *contradiction* and *neutral*.

3.2 NatLogAttack Setups and Principles

Formally, given a premise sentence P , its n -word hypothesis $H = (h_1, h_2, \dots, h_n)$, and the ground-truth natural language inference label $y_g = \mathbb{L}(P, H)$, NatLogAttack generates a hypothesis H^* that satisfies a desired target label $y_g^* = \mathbb{L}(P, H^*)$. The attacking pair $\langle P, H^* \rangle$ is generated only if the original pair $\langle P, H \rangle$ is correctly classified by a victim model \mathbb{F} . Accordingly, we denote $y = \mathbb{F}(P, H)$ as the natural language inference label predicated by the victim model \mathbb{F} for the original pair and denote $y^* = \mathbb{F}(P, H^*)$ as the predicted label for the attacking pair.

We propose to perform the attacks in two setups: the *label-preserving* and *label-flipping* attacks. The attack principles and setups are summarized in Table 1. A *label-preserving* attack generates adversarial examples with $y_g^* = y_g$, aiming to test the robustness of victim models on different inputs that have the same label—it attacks victim models under perturbations that do not change the inferential labels of the original premise-hypothesis pair.

The *label-flipping attacks*, on the other hand, aim at attacking victim models with perturbations that are key to differentiating two different logical relations where $y_g^* \neq y_g$. Note that natural logic can be naturally used to generate label-flipping attacks, and our work here is among the first to explore this type of attacks for natural language understanding, although label-flipping attacks have been explored in image attacks (Tramèr et al., 2020).

The third column of the table (*strategy*) lists the logic conditions between the generated hypothesis H^* and the original hypothesis H that satisfy the desired properties of preserving or flipping labels to obtain the target label y_g^* . Consider the second row of the label-preserving setup (*i.e.*, $C \rightarrow C$), in which NatLogAttack generates a hypothesis H^* with $y_g^* = y_g = \text{contradiction}$. This is achieved by ensuring the natural language inference label between H^* and H to obey *entailment*: $H^* \models H$.² This guarantees the sentence pair $\langle P, H^* \rangle$ to have a *contradiction* relation. In the natural logic formalism (MacCartney, 2009), this is implemented with $H \equiv H^*$ or $H \sqsupset H^*$. Consider another example. In the last row of the *label-flipping* setup, NatLogAttack generates a new hypothesis H^* with $y_g^* = \text{entailment}$ from a *contradiction* pair, implemented by following the natural logic relations $H \equiv \neg H^*$ or $H \sqsupset \neg H^*$.

Constraint 3.1 We constrain NatLogAttack from generating neutral attack examples ($y_g^* = \text{neutral}$) using the premise-hypothesis pairs with $y_g = \text{contradiction}$, because two contradictory sentences may refer to irrelevant events from which a neutral pair cannot be reliably generated.³

Constraint 3.2 NatLogAttack is also constrained from generating contradiction and entailment attacks ($y_g^* = \text{contradiction}$ or $y_g^* = \text{entailment}$) from neutral pairs ($y_g = \text{neutral}$), as there are many ways two sentences being neutral, including reverse entailment and diverse semantic relations. The contradiction and entailment pairs cannot be reliably generated.

3.3 Generation and Quality Control

3.3.1 Preparing Natural Logic Relations

As shown in the bottom-left part of Figure 1, given a premise-hypothesis pair $\langle P, H \rangle$, the ground-truth label y_g , and the target label y_g^* , NatLogAttack retrieves natural logic relations from the last column of Table 1. Consider *label-preserving* attacks and take $y_g^* = y_g = \text{entailment}$ as an example. From the last column in the first row of the *label-preserving* setup, NatLogAttack finds and pushes the relations \equiv and \sqsupset into the *natural-logic relations set*, $\mathfrak{R}_g^* = \{\equiv, \sqsupset\}$, where \mathfrak{R}_g^* includes the natural-logic

²We use the *entailment* notation that is same as in (MacCartney and Manning, 2009).

³For example, The SNLI (Bowman et al., 2015) and MNLI datasets (Williams et al., 2018) were annotated under a guideline with a specific assumption of treating potentially irrelevant events as *contraction*.

relations between H and H^* and will be used to generate the latter. Note that $r_g^* \in \mathfrak{R}_g^*$ is one of relations in \mathfrak{R}_g^* .

We first copy H to $H^{(1)}$, denoted as $H^{(1)} \leftarrow H$ for the convenience of notation, because the generation-and-attack process may be performed multiple rounds if one round of attacks fail. Then we use the notation $H^{(1)}$ and $H^{(2)}$ to refer to the original and a generated hypothesis sentence in each round. Note that in the above example, as will be discussed below, within each round of generation, NatLogAttack will provide a set of attacks to perform multiple (iterative) attacks.

3.3.2 Candidate Generation

Algorithm 1: Candidate Generation

Input: Sentence $H^{(1)}$ with tokens $(h_1^{(1)}, \dots, h_n^{(1)})$, target natural-logic relation set \mathfrak{R}_g^*
Output: Candidate sentence set \mathcal{H}

