Systematic Monotonicity and Consistency for Adversarial Natural Language Inference

Brahmani Nutakki(✉), Akshay Badola, and Vineet Padmanabhan

School of Computer and Information Sciences, University of Hyderabad,
Hyderabad, India
brahmani3110@gmail.com, {badola,vineetnair}@uohyd.ac.in

Abstract. Natural Language Inference is a fundamental task required for understanding natural language. With the introduction of large Natural Language Inference (NLI) benchmark datasets such as SNLI and MultiNLI, NLI has seen an uptake in models achieving near-human accuracy. Deeper analyses through adversarial methods performed on these models however have cast doubts on their ability to actually understand the inference process. In this work, we attempt to define a principled way to generate adversarial attacks based on monotonic reasoning and consistency to examine their language understanding abilities. We show that the language models trained for general tasks have a poor understanding of monotonic reasoning. For this purpose, we provide methods to generate an adversarial dataset from any NLI dataset based on monotonicity and consistency principles and conduct extensive experiments to support our hypothesis. Our adversarial datasets preserve these crucial aspects of monotonicity, consistency and semantic similarity and are still able to fool a model finetuned on SNLI 79% of the time while preserving semantic similarity to a much greater extent than previous methods.

1 Introduction

Natural Language Inference (NLI), initially known as Recognizing Textual Entailment, was introduced as a PASCAL Challenge Benchmark task (RTE-1) [17]. The task involves determining if a natural language hypothesis h can be reasonably inferred from the given premise p [15]. Owing to its use as a comparison metric to quantify the semantic inference of models, it is often used as a proxy to gauge a model's ability to understand natural language. Significant advances have been made in the field of NLI, which were further propelled by the advent of huge benchmark datasets such as the Stanford Natural Language Inference Corpus (SNLI) [2] and the Multi-Genre Natural Language Inference Corpus (MNLI) [32].

Language Models and specifically Neural Language Models based on Recurrent Neural Network (RNN) [26] and large Transformers [29] have been a paradigm shift in Natural Language Modeling and have achieved state-of-the-art results in many Natural Language tasks including NLI. However, adversarial

H. Aziz et al. (Eds.): AI 2022, LNAI 13728, pp. 353–366, 2022.
https://doi.org/10.1007/978-3-031-22695-3_25

attacks and stress tests have questioned the actual language understanding ability of these models. The NLI task is particularly amenable to logical inspection and assessment and a model's failures for a given example helps to identify its shortcomings. A very instructive example is [8] which analyzes negation and shows the Language Model's inability to understand it.

In this work, we investigate the role of semantic *monotonicity* and logical *consistency* in the NLI task and introduce a framework for lexical attacks based on them. Monotonicity in this case refers to the semantic relations between generalizations and specializations of a word and inferences which can be drawn from them. By consistency we mean rules of logic; e.g. symmetry transitivity etc. are maintained across the sentences. We transform a given $< premise, hypothesis, label > \equiv (p, h, l)$ triplet in the dataset, by substituting certain words such that the change in label l is deterministic corresponding to the *monotonicity* and *consistency* rules.

For example, consider the sentence pair <*People are marching towards the mountains*, *The people are going towards the mountains*>$\equiv < p, h >$, with the label $l =$ *entailment* or e. Replacing *marching* in p with its hypernym *walking* does not change the meaning of p or the label, as it is an upward monotone. Similarly, we can derive rules for label changes for various combinations of substitutions in both p and h which lead to a specific change in label l. We call these substitutions *two-hop* label shifts as they transform both p and h. Our approach differs from prior work which have used brute force or embeddings-based perturbations [11,19] and have focused on transforming only premises. These attacks reveal critical deficiencies in the Language Model's lexical and syntactic understanding. Although we focus on NLI datasets, the methods can be generalized to other language tasks. To the best of our knowledge, this is the first work that uses attacks based on both monotonicity and consistency rules across both the premise and hypothesis.

To sum up, our contributions are:

- We provide a general principled adversarial attack method using our novel two-hop label shift rules.
- We demonstrate the efficacy of our generated datasets on State-of-the-art NLI models, and compare them against existing adversarial text generation frameworks.
- We release the code for the experiments which can be found at https://github.com/nbrahmani/Two-hop-adversarial-attacks

The rest of this paper is organized as follows: Sect. 2 gives an overview of the existing work. Section 3 gives an overview of NLI and Adversarial NLI. Section 4 describes our methodology of the proposed attacks, and Sect. 5 is about the experiments performed and the results obtained. We follow up with discussions in Sect. 6 and conclude in Sect. 7.

