ROBUSTLR: Evaluating Robustness to Logical Perturbation in Deductive Reasoning

Soumya Sanyal **Zeyi Liao** **Xiang Ren**
University of Southern California
{soumyasa, zeyiliao, xiangren}@usc.edu

Abstract

Transformers have been shown to be able to perform deductive reasoning on a logical rulebase containing rules and statements written in English natural language. While the progress is promising, it is currently unclear if these models indeed perform logical reasoning by understanding the underlying logical semantics in the language. To this end, we propose ROBUSTLR, a suite of evaluation datasets that evaluate the robustness of these models to minimal logical edits in rulebases and some standard logical equivalence conditions. In our experiments with RoBERTa and T5, we find that the models trained in prior works do not perform consistently on the different perturbations in ROBUSTLR, thus showing that the models are not robust to the proposed logical perturbations. Further, we find that the models find it especially hard to learn logical negation and disjunction operators. Overall, using our evaluation sets, we demonstrate some shortcomings of the deductive reasoning-based language models, which can eventually help towards designing better models for logical reasoning over natural language.

1 Introduction

Building systems that can automatically reason over a given context to generate valid logical inferences has been a long pursued goal within the field of AI (McCarthy, 1959; Rocktäschel and Riedel, 2017; Manhaeve et al., 2019). Recently, Clark et al. (2020) have shown that transformers (Vaswani et al., 2017) are able to emulate deductive reasoning on a logical rulebase (henceforth referred to as a *theory*) containing rules and statements written in natural language. Following this, recent works (Tafjord et al., 2021; Saha et al., 2020, 2021; Sanyal and Ren, 2021) show that such models can also produce the reasoning steps (i.e., the *proof graph*) that emulate the model's logical reasoning process. While these advances are impressive, it is currently unclear if these models are indeed able to use logical reasoning robustly by understanding the semantics of the different logical operators involved in this task.

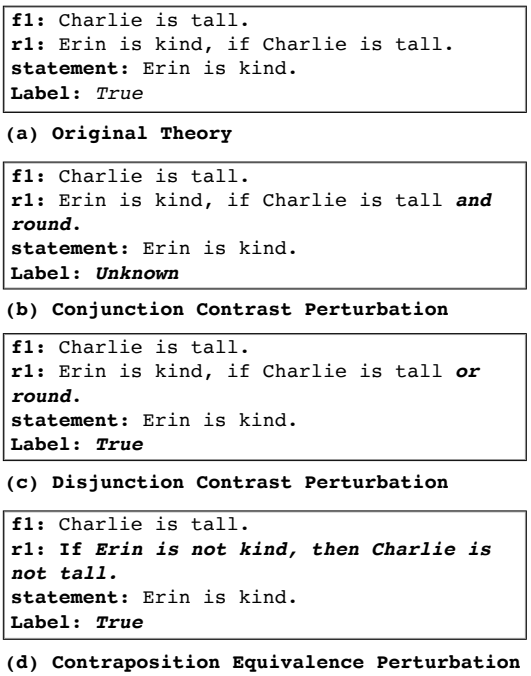

Figure 1: **Overview of ROBUSTLR.** (a) An original theory contains facts, rules, a statement, and the entailment label. The Logical Contrast set perturbations using conjunction and disjunction are shown in bold in (b) and (c), respectively. In the first case the label changes to Unknown as the statement is no longer provable. In (d), we show one of the Logical Equivalence perturbations where the rule is paraphrased using logical contraposition. Please refer to Section 3.1 for more details.

Logical reasoning, in combination with other abilities, is an important skill required in all kinds of NLP tasks such as NLI (Dagan et al., 2006), Question Answering (Yang et al., 2018a), etc. Thus a fundamental question that needs to be addressed is whether the above models perform deductive rea-

soning by using the theory in the right manner, or rather learn some spurious patterns from the data to answer the question. Some prior works Gururangan et al. (2018); Chen and Durrett (2019); McCoy et al. (2019) have found that models solving different reasoning tasks tend to exploit spurious correlations between the context/question and the label. A similar study is missing for deductive reasoning tasks involving logical operations.