- 1 **Init** $\mathcal{H} = \emptyset$
- 2 $\mathfrak{L} = \text{natlog}(H^{(1)})$
- 3 **foreach** $h_i^{(1)} \in H^{(1)}$ and $r_g^* \in \mathfrak{R}_g^*$ **do**
- 4 $\mathfrak{R}_{local}^* = \mathfrak{L}_{\mathfrak{B}}[\text{id}x^{\mathfrak{L}_i}(r_g^*)]$
- 5 **if** $\equiv \in \mathfrak{R}_{local}^*$ **then**
- 6 $\mathcal{H} = \mathcal{H} \cup \text{PerturbSyno}(H^{(1)}, h_i^{(1)})$
- 7 $\mathcal{H} = \mathcal{H} \cup \text{DoubleNegation}(H^{(1)})$
- 8 **end**
- 9 **if** $\sqsupset \in \mathfrak{R}_{local}^*$ **then**
- 10 $\mathcal{H} = \mathcal{H} \cup \text{PerturbHyper}(H^{(1)}, h_i^{(1)})$
- 11 $\mathcal{H} = \mathcal{H} \cup \text{Deletion}(H^{(1)}, i)$
- 12 **end**
- 13 **if** $\sqsupset \in \mathfrak{R}_{local}^*$ **then**
- 14 $\mathcal{H} = \mathcal{H} \cup \text{PerturbHypo}(H^{(1)}, h_i^{(1)})$
- 15 $\mathcal{H} = \mathcal{H} \cup \text{Insertion}(H^{(1)}, i)$
- 16 **end**
- 17 **if** $\sqsupset \in \mathfrak{R}_{local}^*$ **then**
- 18 $\mathcal{H} = \mathcal{H} \cup \text{PerturbCoHyper}(H^{(1)}, h_i^{(1)})$
- 19 $\mathcal{H} = \mathcal{H} \cup \text{PerturbAnto}(H^{(1)}, h_i^{(1)})$
- 20 $\mathcal{H} = \mathcal{H} \cup \text{AltLM}(H^{(1)}, i)$
- 21 **end**
- 22 **if** $\wedge \in \mathfrak{R}_{local}^*$ **then**
- 23 $\mathcal{H} = \mathcal{H} \cup \text{AddNeg}(H^{(1)}, h_i^{(1)})$
- 24 **end**
- 25 **end**

Return: \mathcal{H}

Our candidate attack generation process is described in Algorithm 1. Taking $H^{(1)}$ and \mathfrak{R}_g^* as the input, the algorithm aims to generate a set of candidate hypotheses $\mathcal{H} = \{H_1^{(2)}, \dots, H_m^{(2)}\}$ with each pair $\langle H^{(1)}, H_i^{(2)} \rangle$ following a target relation $r_g^* \in \mathfrak{R}_g^*$ where $H_i^{(2)} \in \mathcal{H}$. For each token $h_i^{(1)} \in H^{(1)}$ and $r_g^* \in \mathfrak{R}_g^*$, the algorithm obtains the monotonicity and relation projection infor-

mation using the Stanford *natlog* parser⁴ (line 2). Specifically for $h_i^{(1)}$, suppose the parser outputs an ordered relation list: $\mathcal{L}_i = \langle \equiv, \sqsupset, \sqsubset, \wedge, |, \smile, \# \rangle$, this returned list actually encodes the contextualized projection information, which we leverage to substitute $h_i^{(1)}$ with h'_i to generate $H_i^{(2)}$ that satisfies relation r_g^* .

In natural logic, when determining the sentence-level logical relation between a premise and hypothesis sentence, *projection* is used to map local lexicon-level logical relation to sentence-level relations by considering the context and monotonicity. However, in adversarial attacks, NatLogAttack needs to take the following reverse action:

$$\mathfrak{R}_{local} = \mathcal{L}_{\mathfrak{B}}[idx^{\mathcal{L}_i}(r_g^*)] \quad (1)$$

where r_g^* is the target sentence-level natural logic relation (in our above example, suppose $r_g^* = \text{'}\sqsubset\text{'}$). Then $idx^{\mathcal{L}_i}(\cdot)$ returns the index of that relation in \mathcal{L}_i . For $\text{'}\sqsubset\text{'}$, the index is 3. Then the index is used to find the lexicon-level (local) relation from the predefined ordered list $\mathcal{L}_{\mathfrak{B}} = \langle \equiv, \sqsupset, \sqsubset, \wedge, |, \smile, \# \rangle$. In the above example we will get $\mathcal{L}_{\mathfrak{B}}[3] = \text{'}\sqsubset\text{'}$. Again, Equation 1 presents a reverse process of the regular *projection* process in natural logic. In other words, the ordered relation list provided by the *natlog* parser for each word token, when used together with the predefined (ordered) relation list $\mathcal{L}_{\mathfrak{B}}$, specifies a mapping between global (sentence-level) natural-logic relations and local (lexicon-level) relations. Note also that the output \mathfrak{R}_{local} is a set, because \mathcal{L}_i is an ordered list that may contain the same relation multiple times.

Basic Word Perturbation. For a word token h_i , we replace it with word h'_i to ensure the local relation $\langle h_i, h'_i \rangle$ to be $r_{local} \in \mathfrak{R}_{local}$. NatLogAttack extracts natural-logic relation knowledge from knowledge bases to obtain word candidates for the desired relation types. The word perturbation of NatLogAttack focused on five relations in Table 8.

Constraint 3.3 *Since cover (\smile) is very rare and independence ($\#$) is ambiguous, NatLogAttack is constrained to only focus on utilizing the remaining five relations: $\{ \equiv, \sqsupset, \sqsubset, \wedge, | \}$.*

We attack the victim models using the most basic semantic relations explicitly expressed in knowledge bases and knowledge implicitly embedded in large pretrained language models. Specifically, we

Monotonicity	Upward	Downward
Syntax	$adj + n \sqsupset n$ $v + adv \sqsupset v$ $s + PP \sqsupset s$	$adj + n \sqsubset n$ $v + adv \sqsubset v$ $s + PP \sqsubset s$

Table 2: Insertion and deletion operations applied in the upward and downward context. s is short for *sentence*.

use WordNet (Miller, 1995) to extract the desired lexical relations. For a word token h_i , we search candidate words h'_i that has one of the following relations with h_i : $\{ \equiv, \sqsupset, \sqsubset, \wedge, | \}$. Synonyms are used as h'_i to substitute h_i for constructing $H^{(2)}$ with an *equivalence* relation to $H^{(1)}$ (line 6), hypernyms are used for *forward entailment* (line 10), and hyponyms for *reverse entailment* (line 14). Due to the transitiveness of *forward entailment* (\sqsupset) and *reverse entailment* (\sqsubset), we centre around h_i to find its hypernyms and hyponyms but restrict the distances within a threshold to avoid generating sentences that are semantically unnatural, contain overgeneralized concepts, or are semantically implausible. Later, we will further use a language model to control the quality.