2 Related Work

Adversarial methods in Neural Models have been gaining prominence with the success of Image Classification models [21]. With the growing success of Neural Language Models, methods to determine the weaknesses of these models have also gained attention [6,10]. These methods are usually classified into White-box and Black-box attacks, and the Black-box attacks can be further classified into Score-based, Decision-based, and Transfer-based attacks [16].

White-box attacks have access to the gradient information of the loss function and construct the adversarial instances based on this information. Li et al. [12] use the loss function gradient of each word to find their importance and replace the words with similar words. Ebrahimi et al. [5] attack the model by flipping a character in the sentence that maximizes the model loss. Although these attacks are successful, their methodology is cumbersome.

Black-box attacks, on the other hand, only use the model outputs to generate the adversarial instances. They do not require access to the model's gradient information and are agnostic to the model. For example, Jin et al. [11] use the model's confidence scores to create adversarial perturbations. Zhao et al. [34] use only the final predicted output of the model to generate attacks instead of the confidence scores. A different approach is taken in [30] who train a classifier to mimic the decisions of the model, after which attacks are performed on this model and are then transferred to the original model.

As useful as these attacks are, they are not systematic in nature and introduce random perturbations in the data to craft adversarial examples. While in search of a more principled manner to analyze the adversarial examples in text, research has turned to gauge the model's understanding of logic. It has been observed that language models struggle to understand logic due to its discrete nature. Traylor et al. [28] test whether the models can differentiate between logical symbols such as disjunction ($\vee$), conjunction ($\wedge$) or negation ($\neg$). They find the models largely fail on their newly generated dataset. Meanwhile, the model's ability to infer over conjuncts is probed in [24]. Tarunesh et al. [27] create a huge dataset that tests the models against 17 reasoning tasks, including logical tasks such as Boolean (sentences containing logical and ($\wedge$), or ($\vee$) and their combinations) and quantifier (sentences containing universal ($\forall$) and existential ($\exists$) operators) apart from world knowledge, causality etc.

Richardson et al. [23] and Naik et al. [20] probe the models on various semantic phenomena, including logical aspects such as negation, along with monotonicity-related aspects. Glockner et al. [6] generate perturbations by replacing one word in the premise using lexical knowledge. Similarly, Yanaka et al. [33] have proposed the MED dataset that checks the model's understanding of monotonicity. They synthesize examples based on the monotonicity inference rules using contextual grammar.

Gururangan et al. [7] showed that a simple classification model achieves 67% accuracy on SNLI and 53% on MNLI when only hypotheses are given, thus showing that the models are sensitive to annotation artifacts. Certain words such as negations and gender-neutral terms lead to false predictions by the model.

Poliak et al. [22] tested a hypothesis-only model on ten different datasets and found that the model performed better than most baselines.

Our work follows [33] and [14] in that we use monotonicity and consistency to generate an adversarial dataset from the given dataset[1]. Our approach differs in our use of *two-hop* label shift rules across the premise-hypothesis pair.

3 Adversarial NLI

We discuss NLI first and then Adversarial NLI in detail:

The standard NLI task consists of predicting a label l from a sentence pair of Premise and Hypothesis (p, h). For example, the sentence pair <*A man is riding a horse in a meadow*, *A person is outside*> has the label *entailment*. Usually we deal with only three labels, *entailment, contradiction, neutral*. For our purposes we'll focus on Neural Language Models, specifically variants of BERT [4] which have achieved state-of-the-art in many NLP tasks. These models transform the sentences into distributed representations and posit them as a classification task.