A key hypothesis of this work is that if a model performs logical reasoning in natural language as expected, it should be able to correctly solve theories that are logically perturbed within some logical constraints. To test this, we develop RobustLR, a suite of deductive reasoning evaluation sets, containing multiple logical perturbations across two main aspects. The first aspect we aim to evaluate is how well these models understand the three common logical operators: AND ($\wedge$), OR ($\vee$), and NOT ($\neg$). Inspired by the idea of creating contrast sets (Gardner et al., 2020), we design the Logical Contrast set, where theories are minimally modified so that we can test the model's robustness across different operators. Examples of this set are shown in Figure 1(b) and 1(c). The second aspect we focus on is the ability of the model to perform consistently across different *logical paraphrases*. A logical paraphrase uses standard equivalence conditions in logic to replace a rule with another equivalent form, essentially rewriting the existing theory. This poses a different challenge than standard language paraphrase since the model needs to understand that the underlying logical structure of two equivalent sentences mean the same thing. Based on this aspect, we design the Logical Equivalence set, where we test three logical equivalences. An example of the equivalence perturbation is shown in Figure 1(d).

To test the model performance on RobustLR, we first fine-tune them on different training datasets and then evaluate on the test sets mentioned above. Overall, we find that language models (LMs) fine-tuned on different deductive reasoning datasets are not sufficiently robust to the Logical Contrast set. Specifically, we find that models are poor at understanding logical OR operators and negations in sentences. Additionally, we find that the model performance is significantly inconsistent on some subsets of Logical Equivalence set. Thus, using RobustLR, we demonstrate some important limitations of the language models trained for deductive reasoning. We hope that this research should eventually help with designing better models targeted to solving logical reasoning in a more robust manner.

2 Background

In this section, we first give some background on the task of deductive reasoning, and also describe the models proposed by Clark et al. (2020) to solve this task.

Deductive Reasoning In deductive reasoning, we predict whether a given theory T supports a statement s or not. We define a theory T as a set of facts $F = \{f_1, f_2, \ldots, f_n\}$ and rules $R = \{r_1, r_2, \ldots, r_m\}$ expressed in natural language (See Figure 1 for an example). For a given theory, a statement can be either provably supported, provably unsupported (i.e., the negation of the statement is provable), or not provable at all. This leads to a 3-class classification problem, with the labels being *True*, *False*, and *Unknown*, respectively. In this work, we focus on this classification task, where we expect the model to correctly predict the entailment of a statement for a given theory. In Figure 1, the statement is entailed by the theory, leading to the label True. It can be proved by simply using fact f_1 and rule r_1 to derive the statement. Formally, we define the proof set of a statement s, denoted by $G(T, s)$, as the set of rules and facts that are required to obtain the statement s from the theory.

Models There have been some recent progress in solving deductive reasoning for natural language using pre-trained language models. Clark et al. (2020) finetune a RoBERTa-Large model on a synthetic dataset to solve deductive reasoning. The theory and statement are concatenated to generate the input, and the model output is the predicted label. On similar lines, Tafjord et al. (2021) propose a T5-based model to both solve the deductive reasoning task as well as generate proofs for entailment. Please refer to Appendix A for more details on the usage of these models for deductive reasoning.

3 The RobustLR dataset

In this section, we first give an overview of RobustLR, set some notations, and then describe the proposed evaluation sets one by one.