For *alternation*, the perturbation candidates h'_i are words that share the common hypernym with h_i (line 18). Following MacCartney (2009), we do not use antonyms of content words for the *negation* relation but instead use them to construct *alternation* hypotheses (line 19). For the *negation* (line 23), a list of negation words and phrases is used to construct new hypotheses. Note that while our experiments show the NatLogAttack has been very effective and outperforms other attack models, some of the components can be further augmented as future work.

Enhancing Alternation. As discussed above, attacks may run multi-rounds if the prior round fails. For *alternation* substitution, NatLogAttack does not replace the word token that has been substituted before, since the *alternation* of *alternation* does not guarantee to be the *alternation* relation. In addition to constructing *alternation* hypotheses using WordNet, we further leverage DistilBert (Sanh et al., 2019) to obtain the *alternation* candidates using the function *AltLM* (line 20). Specifically, we mask the target word (which is a verb, noun, adjective or adverb) and prompt the language model to provide candidates. The provided candidates and replaced words are required to have the same POS tags.

Insertion and Deletion. In addition to substitution, NatLogAttack also follows natural logic and

⁴<https://stanfordnlp.github.io/CoreNLP/natlog.html>.

monotonicity to construct examples using the insertion and deletion operations. As shown in Table 2, adjectives, adverbs and prepositional phrases are leveraged in the upward and downward context of monotonicity to enhance the attacks for entailment (\sqsubset) and reverse entailment (\sqsupset). We include the details in Appendix B, which is built on Stanford *CoreNLP* parser and pretrained language models. Note that the syntactic rules do not guarantee to generate sentences with the desired NLI labels (e.g., see (Partee, 1995) for the discussion on the semantic composition of *adjective + noun*) and the process is only for generating candidates. We will use the pretrained language model to further identify good adversarial examples at a later stage. Both the insertion and deletion operations are used with monotonicity and projection context to generate different relations.

3.3.3 Attack Quality Control

NatLogAttack uses DistilBert (Sanh et al., 2019) to calculate the pseudo-perplexity scores (Salazar et al., 2020) for all generated hypotheses $\mathcal{H} = \{H_1^{(2)}, H_2^{(2)}, \dots, H_m^{(2)}\}$, and keeps only a maximum of 100 candidates with the lowest perplexity values. In our development, we found that the quality control stage is important for ensuring the quality of attack examples, particularly for reducing word perturbation mistakes resulting from incorrect interpretation of the words being substituted, which often results in unnatural hypothesis sentences, as well as reducing other sources of low-quality attacks including over-generalization of concepts and implausible semantics caused by insertion and deletion. The output of this stage is an ordered list of candidate attacks $\mathcal{H}_{sqc} = \langle H_{r_1}^{(2)}, H_{r_2}^{(2)}, \dots, H_{r_k}^{(2)} \rangle$.

3.4 Iterative and Multi-rounds Attacking

As discussed above, NatLogAttack performs iterative attacking within each round of generation and then multi-round attacks if the current round fails. Within each round, the original premise P and each hypothesis in the ranked hypotheses list \mathcal{H}_{sqc} form an attack list $\langle \langle P, H_{r_1}^{(2)} \rangle, \dots, \langle P, H_{r_k}^{(2)} \rangle \rangle$. As shown in Figure 1, when an attack succeeds, we output the corresponding hypothesis as H^* , which is sent for evaluation. If an attack fails, the next pair in the ranked attack list will be tried until the list is exhausted. Then NatLogAttack organizes the next round of attacks. In total NatLogAttack generates a maximum of 500 attacks for each $\langle P, H \rangle$ pair.

Models	SNLI	MED	MED _{up}	MED _{down}	MNLI	SICK
BERT	89.99	77.68	74.42	81.72	84.32	87.06
RoBERTa	91.53	73.37	80.97	70.72	87.11	87.79

Table 3: Victim models’ accuracy on different datasets.

When generating the next round attacks, we identify the adversarial pair for which the victim model has the lowest confidence (indexed as j_{lc}) over the ground-truth class y_g^* :

$$j_{lc} = \arg \min_{j \in \{r_1, \dots, r_k\}} \{s_{r_1}, \dots, s_{r_k}\} \quad (2)$$

$$s_{r_j} = o(y_g^* | (P, H_{r_j}^{(2)})) \quad (3)$$

where $o(*)$ returns the corresponding softmax probabilities of the output layer. We then copy $H_{j_{lc}}^{(2)}$ to $H^{(1)}$, denoted as $H^{(1)} \leftarrow H_{j_{lc}}^{(2)}$. The attack continues until the victim model is deceived to make a wrong prediction y^* that is different from the ground truth y_g^* or the maximum number of attacks is reached.

4 Experiments and Results

4.1 Experimental Setup

Dataset Our study uses SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), MED (Yanaka et al., 2019a), HELP (Yanaka et al., 2019b), and SICK (Marelli et al., 2014; Hu et al., 2020) datasets. The MED upward and downward subsets are denoted as MED_{up} and MED_{down}, respectively. Details of the datasets and the setup for training can be found in Appendix C.

Attack and Victim Models We compared the proposed model to five representative attack models including the recent state-of-the-art models: Clare (Li et al., 2021), BertAttack (Li et al., 2020), PWWS (Ren et al., 2019), TextFooler (Jin et al., 2020) and PSO (Zang et al., 2020). Specifically, we used the implementation made publicly available in TextAttack.⁵ For victim models, we used uncased BERT (Devlin et al., 2019) and RoBERTa base models (Liu et al., 2019). The accuracy of victim models is included in Table 3, which is comparable to the state-of-the-art performance.

Evaluation Metrics Three metrics are used to evaluate the models from different perspectives. The sign \uparrow (\downarrow) indicates that the higher (lower) the values are, the better the performance is.