For NLI, the data is a set of ordered triplets of Premise, Hypothesis and Label: $\mathcal{D} = \{(p, h, l)\}$. The objective is to find a model $\mathcal{M}$ parameterized by weights Θ, such that it predicts the correct label l given (p, h), i.e.:

$$\mathcal{M}_{\Theta} : (P, H) \rightarrow L$$

In this case, the model here is a Neural Language Model which is learned by maximizing the likelihood of Θ over the dataset. That is, the number of predicted labels l_i over the input sentences (p_i, h_i) in the dataset.

$$\mathcal{M}_{\Theta} = \underset{\Theta}{\operatorname{argmax}}\ \mathcal{L}_{\Theta} = \underset{\Theta}{\operatorname{argmax}}\ P(l_i|p_i, h_i) \qquad \forall (p_i, h_i, l_i) \in \mathcal{D}$$

Adversarial NLI on the other hand can be considered as the process of finding a set of transformations $\mathcal{T} : (S, L) \rightarrow (S, L)$ where (S, L) is the set of all <*sentence*, *label*> pairs, such that the *trained model* fails for a given example. Formally:

$$\mathcal{M}(\mathcal{T}(p_i, h_i)) \neq l_i', \qquad (p_i, h_i, l_i) \in \mathcal{D}$$

where $\mathcal{T}(p_i, h_i)$ changes either p_i or h_i or both, and l_i is the true label corresponding to the transformation $\mathcal{T}(p_i, h_i)$.

In other words, the goal is to find a method to transform the inputs so that the model's output is not the same as the expected output.

4 Towards Systematic Adversarial NLI

As we mentioned earlier, while approaches for Adversarial NLI exist, they are not systematic in nature. Here, we describe our approach used in determining the transformation $\mathcal{T}$ for Systematic Adversarial NLI.

[1] We use both SNLI and MNLI, but in practice, it can be any NLI dataset or the methods can even be adapted for any other language dataset.

Consider a data point $(p, h, l) \in \mathcal{D}$. The transformation $\mathcal{T}$ we propose is based on *two-hop* rules. Recall from Sect. 1 that these are rules which apply to *both* the premise and hypothesis, instead of only the premise. We focus only on single word substitutions using an existing ontology. We choose Wordnet [18] for our purpose, but any other ontology can be used.

Monotonicity and Consistency Rules. Let, $E(p, h)$ denote an entailment, $C(p, h)$ a contradiction and $N(p, h)$ a neutral label for premise-hypothesis pair (p, h). For a sentence $s \in \{p, h\}$, the following rules are applicable:

1. Rules of Consistency [14]:
   - $E(p, h) \wedge E(h, z) \rightarrow E(p, z)$
   - $E(p, h) \wedge C(h, z) \rightarrow C(p, z)$
   - $N(p, h) \wedge E(h, z) \rightarrow \neg C(p, z)$
   - $N(p, h) \wedge C(h, z) \rightarrow \neg E(p, z)$
   - $C(p, h) \rightarrow C(h, p)$
2. Rules of Equivalence:
   - $s' = W_{Eq}(s) \rightarrow E(s, s') \wedge E(s', s)$

   Where W_{Eq} stands for equivalent word substitution.
3. Rules of monotonicity [33]:
   - $s' = W_{ME}(s) \rightarrow E(s, s') \wedge N(s', s)$
   - $s' = W_{MN}(s) \rightarrow N(s, s') \wedge E(s', s)$

   Where W_{ME}, W_{MN} stand for Monotonically Entailment and Neutral word substitutions, respectively.

Deriving the Label Changes. Using the aforementioned consistency, equivalence and monotonicity based rules, the corresponding changes in label (shifts) for each transformation are deterministic and can be derived. We list here only the effective shift rules for the transformations as the rest of the shift rules do not induce a label change required for an adversarial attack.

We use the following notation for describing the transformations:

- **Single Sentence Transformation**: $\mathcal{T}_M(p, h) : (p', h)$ (or (p, h')) is a transformation $\mathcal{T}$ for a premise-hypothesis pair (p, h) such that only p (or h) is changed to p' (or h') via method M.
- **Dual Sentence Transformation**: $\mathcal{T}_{M,M}$, e.g., $\mathcal{T}_{E,ME}(p, h) : (p', h')$ means that premise p is changed to p' using an equivalent substitution and hypothesis h is changed to h' using a monotonically entailed substitution.

We take $\neg C(p, h)$ and $\neg E(p, h)$ to be $N(p, h)$. Based on a given transformation $\mathcal{T}_M$, we then determine the new label l'. Table 1 lists all the label shift rules.

One issue we faced was that effecting multiple transformations can cause an exponential increase in the number of possible combinations of label changes. To mitigate that, we find the words (which we call markers) which are most representative of the meaning of the word and transform them which we describe in the next Sect. 4. We use a separate model to determine the markers.