3.1 Overview

The main goal of RobustLR is to evaluate the robustness of the model behavior on various log-

ical reasoning probes. First, we assess the ability of the model to correctly capture the semantics of different logical operators, when presented in minimally edited contrast inputs. A contrast set (Gardner et al., 2020) is one where the input is changed minimally but meaningfully, such that there is (typically) some change in the label. We further evaluate whether the model can perform consistently when shown the same input with different *logical paraphrases*. A theory can be logically paraphrased by modifying the rules using standard logical equivalence conditions [1]. Thus, this leads to two evaluation categories in ROBUSTLR. The Logical Contrast set tests the LM's robustness to three logical operators: conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$). And the Logical Equivalence set evaluates the model's consistency in solving various logically equivalent theories. A strong model should be robust to both the minimally edited inputs and logical equivalences. Overall, these evaluation sets probe a model trained to solve the deductive reasoning task, to check whether it indeed learns the semantics of the logical operators and their underlying working principles.

3.2 Notations

We consider two predicate forms in our dataset - unary and binary. A unary predicate contains only one argument and is denoted by $X(a)$. Similarly, a binary predicate is represented as $X(a, b)$. Here, X is the predicate relation and a, b are the variables. An *atomic* predicate is defined as either a predicate or the negation of the predicate (denoted as $\neg X(a)$). In contrast, a *complex* predicate can contain multiple predicates (or their negated forms) combined using logical operators conjunction ($\wedge$) and disjunction ($\vee$).

Internally, we maintain a symbolic representation of these facts and rules, enabling us to later create the different evaluation sets of ROBUSTLR. A fact is symbolically represented by a predicate. In fact, the facts in our dataset are always atomic predicates. A rule is symbolically represented by a logical connection between predicates, separated by the "implies that" logical symbol ($\implies$). Thus, a rule can be defined as $p \implies q$, where the LHS p and RHS q are atomic or complex predicates. If both p and q consist of atomic predicates, then the rule is called a *simple* rule. A *compound* rule is one

[1] https://en.wikipedia.org/wiki/Logical_equivalence

```
f1: Charlie is tall.
f2: Erin is not the brother of Gary.
r1: If Charlie is tall or smart, then
Gary is kind.
r2: Charlie is round if Gary is kind.
statement: Charlie is not round.
label: False
```

(a) Natural Language Form

```
f1: tall(Charlie)
f2: ¬brother(Erin, Gary)
r1: tall(Charlie) ∨ smart(Charlie) →
kind(Gary)
r2: kind(Gary) → round(Charlie)
statement: ¬round(Charlie)
label: False
```

(b) Logical Form

Figure 2: **Logical Form of a Theory.** (a) A theory in natural language. (b). The corresponding logical form of the theory. Please refer to Section 3.2 for more details.

where at least one of p and q contain some complex predicates. An example of a natural language theory and its corresponding logical form is shown in Figure 2. In this, fact f_1 is a unary atomic predicate, while fact f_2 is a binary predicate. Rule r_1 is a compound rule, with the LHS p of the rule being a complex predicate. Rule r_2 is a simple rule.

3.3 Logical Contrast Sets

In this evaluation set we probe the ability of the model to robustly understand the three different logical operators ($\wedge, \vee, \neg$). For this, we develop different contrast sets (Gardner et al., 2020) with minimal editing of the theory, probing specific reasoning abilities of different operators. The key intuition is to evaluate if the model is able to understand the minor changes in the theory brought by the addition of logical operators, and predict the change in label accordingly. First, we describe the different perturbations below, and then propose two versions of the contrast sets.

Contrast Perturbations For a given theory T and statement s, we first select a rule to be modified such that it part of the proof set $G(T, s)$. This ensures that our perturbation would likely influence the model's reasoning process while predicting entailment of the statement s. Next, we add an unseen predicate t to the rule LHS p of one of the rules using conjunction ($\wedge$) or disjunction ($\vee$).

In some further variants of perturbations, we include the predicate t (or the negated $\neg t$) as a fact in the theory, leading to different labels. Lastly, we also negate the rule RHS q to introduce the logical negation ($\neg$) perturbations.