⁵<https://github.com/QData/TextAttack>

Victim Model	Attack Model	SNLI			MED			MED _{up}			MED _{down}			MNLI			SICK		
		HVASR	QN	PPL	HVASR	QN	PPL	HVASR	QN	PPL	HVASR	QN	PPL	HVASR	QN	PPL	HVASR	QN	PPL
BERT	PWWS	29.9	175.8	15.96	45.9	115.3	18.18	43.1	119.1	17.98	48.3	111.6	18.38	27.8	184.2	13.87	31.0	147.1	17.75
	Textfooler	<u>34.5</u>	<u>58.4</u>	<u>15.88</u>	<u>47.3</u>	<u>51.2</u>	<u>17.96</u>	<u>47.8</u>	<u>51.2</u>	<u>17.77</u>	46.9	51.2	<u>18.15</u>	37.3	<u>74.7</u>	<u>13.62</u>	30.7	<u>50.0</u>	<u>17.62</u>
	PSO	20.5	91.8	16.06	38.8	81.9	18.19	37.7	83.9	18.14	39.7	79.7	18.25	32.0	103.4	13.81	22.3	115.86	17.77
	BertAttack	31.6	76.4	17.07	39.9	62.3	18.86	31.1	63.2	18.7	47.4	61.5	19.02	<u>37.4</u>	86.5	14.47	<u>32.2</u>	91.7	18.18
	Clare	19.9	328.3	16.87	36.7	199.7	18.31	29.9	205.5	18.30	42.8	194.8	18.33	25.2	299.8	16.87	23.1	246.9	18.60
	NatLogAtt*	35.7	42.8	14.78	56.9	42.7	17.43	57.9	30.1	17.24	56.0	<u>55.4</u>	17.62	39.7	50.1	13.47	43.6	40.3	16.73
RoBERTa	PWWS	<u>35.5</u>	177.1	16.05	39.8	118.5	18.15	41.3	121.1	18.30	38.7	115.8	18.00	28.7	189.6	13.83	35.2	143.4	17.91
	Textfooler	30.0	<u>59.7</u>	<u>15.93</u>	42.6	<u>50.2</u>	<u>18.06</u>	38.7	<u>49.5</u>	<u>17.98</u>	45.6	<u>50.82</u>	<u>18.13</u>	34.0	<u>78.2</u>	<u>13.61</u>	33.8	<u>49.6</u>	<u>17.69</u>
	PSO	19.2	92.9	16.17	34.3	81.8	18.14	27.1	83.2	18.03	39.3	80.19	18.26	28.3	99.4	13.85	24.9	115.0	17.75
	BertAttack	34.9	78.3	16.89	<u>47.3</u>	61.1	18.77	<u>47.2</u>	59.7	18.66	<u>47.4</u>	62.4	18.89	<u>39.2</u>	91.2	14.65	<u>35.6</u>	95.8	18.21
	Clare	14.7	326.6	16.65	27.4	199.8	18.54	17.9	203.7	18.20	35.2	195.9	18.88	22.6	296.7	16.44	27.5	244.0	18.16
	NatLogAtt*	36.5	45.0	14.69	55.5	33.9	17.37	59.7	27.5	17.34	52.3	40.2	17.40	39.7	46.1	13.53	49.3	42.9	16.61

Table 4: Performance of different attack models in label-preserving attacks. The bold font marks the best performance under each evaluation setup. The improvements of NatLogAtt over the second-best results (marked with underscores) are statistically significant ($p < 0.05$) under one-tailed paired t-test.

- **Human Validated Attack Success Rate (HVASR \uparrow).** Most existing attacking methods are evaluated with attack success rates that are not validated by human subjects, assuming that the attacking methods could generate adversarial examples of the desired labels. This assumption works for many NLP tasks such as sentiment analysis and text classification. However, this is not the case in NLI, since the logical relationships can be easily broken during the generation process. As observed in our experiments, although the state-of-art attacking models (BertAttack and Clare) attain high attack success rates on various NLP tasks, human-validated evaluation demonstrates that they are much less effective in attacking natural language reasoning. To reliably evaluate the attack performance, we use *Human Validated Attack Success Rate* (HVASR). Specifically, we used Amazon Mechanical Turk⁶ to validate if the generated attack examples belong to the desired relations. Each example was annotated by at least three workers and the label is determined by the majority voting. HVASR is the percentage of *successful-and-valid* adversarial examples that successfully deceived the victim models to make the wrong prediction and at the same time the majority of the annotators think their NLI labels are the desired target labels y_g^* . While HVASR is our major evaluation metric, we also use query numbers and perplexity to provide additional perspectives for observations.
- **Query number (QN \downarrow)** refers to the average number of times that a successful attack needs to query the victim model (Zang et al., 2020; Li et al., 2020). QN can reflect the efficiency (but

not effectiveness) of an attack model.

- **Perplexity (PPL \downarrow)** reflects the fluency and quality of generated examples. Same as in (Zang et al., 2020; Li et al., 2021), it is computed with GPT-2 (Radford et al., 2019) during evaluation.

4.2 Results and Analysis

Results on Label Preserving Attacks Table 4 shows the performance of different models on *label-preserving attacks*. We can see that NatLogAttack consistently achieves the best performance on HVASR. The detailed results on MED also show that NatLogAttack has a better ability to construct adversarial examples in both upward and downward monotone. NatLogAttack also shows superior performance on average QN and PPL in nearly all setups.

We can see that NatLogAttack has a large HVASR and small QN value in MED_{up}, suggesting that NatLogAttack can easily generate attacks in the upward monotone. However, in MED_{down}, NatLogAttack needs more efforts (QN). Our further analysis reveals that this is because in the downward monotone, the attack model relies more on the insertion operation than deletion, and the former is more likely to result in unsuccessful attempts.

Figure 2 further compares the query numbers (QNs) of different attack models on BERT and RoBERTa in terms of the medians (instead of means) and density of QN. We can see that the majority of query numbers of NatLogAttack are rather small and medians are less than 12 for on both SNLI and MED, showing that NatLogAttack could attack successfully with very limited attempts in most cases. For each attack model, the density of QN on

⁶<https://www.mturk.com/>

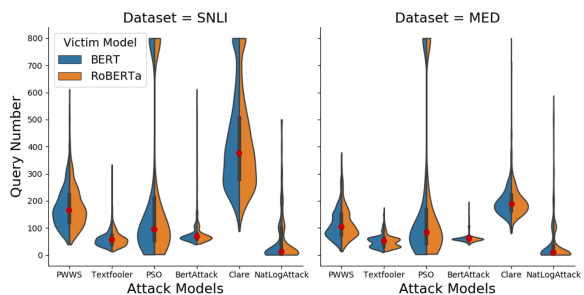


Figure 2: Query numbers (QNs) of attack models. Red dots are the medians of QNs of different attack models. The blue and orange shapes show the densities of query numbers for BERT and RoBERTa, respectively.