Table 1. Table of transformations

$\mathcal{T}_E(p,h) : (p',h) \rightarrow E(p,p'), E(p',p)$
- $E(p,h) \rightarrow E(p',h)$
- $C(p,h) \rightarrow C(p',h)$

$\mathcal{T}_{ME}(p,h) : (p',h) \rightarrow E(p,p'), N(p',p)$
- $E(p,h) \rightarrow \neg C(p',h)$
- $C(p,h) \rightarrow \neg E(p',h)$

$\mathcal{T}_{MN}(p,h) : (p',h) \rightarrow N(p,p'), E(p',p)$
- $E(p,h) \rightarrow E(p',h)$
- $C(p,h) \rightarrow C(p',h)$

$\mathcal{T}_E(p,h) : (p,h') \rightarrow E(h,h'), E(h',h)$
- $E(p,h) \rightarrow E(p,h')$
- $C(p,h) \rightarrow C(p,h')$
- $N(p,h) \rightarrow \neg C(p,h')$

$\mathcal{T}_{ME}(p,h) : (p,h') \rightarrow E(h,h'), N(h',h)$
- $E(p,h) \rightarrow E(p,h')$
- $N(p,h) \rightarrow \neg C(p,h')$

$\mathcal{T}_{MN}(p,h) : (p,h') \rightarrow N(h,h'), E(h',h)$
- $C(p,h) \rightarrow C(p,h')$

$\mathcal{T}_{E,E}(p,h) : (p',h') \rightarrow E(p,p'), E(p',p), E(h,h'), E(h',h)$
- $E(p,h) \rightarrow E(p',h')$
- $C(p,h) \rightarrow C(p',h')$

$\mathcal{T}_{E,ME}(p,h) : (p',h') \rightarrow E(p,p'), E(p',p), E(h,h'), N(h',h)$
- $E(p,h) \rightarrow E(p',h')$

$\mathcal{T}_{E,MN}(p,h) : (p',h') \rightarrow E(p,p'), E(p',p), N(h,h'), E(h',h)$
- $C(p,h) \rightarrow C(p',h')$

$\mathcal{T}_{ME,E}(p,h) : (p',h') \rightarrow E(p,p'), N(p',p), E(h,h'), E(h',h)$
- $E(p,h) \rightarrow \neg C(p',h')$
- $C(p,h) \rightarrow \neg E(p',h')$

$\mathcal{T}_{ME,ME}(p,h) : (p',h') \rightarrow E(p,p'), N(p',p), E(h,h'), N(h',h)$
- $E(p,h) \rightarrow \neg C(p',h')$

$\mathcal{T}_{ME,MN}(p,h) : (p',h') \rightarrow E(p,p'), N(p',p), N(h,h'), E(h',h)$
- $C(p,h) \rightarrow \neg E(p',h')$

$\mathcal{T}_{MN,E}(p,h) : (p',h') \rightarrow N(p,p'), E(p',p), E(h,h'), E(h',h)$
- $E(p,h) \rightarrow E(p',h')$
- $C(p,h) \rightarrow C(p',h')$

$\mathcal{T}_{MN,ME}(p,h) : (p',h') \rightarrow N(p,p'), E(p',p), E(h,h'), N(h',h)$
- $E(p,h) \rightarrow E(p',h')$

$\mathcal{T}_{MN,MN}(p,h) : (p',h') \rightarrow N(p,p'), E(p',p), N(h,h'), E(h',h)$
- $C(p,h) \rightarrow C(p',h')$

Selection of the Markers and Extraction of Sense. Changing all words or a random combination of words would be too computationally intensive and not helpful in generating good adversarial examples. Therefore, based on a transformation T, we select the top 5 most similar words (markers) in the sentence S ($S \in \{P, H\}$). These are selected by comparing the cosine similarities between individual word embeddings and sentence embedding. The word and sentence embeddings are obtained using a pre-trained model.

After that, a word sense disambiguation model is used to obtain the sense of the markers to ensure that the generated examples are semantically similar to original sentences. For this, we use Wordnet sense ids [18]. These transformations and the *two-hop* rules which change only the markers form the basis of our adversarial attacks.