The proposed perturbations described above are shown in Table 1 for the conjunction operator. The first row is a base theory which is used to generate these contrast sets. In the next set of triads, the rule is modified to have an unseen predicate t in conjunction with the existing rule LHS. Here t is a predicate that is not part of the existing facts and inferences in the theory (hence, referred to as unseen predicate). Additionally, we add t (or $\neg t$) as part of the facts in the theory. This lead to modification of the label as shown in rows 3-4. For the next set of triads, we modify the base rule to have a negated rule RHS $\neg q$. The corresponding label changes are shown in rows 5-7. In this table, we assume the label of the statement is *True* for the base theory in row 1. The perturbation set for the label *False* is shown in Table 9 in Appendix. We group these perturbations into three classes as shown in Table 1: BASE, CONJ, CONJ+NEG. These groups are based on which logical operator is the new addition with respect to the base theory. If a model performs accurately on this contrast set, we expect that the model understands the semantics of conjunction and negation logical operators reasonably well.

Similar to the above Conjunction Contrast Set, we show the perturbations considered in this set in Table 2, where the distractor is added to the rule LHS using disjunction ($\vee$). More such perturbations with the base case being the label *False* is shown in Table 10 in Appendix.

We propose two variants of the Logical Contrast set using the above perturbations, as described below.

Without Distractors We define distractors as the facts and rules that are not part of the proof set for a given theory and statement. For cases where no proof set is feasible (i.e., instances with label *Unknown*) we assume there are no distractors. In layman terms, distractors can confuse the reasoning process of the model by adding unnecessary information. Thus, a set without distractors should be easier for the model to reason with. Thus, in this version, we only keep the essential facts and rules in the theory and remove any unnecessary information. Formally, for a given theory T and statement s, we keep the facts and rules that are in the proof set $G(T, s)$. To create the contrast sets, we follow the same perturbations as described above.

Modified Rule	Facts	Statement	Label	Group
$p \implies q$	$\{p\}$	q	*True*	BASE
$p \wedge t \implies q$	$\{p\}$	q	*Unknown*	CONJ
$p \wedge t \implies q$	$\{p, t\}$	q	*True*	CONJ
$p \wedge t \implies q$	$\{p, \neg t\}$	q	*Unknown*	CONJ+NEG
$p \wedge t \implies \neg q$	$\{p\}$	q	*Unknown*	CONJ+NEG
$p \wedge t \implies \neg q$	$\{p, t\}$	q	*False*	CONJ+NEG
$p \wedge t \implies \neg q$	$\{p, \neg t\}$	q	*Unknown*	CONJ+NEG

Table 1: **Conjunction Contrast Perturbations**. The minimal edits done to a base theory (first row) for testing the conjunction and negation reasoning abilities. The group reflects the overall change in theory w.r.t. the base theory.

Modified Rule	Facts	Statement	Label	Group
$p \implies q$	$\{p\}$	q	*True*	BASE
$p \vee t \implies q$	$\{p\}$	q	*True*	DISJ
$p \vee t \implies q$	$\{p, t\}$	q	*True*	DISJ
$p \vee t \implies q$	$\{\neg p, \neg t\}$	q	*Unknown*	DISJ+NEG
$p \vee t \implies \neg q$	$\{p\}$	q	*False*	DISJ+NEG
$p \vee t \implies \neg q$	$\{p, t\}$	q	*False*	DISJ+NEG
$p \vee t \implies \neg q$	$\{\neg p, \neg t\}$	q	*Unknown*	DISJ+NEG

Table 2: **Disjunction Contrast Perturbations**. The minimal edits done to a base theory (first row) for testing the disjunction and negation reasoning abilities. The group reflects the overall change in theory w.r.t. the base theory.

With Distractors In this version, do no such filtering as above, thus keeping all the facts and rules in the original theory. This set should be more challenging for the model as it has to first understand which facts and rules are important and then use them to predict the entailment.