Vict. Md.	Lab. Flip.	SNLI			MED			MNLI			SICK		
		HVASR	QN	PPL	HVASR	QN	PPL	HVASR	QN	PPL	HVASR	QN	PPL
BERT	E→C	37.9	1.0	14.8	48.7	1.0	16.9	33.2	1.4	13.5	31.8	10.4	16.2
	E→N	57.5	2.9	14.9	50.9	2.8	17.7	50.3	4.7	13.7	55.8	6.5	16.1
	C→E	33.4	1.0	14.4	-	-	-	34.2	1.1	13.0	37.1	1.0	16.0
RoBERTa	E→C	43.5	1.4	14.6	49.8	2.9	16.7	36.8	5.0	13.5	32.1	13.9	16.4
	E→N	56.8	2.6	14.8	52.1	3.0	17.6	50.7	4.8	13.8	57.4	4.4	16.1
	C→E	36.4	1.8	14.5	-	-	-	35.1	1.2	13.0	37.7	1.0	16.0

Table 5: The evaluation for label-flipping attacks.

BERT and RoBERTa is close to each other and the medians are indiscernible and are represented by the same red dot in the figure.

Results on Label Flipping Attacks Table 5 shows the performance of NatLogAttack on the *label-flipping attacks*. Note that there has been little prior work providing systematic label-flipping attacks for NLP tasks. This new angle of evaluation is more easily implemented with logic-based attacks and provides additional insights. Specifically, the table shows that the numbers of queries that NatLogAttack sent to the victim models are much smaller than those in the *label-preserving* setting presented in Table 4, suggesting that the victim models are more vulnerable in *label-flipping setting*. For example, we can see that most of the query numbers are within 1-5 in Table 5. The pre-trained victim models are capable of memorizing the superficial features related to the original label and have difficulty in capturing the logical relationship when we alter them between sentences by keeping the majority of words untouched.

In both the *label-preserving* and *label-flipping* setup, the HVASR may still be further improved, although the proposed models have substantially outperformed the off-the-shelf state-of-the-art attack models and cautions have been exercised in all attack generation steps, which leaves room for

more research on improving logic-based attacks as future work.

Examples and Analysis. Table 6 provides the generated attack examples in the *label-preserving* setup ($E \rightarrow E$), in which we can see the quality of attacks generated by NatLogAttack is clearly higher. The baseline attacking models generate adversarial examples by replacing words based on word embedding or language models, which can easily break the logic relationships. Some examples in Table 6 show that the baselines often rely on semantic *relatedness* to construct adversarial examples, which is not detailed enough for NLI and hence break the logic relations (e.g., the last BertAttack example). Also, the last example of Clare shows that the model deletes words without considering the context (downward) monotonicity, resulting in an invalid attack. Note that the baseline models modify both premises and hypotheses and NatLogAttack focuses only on modifying hypotheses—it is straightforward to copy or adapt the operations of NatLogAttack to modify premises—in many applications, it is more natural to modify the hypotheses and keep the premises (evidences) untouched.

Table 7 shows more adversarial examples generated by NatLogAttack in the *label-flipping* setup. For all the six examples, the prediction of the victim model RoBERTa remains unchanged (*i.e.*, *entailment*, *entailment* and *contradiction* for the first, middle, and last two examples, respectively), while the ground-truth labels are now *contradiction*, *neutral*, and *entailment*, respectively. The victim model had difficulty in telling the difference, which renders an angle to challenge the models’ ability of understanding and perform reasoning.

5 Conclusion

Towards developing logic-based attack models, we introduce a framework NatLogAttack, which centres around the classical natural logic formalism. The experiments with human and automatic evaluation show that the proposed framework outperforms the existing attack methods. Compared to these models, NatLogAttack generates better adversarial examples with fewer visits to the victim models. In addition to the widely used *label-preserving* attacks, NatLogAttack also provides *label-flipping* attacks. The victim models are found to be more vulnerable in this setup and NatLogAttack succeeds in deceiving them with

Attack Model	Premise	Hypothesis
PWWS	Betty lives in Berlin	Betty <u>lives</u> animation in Germany
Textfooler	Betty lives in Berlin	Betty <u>lives</u> dies in Germany
PSO	-	-
BertAttack	Betty lives in Berlin prague	Betty lives in Germany
Clare	Betty lives in Berlin Australia	Betty lives in Germany
NatLogAttack	Betty lives in Berlin	Betty lives in Germany Federal Republic of Germany
PWWS	A snow goose jackass is a water bird	A goose is a water bird
Textfooler	A snow goose is a water bird	A goose is a water bird parakeets
PSO	A snow goose is a water bird	A goose chicken is a water bird
BertAttack	A snow goose the is a water bird	A goose is a water bird
Clare	A snow goose cat is a water bird	A goose is a water bird
NatLogAttack	A snow goose is a water bird	A goose is a water bird chordate
PWWS	-	-
Textfooler	-	-
PSO	-	-
BertAttack	I can't speak German at all	I <u>can't</u> cantheisland speak German confidently and never at all
Clare	I can't speak German at all	I <u>can't</u> speak spoke German confidently at all
NatLogAttack	I can't speak German at all	I can't speak German confidently at all on trampoline
PWWS	The purple majestic alien did not throw balls	The purple alien did not throw tennis balls
Textfooler	The purple alien did not throw balls	The purple crimson alien did not throw tennis opening balls
PSO	The purple alien did not throw balls	The purple alien unicorn did not throw tennis balls
BertAttack	The purple blue alien did not throw balls	The purple alien did not throw tennis balls
Clare	The purple alien did not throw soccer balls	The purple alien did not throw balls
NatLogAttack	The purple alien did not throw balls	The purple alien did not throw tennis balls on her cellphone

Table 6: Adversarial examples generated by different attack models on MED under the *label-preserving* setup ($E \rightarrow E$). The victim model is RoBERTa. Insertion is marked in red, substitution in blue, and deletion is marked with underline. The symbol ‘-’ indicates that the attack model fails to generate examples. The top two groups of examples are upward monotone and the bottom two groups are downward monotone.