Other methods like TextFooler [11] replace the selected word in the hypothesis from a list of synonyms by comparing the cosine similarities of their embeddings. The attack labels of such perturbations are riddled with errors. The sense of the word can also change due to the replacements. Our attacks are performed by the *two-hop* rules governed by the word-replacement technique and the ground truth and do not suffer from these issues. We also perform sense-based replacement to ensure the sense of the perturbations remains the same.

4.1 Word-Replacement Techniques

After selecting the markers and their sense, the sentences are perturbed using the three word-replacement techniques based on the type of transformation applied. They are 1) Equivalent 2) Monotonic-entailment and 3) Monotonic-neutral. These replacements govern the selected word substitute and the corresponding label. The monotonicity of the word is obtained using a polarity annotator.

- **Equivalent** word replacement is achieved by replacing the marker with one of its synonyms. It always results in an entailment in both directions.
- **Monotonic** replacement substitutes a marker by a general phrase (hypernym) or a specific phrase (hyponym). If the word is upward monotone, replacing it with hypernym results in an inferable sentence (*entailment* label), while replacing it with hyponym results in a neutral sentence. Similarly, replacing a downward monotone word with its hyponym results in an inferable sentence, and a hypernym leads to neutral classification. Corresponding to these rules we define two-word replacement methods: **Monotonic-Entailment** and **Monotonic-Neutral**.

The replacement words obtained are then modified to match the morphology of the original word after which they are filtered based on their grammar score or acceptability score. The model is now asked to classify these transformations along with the labels. Only those input sentence pairs are used whose ground truth is the same as the predicted label; the rest are skipped. If the label predicted for the perturbation differs from the one obtained using the derived rules, the attack is successful, else unsuccessful. The complete Algorithm 1 is given below.

5 Experiments and Results

5.1 Experimental Setup

Before detailing the results of the attacks, we briefly give an overview of the different models and approaches used for individual modules mentioned in Sect. 4.

Algorithm 1. Adversarial Attack using Logical Rules

```
Input: $\mathcal{T}_M$, $p$, $h$, $l$, markers $\{m\}$
Output: Transformed tuple $(p', h', l')$
Select $p$, $h$ or both based on $\mathcal{T}_M$
Treating it as a single sentence $s$ of two clauses, select top 5 words from $s \equiv \{m\}$.
for $m_i \leftarrow \{m\}$ do
    Extract the sense and replace the marker according to method $M$ with a word
end for
Remove perturbations where grammar score varies significantly from that of $s$
Query model $\mathcal{M}$ with the perturbed sentence pair $(p', h')$ and check with expected
  label $l'$
```

Selecting Markers and Extracting Sense. For selecting top 5 markers, the embeddings for the premise-hypothesis pair are extracted using a MPNet [25] based sentence encoder which has been fine-tuned on a 1B sentence dataset. This model takes the input sentences and produces word embeddings and sentence embeddings. The top 5 similar words based on cosine similarities between the word and the given sentence embedding are chosen as essential markers. The perturbations are generated by extracting sense from ESCHER [1]. These senses are then used to mine synonyms, hypernyms and hyponyms from Wordnet.

Polarity Annotation and Grammar Score. To get the monotonicity of a marker, we need the monotonic polarity. We follow [9] for polarity annotation. The input sentences are first parsed using a CCG parser and *ccg2mono* proposed in [9] is used to polarize the words as *upward*, *downward*, or *no polarity*. We then compare the grammar scores of the original and the modified sentences with a BERT model fine-tuned on the COLA dataset [31]. The model gives a probability output of the given sentence being acceptable or not. An absolute difference greater than a threshold between the original and the perturbed sentence is ignored. We found empirically that a threshold value of 0.1 works well.

5.2 Results

Using the models mentioned above, we build our attack pipeline to generate adversarial attacks. We randomly sample 5000 sentence pairs from the train splits of the SNLI [2] and MNLI [32] datasets. We then generate perturbations for all 15 types of transformations, picking a different number of markers each time. Then using the *two-hop* label shift rules, attacks are performed on the model with these perturbations. Example perturbations can be found below:

Example 1. p: Man smokes while sitting on a parked scooter.
h: A man smokes a cigarette while sitting on his scooter.
Marker_p: Man, *Marker_h*: Man
Ground Truth: Neutral, *Predicted Label*: Neutral
Transformations:

1. $\mathcal{T}_E(p, h) : (p, h')$: No perturbations as no valid perturbation exists.
2. $\mathcal{T}_{ME}(p, h) : (p, h')$:
   - H': an *adult* smokes a cigarette while sitting on his scooter.
     Label: Neutral, *Attack Status*: Failed
   - H': a *person* smokes a cigarette while sitting on his scooter.
     Label: Neutral, *Attack Status*: Failed
   - H': a *male* smokes a cigarette while sitting on his scooter.
     Label: Neutral, *Attack Status*: Failed
   - H': an *organism* smokes a cigarette while sitting on his scooter.
     Label: Contradiction, *Attack Status*: Success
3. Remaining Transformations: No perturbations as the label shift rule does not exist for this transformation.

We run the experiments on the BERT base model with both SNLI and MNLI datasets. The results for a different number of markers are given in the tables below.

Table 2. Attack Results on BERT finetuned on SNLI and MNLI

No. of markers	SNLI			MNLI		
	Successful attacks	Failed attacks	Attack accuracy	Successful attacks	Failed attacks	Attack accuracy
1	2181	3086	41.4%	244	4699	4.9%
2	3289	1978	62.4%	466	4477	9.4%
3	3833	1434	72.7%	588	4355	11.8%
4	4095	1172	77.7%	711	4232	14.3%
5	4199	1068	79.7%	763	4180	15.4%

6 Discussion

As seen in Table 2, our attacks achieved an attack accuracy of 79% on the BERT model finetuned on SNLI. This shows that though the model performed well on benchmark datasets, it has a poor understanding of monotonic reasoning and fails at simple lexical monotonic inferences. Meanwhile, BERT finetuned on MNLI has achieved 84.6% accuracy (Attack accuracy being 15.4%) on the adversarial dataset. $BERT_{MNLI}$ being more powerful than $BERT_{SNLI}$, it can be surmised that the model can withstand the attacks better than the latter. From these results, we may assume that the $BERT_{MNLI}$ model has managed to capture simple monotonic inferences. However, keeping in mind the length of the sentences in MNLI it may be that single-word substitutions performed might not be sufficient to validate their monotonic reasoning capacity.

We compare our attack accuracies with adversarial attack methods, namely TextFooler [11] and BERT-Attack [13] as seen in Table 3. We also give a detailed comparative analysis of our model with TextFooler and BertAttack. TextFooler is a state-of-the-art baseline to generate adversarial text. Similar to our methodology, they select markers and replace them to create perturbations. In TextFooler, a marker is selected by sorting the words on their importance ranking and picking the highest word after removing the stop words. Once the marker is selected, its synonyms are extracted for replacement. Synonyms are picked by comparing the cosine similarities of the words in the vocabulary with that of the marker. Parts of speech is ensured to be the same to generate grammatically valid statements. The semantic similarity of the sentences is obtained from the cosine similarity of their embeddings. The attacks are performed by replacing the marker with the best synonym resulting in label preserving perturbations.

Similarly, BertAttack finds vulnerable words by masking each word in the sentence and comparing their logit scores. K replacement words for the vulnerable words are then generated using the BERT model. No additional grammatical or semantic checks are performed as BERT is context aware. Although the accuracies of TextFooler and BertAttack are higher than our attack accuracy, the semantic similarity score for our attacks obtained using Universal Sentence Encoding model [3] is considerably greater as seen in Table 3.

As earlier we also note that the attack labels of the two above methods can be prone to errors due to lack of checking of sense of the word and illegitimate words being introduced into the text. Our method for generating adversarial examples is much more computationally efficient than TextAttack [19]. We give some examples below.

Table 3. Accuracies and semantic similarity of the attacks

Attack	Accuracy on SNLI	Accuracy on MNLI	semantic Similarity
TextFooler	96%	90.4%	0.45
BERT-Attack	92.6%	92.1%	0.40
Ours	79.7%	15.4%	0.87

6.1 Comparison of Examples with TextFooler and BertAttack

The following examples illustrate the issues with the approach followed by TextFooler and BertAttack:

- **Errors in label shifts**: The replacement words considered are not always synonyms, thus leading to incorrect attacks as the perturbations are not label preserving.
  - **TextFooler-** Original: A man in a blue shirt is looking up at a *dog.*
    Perturbation: A man in a blue shirt is looking up at a *canine.*
  - **BertAttack-** Original: A person throwing something for her *dog.*
    Perturbation: A person throwing something for her *puppy.*
    Explanation: The relation between canine and dog is hypernymy, while that between dog and puppy is hyponymy rather than synonymy. The label will therefore be dependent on the monotonicity of the word.
- **Improper Perturbations**
  - Original: There is a little *boy* who *likes* the colour brown.
    Perturbation:
    - **TextFooler**: There is a little boy who *iikes* the colour brown.
    - **Ours**: There is a little *person* who likes the colour brown.
  - Original: Girl *plays nintendo.*
    Perturbation:
    - **BertAttack**: Girl *and facebook.*
    - **Ours**: *Scout* plays nintendo.

    Explanation: Non-existent words or unrelated words.
- **Incorrect Sense** The sense of the replacement word is completely different from the original sense, thus changing the semantics of the sentence. Though parts of speech is considered to ensure the grammaticality of the text, the morphology of the words is not maintained, resulting in sentences with improper grammar.

- Original: The dogs are *running* along the shore to meet their master who just beached his *kayak*.
  Perturbation:
  - **TextFooler**: The dogs are *executed* along the shore to meet their master who just beached his kayak.
  - **Ours**: The dogs are running along the shore to meet their master who just beached his *canoe*.

7 Conclusion

We have proposed a novel approach to generate adversarial datasets from benchmark NLI datasets. These attacks help in assessing a Neural Language Model's understanding of monotonicity reasoning. We evaluate the generated datasets on state-of-the-art NLI models and analyze their performance. We conclude with a comparison with state-of-the-art adversarial attacks and show that our methods produce more semantically similar sentences and do not suffer from lexical errors.

While single word substitutions are easy to incorporate and effective, not all concepts can be encapsulated by a single word. Future work can focus on structural changes with phrase replacement to better test the model's monotonic reasoning ability. Another line of work can be explanation-based attacks that can probe the model's ability to generalize utilizing the context of the sentences. While adversarial analysis illuminates the workings of the model, it remains to be seen if such rules can be incorporated into the models efficiently. So far, while there's work [6] which tries to do so, retraining a model for such a task is computationally expensive while humans can integrate such logical reasoning much more easily. This remains an open area of research.

Acknowledgements. Part of this work was funded by the Institute of Eminence Grant, UoH-IoE-RC3-21-050.

References

1. Barba, E., Pasini, T., Navigli, R.: ESC: Redesigning WSD with extractive sense comprehension. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. ACL, June 2021. https://doi.org/10.18653/v1/2021.naacl-main.371
2. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. ACL, September 2015. https://doi.org/10.18653/v1/D15-1075
3. Cer, D., et al.: Universal sentence encoder for English. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. ACL, November 2018. https://doi.org/10.18653/v1/D18-2029

4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics NAACL (2019). https://doi.org/10.18653/v1/N19-1423
5. Ebrahimi, J., Rao, A., Lowd, D., Dou, D.: 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). ACL, July 2018. https://doi.org/10.18653/v1/P18-2006
6. Glockner, M., Shwartz, V., Goldberg, Y.: 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). ACL, July 2018. https://doi.org/10.18653/v1/P18-2103
7. Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., Smith, N.A.: Annotation artifacts in natural language inference data. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 2 (Short Papers). ACL, June 2018. https://doi.org/10.18653/v1/N18-2017
8. Hossain, M.M., Kovatchev, V., Dutta, P., Kao, T., Wei, E., Blanco, E.: An analysis of natural language inference benchmarks through the lens of negation. In: EMNLP (2020). https://doi.org/10.18653/v1/2020.emnlp-main.732
9. Hu, H., Moss, L.: Polarity computations in flexible categorial grammar. In: Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics. ACL, June 2018. https://doi.org/10.18653/v1/S18-2015
10. Jia, R., Liang, P.: Adversarial examples for evaluating reading comprehension systems. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. ACL, September 2017. https://doi.org/10.18653/v1/D17-1215
11. Jin, D., Jin, Z., Zhou, J.T., Szolovits, P.: 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, vol. 34(05), April 2020. https://doi.org/10.1609/aaai.v34i05.6311
12. Li, J., Ji, S., Du, T., Li, B., Wang, T.: TextBugger: generating adversarial text against real-world applications. In: Proceedings of the Symposium on Networks and Distributed System Security, December 2018. https://doi.org/10.14722/ndss.2019.23138
13. Li, L., Ma, R., Guo, Q., Xue, X., Qiu, X.: BERT-ATTACK: Adversarial attack against BERT using BERT. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). ACL, November 2020. https://doi.org/10.18653/v1/2020.emnlp-main.500
14. Li, T., Gupta, V., Mehta, M., Srikumar, V.: A logic-driven framework for consistency of neural models. 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). ACL, November 2019. https://doi.org/10.18653/v1/D19-1405
15. MacCartney, B., Manning, C.D.: An extended model of natural logic. In: Proceedings of the Eighth International Conference on Computational Semantics. ACL, January 2009. https://doi.org/10.3115/1693756.1693772, https://aclanthology.org/W09-3714
16. Maheshwary, R., Maheshwary, S., Pudi, V.: Generating natural language attacks in a hard label black box setting (2021). https://doi.org/10.1609/aaai.v35i15.17595

17. Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., Zamparelli, R.: SemEval-2014 task 1: evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). ACL, August 2014. https://doi.org/10.3115/v1/S14-2001
18. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM. **38**(11), 39-41 (1995). https://doi.org/10.1145/219717.219748
19. Morris, J., Lifland, E., Yoo, J.Y., Grigsby, J., Jin, D., Qi, Y.: 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. ACL, October 2020. https://doi.org/10.18653/v1/2020.emnlp-demos.16
20. Naik, A., Ravichander, A., Sadeh, N., Rose, C., Neubig, G.: Stress test evaluation for natural language inference. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 2340–2353. ACL, Santa Fe, New Mexico, USA, August 2018. https://aclanthology.org/C18-1198
21. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015). https://doi.org/10.1109/CVPR.2015.7298640
22. Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., Van Durme, B.: Hypothesis only baselines in natural language inference. In: Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics. ACL, June 2018. https://doi.org/10.18653/v1/S18-2023
23. Richardson, K., Hu, H., Moss, L., Sabharwal, A.: Probing natural language inference models through semantic fragments. Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34(05), April 2020. https://doi.org/10.1609/aaai.v34i05.6397
24. Saha, S., Nie, Y., Bansal, M.: ConjNLI: Natural language inference over conjunctive sentences. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). ACL, November 2020. https://doi.org/10.18653/v1/2020.emnlp-main.661
25. Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y.: MPNet: masked and permuted pretraining for language understanding. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 16857–16867 (2020)
26. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K. (eds.) Advances in Neural Information Processing Systems, vol. 27 (2014)
27. Tarunesh, I., Aditya, S., Choudhury, M.: LoNLI: an extensible framework for testing diverse logical reasoning capabilities for NLI (2021). https://doi.org/10.48550/ARXIV.2112.02333
28. Traylor, A., Feiman, R., Pavlick, E.: AND does not mean OR: using formal languages to study language models' representations. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, (Volume 2: Short Papers). ACL, August 2021. https://doi.org/10.18653/v1/2021.acl-short.21
29. Vaswani, A., et al.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS 2017, vol. 30 (2017)

30. Vijayaraghavan, P., Roy, D.: Generating black-box adversarial examples for text classifiers using a deep reinforced model. In: Brefeld, U., Fromont, E., Hotho, A., Knobbe, A., Maathuis, M., Robardet, C. (eds.) ECML PKDD 2019. LNCS (LNAI), vol. 11907, pp. 711–726. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-46147-8_43
31. Warstadt, A., Singh, A., Bowman, S.R.: Neural network acceptability judgments. Trans. Assoc. Comput. Linguist. **7**, 625–641 (2019). https://doi.org/10.1162/tacl_a_00290
32. Williams, A., Nangia, N., Bowman, S.: 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, vol. 1 (Long Papers). ACL, June 2018. https://doi.org/10.18653/v1/N18-1101
33. Yanaka, H., et al.: Can neural networks understand monotonicity reasoning? In: Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. ACL, August 2019. https://doi.org/10.18653/v1/W19-4804
34. Zhao, Z., Dua, D., Singh, S.: Generating natural adversarial examples. In: International Conference on Learning Representations (2018). https://openreview.net/forum?id=H1BLjgZCb