3.4 Logical Equivalence Sets

The Logical Equivalence set contain theories where the underlying symbolic representation of a rule is replaced by another representation that is logically equivalent. The logical equivalent form of a rule can be derived from standard logical equivalence conditions, as described below:

- **Contrapositive**: $p \implies q \equiv \neg q \implies \neg p$
- **Distributive 1**: $(p \implies q) \wedge (p \implies r) \equiv p \implies (q \wedge r)$
- **Distributive 2**: $(p \implies q) \wedge (r \implies q) \equiv (p \vee r) \implies q$

Here p, q, r can be both atomic predicates or complex predicates. For the Contrapositive equivalence, every rule r_i in the theory T is replaced by the logically equivalent form to create a new logically equivalent theory T'. Similarly, for Distributive 1 and Distributive 2, a pair of rules in T are merged according to the equivalence to create a new theory

T'. Note that in both instances, the theory T' would still have the same label for a given statement as the logical steps required to solve the task remains the same. These modifications are more challenging than traditional surface-level paraphrases of the natural language text, as it forces the model to understand the equivalence of different symbolic representations. Overall, the Logical Equivalence set evaluates whether the LM is robust to logical perturbations of the theory.

3.5 Evaluation Protocol

We report the model performance on these evaluation datasets as the weighted-F1 score from the Scikit-learn (Pedregosa et al., 2011). Since all the labels are equally important, a macro-average is more meaningful for us. The weighted-F1 score modifies the macro-F1 to take any class imbalance into account. We have label imbalance by design of the perturbations in the Logical Contrast sets, as is evident from Tables 1 and 2. We compute the F1-score for the base theory and all its perturbations, and average the score across all theories in the evaluation set.

4 ROBUSTLR Design Details

The main limitation of the RuleTaker dataset (Clark et al., 2020) is the lack of any systematic control over the underlying symbolic representation of facts and rules used in the theories. This makes it quite challenging to perturb the theories in any logical manner. In ROBUSTLR, we clearly define the symbolic form of each fact and rule in the theory, enabling us to automatically generate the Logical Equivalence and Logical Contrast sets described above. In this section, we first describe the details of the domain of our dataset and then outline the sampling technique used to generate the dataset.

4.1 Dataset Domain

We keep the domains of our dataset fairly simple. The domains of X and a in the unary predicate $X(a)$ are the simple English adjectives and the proper names respectively. Examples of this predicate form are "green(Alex)", "kind(John)", etc. Each predicate is associated to the English template sentence form "$\{a\}$ is $\{X\}$.". We note that RuleTaker (Clark et al., 2020) also contains similar predicates. For the binary predicate $X(a, b)$, we consider family relationships and proper names as the domain of X and a respectively. Some examples of this predicate form are "daughter(Mary, Gary)","father(Bob, John)", etc. Each predicate is associated with template sentences such as "$\{a\}$ is the $\{X\}$ of $\{b\}$.", "The $\{X\}$ of $\{b\}$ is $\{a\}$.", etc. Note that, currently, we do not enforce any gender constraints on the names, thus allowing predicates such as "daughter(Bob, Gary)", which might be unlikely based on the gender associated statistically to names in English.

For the rules, we follow the same domain as the facts as mentioned above. Thus, examples of some simple rules consisting of atomic predicates are "green(Alex) $\implies$ daughter(Bob, Gary)", "$\neg$ father(Bob, John) $\implies$ kind(John)", etc. Similarly, examples of some compound rules containing complex predicates are "green(Alex) $\vee$ smart(Bob) $\implies$ daughter(Bob, Gary) $\wedge$ $\neg$ kind(John)", etc. We note that, for the sake of keeping the theories deterministic, we avoid using the disjunction operator in the RHS of a rule. A rule of the form $p \implies q$ is associated with templates such as "If $\{p\}$ then $\{q\}$.", "$\{q\}$ if $\{p\}$.", where the p's and q's can be recursively resolved to their own templates as defined in the predicates.