Label Flip.	Premise	Hypothesis
$E \rightarrow C$	Many aliens drank some coke He lied, without hesitation	Many aliens drank some soda alcohol He lied did not lie , without any hesitation
$E \rightarrow N$	She's wearing a nice big hat Two formally dressed, bald older women	She's wearing a nice straw hat Two bald women matriarchs
$C \rightarrow E$	A little boy is riding a yellow bicycle across a town square Two men in orange uniforms stand before a train and do some work	It is false that the boy's bike is blue It is not true that nobody is working

Table 7: Adversarial examples generated by the NatLogAttack model in the *label-flipping* setup. The victim model is RoBERTa. The red and blue colours highlight the insertion or substitution, respectively.

much smaller numbers of queries. NatLogAttack provides an approach to probing the existing and future NLI models' capacity from a key viewpoint and we hope more logic-based attacks will be further explored for understanding the desired property of reasoning.

Limitations

Our research focuses on the adversarial attack itself and provides a framework that can be potentially used in different adversarial training strategies. We limit ourselves on attacks in this work, but it would be interesting to investigate logic-based attacks in adversarial training. We will leave that as future work. The proposed attack approach is also limited by the limitations of natural logic, while the latter has been a classical logic mechanism. For example, our proposed framework has less deductive power than first-order logic. It cannot construct

attacks building on inference rules like *modus ponens*, *modus tollens*, and *disjunction elimination*. As discussed in the paper, some components of the generation and quality control process can be further enhanced.

Acknowledgements

The research is supported by the NSERC Discovery Grants and the Discovery Accelerator Supplements. We thank Bairu Hou for his contributions to an early version of the proposed model.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. *Generating natural language adversarial examples*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Process-*

- ing (EMNLP), pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.
- Gabor Angeli and Christopher D Manning. 2014. Naturali: Natural logic inference for common sense reasoning. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, Qatar.
- Gabor Angeli, Neha Nayak, and Christopher D Manning. 2016. Combining natural logic and shallow reasoning for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Berlin, Germany.
- Yonatan Belinkov, Adam Poliak, Stuart M Shieber, Benjamin Van Durme, and Alexander M Rush. 2019. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 877–891.
- 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)*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Qian Chen, Xiaodan Zhu, Zhenhua 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)*, Vancouver. ACL.
- Zeming Chen and Qiyue Gao. 2021. Monotonicity marking from universal dependency trees. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 121–131.
- Zeming Chen, Qiyue Gao, and Lawrence S Moss. 2021. Neurallog: Natural language inference with joint neural and logical reasoning. In *Proceedings of* SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 78–88.
- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Proceedings of the First international conference on Machine Learning Challenges: evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*.
- Luc De Raedt, Robin Manhaeve, Sebastijan Dumancic, Thomas Demeester, and Angelika Kimmig. 2019. Neuro-symbolic= neural+ logical+ probabilistic. In *NeSy'19@ IJCAI, the 14th International Workshop on Neural-Symbolic Learning and Reasoning*, Macao, China.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Zhendong Dong, Qiang Dong, and Changling Hao. 2010. [HowNet and its computation of meaning](#). In *Coling 2010: Demonstrations*, pages 53–56, Beijing, China. Coling 2010 Organizing Committee.
- Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. [HotFlip: White-box adversarial examples for text classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.
- Richard Evans and Edward Grefenstette. 2018. Learning explanatory rules from noisy data. In *Journal of Artificial Intelligence Research (JAIR)*, volume 61, pages 1–64.
- Yufei Feng, Xiaoyu Yang, Michael Greenspan, and Xiaodan Zhu. 2022. Neuro-symbolic natural logic with introspective revision for natural language inference. *Transactions of the Association for Computational Linguistics (TACL)*, 10:240–256.
- Yufei Feng, Zi'ou Zheng, Quan Liu, Michael Greenspan, and Xiaodan Zhu. 2020. Exploring end-to-end differentiable natural logic modeling. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pages 1172–1185.
- Artur d'Avila Garcez, Tarek R Besold, Luc De Raedt, Peter Földiák, Pascal Hitzler, Thomas Icard, Kai-Uwe Kühnberger, Luis C Lamb, Risto Miikkilainen, and Daniel L Silver. 2015. Neural-symbolic learning and reasoning: contributions and challenges. In *2015 AAAI Spring Symposium Series*.
- Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173.
- Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. [Breaking NLI systems with sentences that require simple lexical inferences](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

- Hai Hu, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S Moss, and Sandra Kuebler. 2020. Monalog: a lightweight system for natural language inference based on monotonicity.
- Hai Hu and Larry Moss. 2018. Polarity computations in flexible categorial grammar. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 124–129.
- Thomas F Icard. 2012. Inclusion and exclusion in natural language. *Studia Logica*.
- Thomas F Icard and Lawrence S Moss. 2014. Recent progress on monotonicity. In *Linguistic Issues in Language Technology*. Citeseer.
- Adrian Iftene and Alexandra Balahur-Dobrescu. 2007. *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Prague, Czech.
- Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885.
- 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)*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.
- Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.
- Dongyeop Kang, Tushar Khot, Ashish Sabharwal, and Eduard Hovy. 2018. Adventure: Adversarial training for textual entailment with knowledge-guided examples. In *The 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.
- Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC.
- George Lakoff. 1970. Linguistics and natural logic. *Synthese*, 22(1-2):151–271.
- Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. **BERT-ATTACK: Adversarial attack against BERT using BERT**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.
- Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4208–4215.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Bill MacCartney. 2009. *Natural Language Inference*. Ph.D. thesis, Stanford University.
- Bill MacCartney and Christopher D Manning. 2009. An extended model of natural logic. In *Proceedings of the 8th international conference on computational semantics (IWCS)*, Stroudsburg, United States.
- Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. 2019. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, New Orleans, USA.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, Reykjavik, Iceland.
- Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3428–3448.
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Pasquale Minervini and Sebastian Riedel. 2018. Adversarially regularising neural nli models to integrate logical background knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL)*, pages 65–74.
- John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

- Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Proceedings of the 5th international workshop on inference in computational semantics*, Buxton, England.
- Barbara Partee. 1995. Lexical semantics and compositionality. *Invitation to Cognitive Science*.
- Jonathan Pilault, Christopher Pal, et al. 2021. Conditionally adaptive multi-task learning: Improving transfer learning in nlp using fewer parameters & less data. In *International Conference on Learning Representations (ICLR)*.
- Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018. [Collecting diverse natural language inference problems for sentence representation evaluation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 67–81, Brussels, Belgium. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.
- Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1085–1097.
- Kyle Richardson, Hai Hu, Lawrence S Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, New York, USA.
- Tim Rocktäschel and Sebastian Riedel. 2017. End-to-end differentiable proving. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, Long Beach, USA.
- Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. [Masked language model scoring](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @NeurIPS*.
- Motoki Sato, Jun Suzuki, HiroYuki Shindo, and Yuji Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4323–4330.
- Florian Tramèr, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Jörn-Henrik Jacobsen. 2020. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. In *International Conference on Machine Learning*, pages 9561–9571. PMLR.
- Víctor Manuel Sánchez Valencia. 1991. *Studies on natural logic and categorial grammar*. Universiteit van Amsterdam.
- Johan van Benthem. 1988. The semantics of variety in categorial grammar. *Categorial grammar*.
- Johan Van Benthem. 1995. *Language in Action: categories, lambdas and dynamic logic*. MIT Press.
- Johan Van Benthem et al. 1986. *Essays in logical semantics*. Springer.
- Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. [Universal adversarial triggers for attacking and analyzing NLP](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.
- Leon Weber, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, and Tim Rocktäschel. 2019. Nlprolog: Reasoning with weak unification for question answering in natural language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Austin, Texas, United States.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.
- Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. 2020. [Do neural models learn systematicity of monotonicity inference in natural language?](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6105–6117.
- Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019a. Can neural networks understand monotonicity reasoning? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Austin, Texas, United States.
- Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and

- Johan Bos. 2019b. Help: A dataset for identifying shortcomings of neural models in monotonicity reasoning. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM)*, Minneapolis, Minnesota, USA.
- Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. In *Advances in Neural Information Processing Systems*, pages 2319–2328.
- Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6066–6080.
- Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. [Openattack: An open-source textual adversarial attack toolkit](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 363–371.
- Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2019. [Generating fluent adversarial examples for natural languages](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5564–5569, Florence, Italy. Association for Computational Linguistics.
- Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware bert for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9628–9635.
- Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

A Background

Our work is based on the specific natural logic formalism proposed by MacCartney and Manning (2009) and MacCartney (2009). To model the entailment relations between two spans of texts, MacCartney and Manning (2009) introduced seven relations inspired by the set theory: $\mathfrak{B} = \{ \equiv, \sqsubset, \sqsupset, \wedge, |, \smile, \# \}$ (see Table 8 for some examples). The inference of natural logic is built on monotonicity, which is a pervasive feature of natural language that explains the impact of semantic composition on entailment relations (Van Benthem et al., 1986; Valencia, 1991; Icard and Moss, 2014). Suppose *dog* \sqsubset *animal*, the upward monotone context keeps the entailment relation when the argument “increases”

(e.g., *dog* \sqsubset *animal*). Downward monotone keeps the entailment relation when the argument “decreases” (e.g., in *all animals* \sqsubset *all dogs*). The system performs monotonicity inference through a projection $\rho: \mathfrak{B} \rightarrow \mathfrak{B}$, which is determined by the context and projection rules. As will be detailed, a monotonicity-based parser can provide monotonicity information for each word token in a sentence and the projection information. For example, consider the sentence *All* \uparrow *the* \downarrow *kids* \downarrow *run* \uparrow , where \uparrow denoted upward polarity and \downarrow downward polarity. If we mutate the word *kids* with *boys*, where *kids* \sqsupset *boys*, the system projects the *reverse entailment* (\sqsupset) into *forward entailment* (\sqsubset) due to its downward polarity, i.e., $\rho(\sqsupset) = \sqsubset$, and thus *All the kids run* \sqsubset *All the boys run*.

With these components ready, the system aggregates the projected local relations to obtain the inferential relation between a premise and hypothesis sentence. Specifically, Table 9 (MacCartney, 2009; MacCartney and Manning, 2009; Angeli and Manning, 2014) shows the composition function when a relation in the first column is joined with a relation listed in the first row, yielding the relations in the corresponding table cell. MacCartney (2009) shows that different orders of compositions yield consistent results except in some rare artificial cases. Therefore, many works, including ours, perform a sequential (left-to-right) composition. Consider two edits from the premise sentence, *All the kids run*, to the hypothesis, *All the boys sleep*. The first edit that replaces *kids* in the premise with *boys* yields *All the kids run* \sqsubset *All the boys run*. The second edit of replacing *run* with *sleep* yields *All the boys run* $|$ *All the boys sleep*. Based on Table 9, the union of the relations resulted from these two edits (i.e., $\sqsubset \bowtie |$) is $|$, where \bowtie is the union operator. As a result, we obtain *All the kids run* $|$ *All the boys sleep*.

The seven natural logic relations at the sentence-pair level can then be mapped to the typical three-way NLI labels (*entailment*, *contradiction*, and *neutral*), where the \equiv or \sqsubset relation can be mapped to *entailment*; the \wedge or $|$ relation to *contradiction*; the \sqsupset , \smile , and $\#$ to *neutral*.