4.2 Dataset Sampling

For sampling the theories in ROBUSTLR, we use the algorithm described in Algorithm 1, which is a modified version of the Label-Priority sampling (Anonymous, 2022). At a high-level, we sample different predicates and assign the labels True/False to them. After that, we divide the set into multiple levels. This helps us in sampling theories with multi-hop reasoning depths. After that, rules are derived by connecting predicates with the same label between two different levels. Finally, the True predicates at level 0 form the facts in the theory, the connections denote the rules, and the highest level denotes the candidate statements.

5 Experimental Setup

Here, we describe the details of the training dataset considered for this task and the models we fine-tune to evaluate on ROBUSTLR.

5.1 Dataset Details

We use five different training datasets as described below:

- **NO LOGIC OP**: This dataset is created such that the theories and statements do not contain any of the three logical operators. This can be

Algorithm 1: Sampling Algorithm

Input : $vocab$ containing the corpus of all predicates, $ruleset$ containing the set of valid rules, predicate negation probability n_1, statement negation probability n_2, max reasoning depth d.

Output : A theory containing a set of $facts$ and $rules$, a $statement$, and a corresponding $label \in \{0, 1, 2\}$

1 $pred\_num \sim U[10, 30]$
2 $preds \leftarrow \text{SAMPLE}(vocab, pred\_num)$
3 set $l \sim U[1, d]$ and group $preds$ into l layers
4 $rules \leftarrow [\,]$
5 **for** *predicate* p *in layer* $1 \leq i \leq l$ **do**
6 Negate p with probability n_1
7 $q \sim U[0, 1]$
8 assign label q to predicate p
9 **if** $i \geq 1$ **then**
10 $k \sim U[1, 2]$
11 $cand \leftarrow p$ in layer $i - 1$ with label q
12 $body \leftarrow \text{SAMPLE}(cand, k)$
13 **if** $len(body) > 1$ **then**
14 $operator \leftarrow \text{SAMPLE}([\wedge, \vee], 1)$
15 Compose the predicates in the $body$
16 using $operator$
17 **end if**
18 $r \leftarrow (body \implies p)$
19 **if** $\text{VALIDATE}(ruleset, r)$ **then**
20 add r to $rules$
21 **else**
22 /* Rule r does not match any valid rule forms, so the predicate is not provable */
23 assign label 0 to predicate p
24 **end if**
25 **end if**
26 **end for**
27 $facts \leftarrow$ predicates in layer 1 with label 1
28 $statement \leftarrow \text{SAMPLE}(preds, 1)$
29 $label \leftarrow$ pre-assigned label for $statement$
30 **if** $label == 1$ **then**
31 Negate the $statement$ with probability n_2
32 $label \leftarrow 2$
33 **end if**
34 **return** $(facts, rules, statement, label)$

done by setting the negation probabilities n_1 and n_2 to 0 in lines 6 and 31 respectively, and $k = 1$ in line 10 of Algorithm 1.

- **NEG**: In this dataset, we allow negations in both facts and rules, but restrict to only using simple rules. Negations are decided uniformly by setting $n_1 = n_2 = 0.5$. We continue with this setting for the rest of the datasets mentioned below.

- **AND + NEG**: Here, we restrict the logical operator for compound rules to logical AND by setting *operator* to $\wedge$ in line 14 of Algorithm 1. Any compound rule in this dataset contains only the AND operator.

Training Dataset	RoBERTa-Large	T5-Large
NO LOGIC OP	99.95	99.86
NEG	99.99	99.97
AND + NEG	99.98	99.8
OR + NEG	99.76	99.24
AND + OR + NEG	99.98	99.34

Table 3: Performance of RoBERTa-Large and T5-Large on in-domain held-out set. Both models perform almost accurately when fine-tuned on these training datasets.

- **OR + NEG**: Similar to AND + NEG, we restrict the logical operator of the compound rules to OR ($\vee$).