B Insertion and Deletion

For both insertion and deletion, the part-of-speech (POS) tags and constituency parse tree for $H^{(1)}$ are first obtained using Stanford *CoreNLP*

Relation	Relation Name	Example	Set Theoretic Definition
$x \equiv y$	equivalence	<i>mommy</i> \equiv <i>mother</i>	$x = y$
$x \sqsubset y$	forward entailment	<i>bird</i> \sqsubset <i>animal</i>	$x \subset y$
$x \sqsupset y$	reverse entailment	<i>animal</i> \sqsupset <i>bird</i>	$x \supset y$
$x \wedge y$	negation	<i>human</i> \wedge <i>nonhuman</i>	$x \cap y = \emptyset \wedge x \cup y = U$
$x \mid y$	alternation	<i>bird</i> \mid <i>dog</i>	$x \cap y = \emptyset \wedge x \cup y \neq U$
$x \smile y$	cover	<i>animal</i> \smile <i>nonhuman</i>	$x \cap y \neq \emptyset \wedge x \cup y = U$
$x \# y$	independence	<i>red</i> $\#$ <i>weak</i>	all other cases

Table 8: Seven natural logic relations proposed by MacCartney and Manning (2009).

\boxtimes	\equiv	\sqsubset	\sqsupset	\wedge	\mid	\smile	$\#$
\equiv	\equiv	\sqsubset	\sqsupset	\wedge	\mid	\smile	$\#$
\sqsubset	\sqsubset	\sqsubset	$\#$	\mid	\mid	$\#$	$\#$
\sqsupset	\sqsupset	$\#$	\sqsupset	\smile	$\#$	\smile	$\#$
\wedge	\wedge	\smile	\mid	\equiv	\sqsubset	\sqsubset	$\#$
\mid	\mid	$\#$	\mid	\sqsubset	$\#$	\sqsubset	$\#$
\smile	\smile	\smile	$\#$	\sqsupset	\sqsupset	$\#$	$\#$
$\#$	$\#$	$\#$	$\#$	$\#$	$\#$	$\#$	$\#$

Table 9: Relation union table (Icard, 2012).

parser⁷, which are then used with a state-of-the-art pretrained model to perform insertion. To insert an *adjective* before a *noun* or an *adverb* after a *verb*, NatLogAttack leverages DistilBert (Sanh et al., 2019) to obtain the candidates in the corresponding locations. The syntactic rules do not guarantee to generate sentences with the desired NLI labels (e.g., see (Partee, 1995) for discussion on the semantic composition of *adjective* + *noun*). The above process is only for generating candidates, and we will use pretrained language models to find good adversarial examples.

In order to insert a prepositional phrase (PP), we first collected from the SNLI training dataset all the PPs that are the constituents of other noun phrases (NPs) for more than 100 times. We also collected PPs that appear in other verb phrases (VPs) at least 100 times. During insertion, these PPs will be added as modifiers to a noun or a verb, respectively. We also insert assertion phrases such as "It is not true that" to deceive the victim models. For the *deletion* operation, we delete the corresponding constituents based on the parse tree and POS tags.

C Details of Datasets and Baselines

As discussed in Section 4.1, our study uses SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), MED (Yanaka et al., 2019a), HELP (Yanaka et al., 2019b), and SICK (Marelli et al., 2014; Hu et al., 2020) to evaluate the models. SNLI and MNLI are widely-used general-purpose NLI datasets. Following Li et al. (2021), for MNLI, we evaluate the performance on the *matched* set. MED and HELP are designed for monotonicity-based reasoning and hence suit for probing models' capacity in natural logic-related behaviour. SICK is rich in lexical, syntactic and semantic phenomena designed for distributional semantic models including those recognizing textual entailment. For SICK,

⁷<https://stanfordnlp.github.io>

we use the corrected labels proposed by Hu et al. (2020). The pretrained victim models tested on the SNLI, MNLI, and SICK test set were finetuned on their own training set and the performances are comparable to the state-of-the-art performances as well as those used in the previous attack models. Following Yanaka et al. (2019a), the models tested on MED are finetuned on both the SNLI training set and the entire HELP dataset. Since HELP is not manually annotated, we do not use it as the test set. The MED upward subset is denoted as MED_{up} and downward subset as MED_{down}. Following (Alzantot et al., 2018; Zang et al., 2020), each test set has 1,000 sentence pairs. Also following Zeng et al. (2021), we set the maximum query number to be 500.

For all the attack models in comparison, we used the implementation made available by Morris et al. (2020). Details of these attack models are as follows.

- **PWWS** (Ren et al., 2019) makes use of the synonyms in WordNet (Miller, 1995) for word substitutions and designs a greedy search algorithm based on the probability-weighted word saliency to generate adversarial samples.
- **TextFooler** (Jin et al., 2020) utilizes counterfitting word embeddings to obtain synonyms and then performs substitution based on that.
- **PSO** (Zang et al., 2020) utilizes the knowledge base HowNet (Dong et al., 2010) to generate word substitutions. It adopts particle swarm optimization, another popular metaheuristic population-based search algorithm, as its search strategy.
- **BertAttack** (Li et al., 2020) leverages the superior performance of pretrained language model and greedily replaces tokens with the predictions from BERT.

- **Clare** (Li et al., 2021) adds two more types of perturbations, *insert* and *merge*, building on BertAttack. Since Clare has a very high query number to the victim models, we reduce the number of each type of perturbation to 10 in order to make sure that Clare can attack the victim model successfully within the maximum query number in most cases.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
They are in the Limitation Section
- A2. Did you discuss any potential risks of your work?
We do not envision there are potential risks.
- A3. Do the abstract and introduction summarize the paper’s main claims?
The abstract and introduction summarize the main claims presented in section 3-5.
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

Section 3 and 4.

- B1. Did you cite the creators of artifacts you used?
Section 3 and 4
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We used the mostly widely use NLP tools.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not relevant.
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
we do not think the data we used contain such information.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
not relevant
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C Did you run computational experiments?

section 4

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
we did not collect the information.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
section 4
- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4: we report statistical significance test.
- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
section 4
- D** **Did you use human annotators (e.g., crowdworkers) or research with human participants?**
section 4
- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.
- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
we follow MTurk standards.
- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
we follow MTurk standards.
- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
We use MTurk and the collection is not qualified for ethics review.
- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
not relevant