- **AND + OR + NEG**: This dataset has all the three logical operators present.

We aim to understand the effect of these training datasets on the evaluation sets by fine-tuning the model on each dataset separately. Please refer to Appendix C for more details on the training and evaluation dataset statistics.

5.2 Models

Following prior works (Clark et al., 2020; Tafjord et al., 2021), we evaluate the performance of two language models: RoBERTa-Large (Liu et al., 2019) and T5-large (Raffel et al., 2020). To evaluate a model, we first fine-tune it on one of the training dataset mentioned above, and then evaluate on the evaluation set. The performance of the LMs on in-distribution held-out data (sampled using same algorithm parameters as the training data) are shown in Table 3. We note that the models are able to solve the training dataset almost perfectly in all cases. Please refer to Appendix A for more details on the specific input formats for each model and Appendix B for the hyperparameter settings and other implementation details.

6 Results

In this section, we first evaluate the LM on RoBUSTLR evaluation sets after fine-tuning on the training datasets mentioned in Section 5.1. Next, we study the effect of pre-training versus fine-tuning a pre-trained checkpoint. Lastly, we perform human evaluation on a subset to understand the human upper bound for our evaluation sets.

6.1 Performance on Logical Contrast set

Overall Result We finetune both RoBERTa-Large and T5-Large models on different training

Training Dataset	RoBERTa-Large				T5-Large			
	Without Distractors		With Distractors		Without Distractors		With Distractors	
	Conjunction	Disjunction	Conjunction	Disjunction	Conjunction	Disjunction	Conjunction	Disjunction
NO LOGIC OP	0.61	0.36	0.60	0.34	0.57	0.34	0.56	0.34
NEG	0.62	0.48	0.62	0.45	0.67	0.47	0.63	0.44
AND + NEG	0.79	0.53	0.70	0.46	0.68	0.43	0.62	0.42
OR + NEG	0.37	0.52	0.39	0.52	0.54	0.51	0.44	0.50
AND + OR + NEG	0.83	0.6	0.74	0.55	0.69	0.53	0.67	0.51
Average	0.64	0.50	0.61	0.46	0.63	0.46	0.58	0.44

Table 4: Performance of RoBERTa-Large and T5-Large on Logical Contrast sets. We report the weighted-F1 score. Overall, seeing all the logical operators leads to best performance. Please refer to Section 6.1 for more details.

Training Dataset	RoBERTa-Large			T5-Large		
	Contrapositive	Distributive 1	Distributive 2	Contrapositive	Distributive 1	Distributive 2
NO LOGIC OP	0.56	0.51	0.50	0.57	0.51	0.50
NEG	0.76	1.00	1.00	0.77	0.99	1.00
AND + NEG	0.79	0.98	1.00	0.77	0.96	1.00
OR + NEG	0.81	1.00	1.00	0.80	0.99	1.00
AND + OR + NEG	0.82	1.00	1.00	0.79	0.97	1.00

Table 5: Performance of RoBERTa-Large and T5-Large on Logical Equivalence sets. We report the weighted-F1 score. Overall, we find that the performance drops for contrapositive equivalences, while it remains consistent on the other two. Please refer to Section 6.2 for more details.

datasets and evaluate them on the Logical Contrast set. The two variants of the Logical Contrast set, with and without distractors, are further subdivided based on the type of operators used in the perturbation. The results are shown in Table 4. We observe that with increasing variety of logical operators in the training data, the performance generally improves across different datasets and models. But there are some notable exceptions. For instance, we find that models trained on the OR + NEG training data perform worse than a model trained on NO LOGIC OP, when testing on Conjunction contrast perturbations. This shows that the model requires training data that is strongly aligned with the operators being evaluated the test set, which is expected. Additionally, we find that the even the best performance on this evaluation set is still significantly degraded from the almost perfect performance in the in-distribution held-out set in Table 3. This shows that these models do not learn the semantics of the logical operators in a robust manner.

Effect of distractors Next, we observe that the performance of both the LMs on the variant without distractors is generally better than with distractors. This shows that there is a non-trivial challenge in retrieving the relevant sentences in the theory and then using the retrieved sentences to reason the entailment within the same model. Doing this in one model can lead to some performance drops.

Variation with logical operators Lastly, we analyze the performance variation with respect to different logical operators. Overall, we find that both the models perform worse on the disjunction contrast perturbations as compared to conjunction. To better understand these performance differences, we plot the model performance for each perturbation group as defined in Section 3.3. We do this for the models trained on the AND + OR + NEG dataset, since the model performance is best when trained on this dataset. The performance plot is shown in Figure 3. We find that the most challenging operator is negation as the score drops significantly on the negation-based contrast sets. This demonstrates that even if the model is trained on negations, it is still not able to learn the negation semantics correctly. Please refer to Appendix E for a detailed results breakdown for each model.

6.2 Performance on Logical Equivalence set

Results on Contrapositive Equivalence Next, we evaluate the fine-tuned LMs on the Logical Equivalence sets. We observe that the model performance degrades significantly for the contrapositive equivalence set, compared to the in-distribution performance in Table 3. Contraposition involves changing the rule into a format that has two negations, thus testing the limits of the model on understanding negations. From the experiments on Logical Contrast sets, we know that negations are

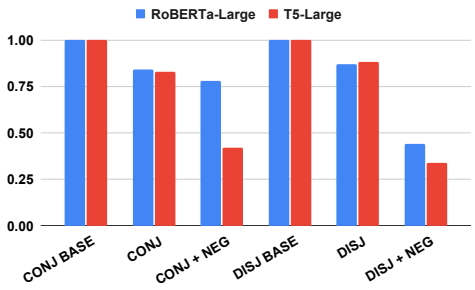

Figure 3: Performance comparison of RoBERTa-Large and T5-Large across different groups of contrast perturbations. We find that negations are hardest to learn across all settings. Refer to Section 6.1 for more details.

not well understood by the model. Thus, these results reinforce our previous findings.

Results on Distributive Equivalence For Distributive 1 and 2 equivalences, we see a nearly perfect score across all settings when trained with at least some logical operator. Only NO LOGIC OP dataset shows a significant drop in performance. These results indicate that the distributive equivalences are not challenging and are likely quite close to the training set distribution. Thus, this shows the importance of the contrast set method to evaluate the language models. While both the Logical Contrast sets and Distributive set contain the logical ops, one is significantly more challenging than the other. We plan to perform further investigation to understand this difference between the two sets.

6.3 Effect of Pre-training

In this part, we evaluate the effect of using a pre-trained checkpoint in our experiments, in comparison with training a RoBERTa-Large architecture from scratch. We train both RoBERTa-Large pre-trained checkpoint and a model from scratch using the AND + OR + NEG dataset, and evaluate on the Logical Contrast set without distractors. The results are shown in rows 1-2 in Table 6. We observe a significant drop in performance, ensuring that knowledge learned during pre-training is crucial for this task.

6.4 Human Evaluation

To better understand the upper limit of some of ROBUSTLR evaluation sets, we ask 3 Computer Science graduate students to annotate 50 randomly sampled theories from the Logical Contrast set without distractor variant. The results are shown in Table 6. We find that humans are roughly equally competent on both the conjunction and disjunction sets. In contrast, the LMs show a biased performance towards conjunction. We believe a strong model should at least perform comparable across both the logical perturbations, similar to humans.

Setup	Conjunction	Disjunction
From scratch using AND + OR + NEG	0.42	0.13
Pre-trained ckpt using AND + OR + NEG	0.83	0.6
Human Performance	0.89	0.84

Table 6: Comparisons between training a model from scratch, training a pre-trained checkpoint, and human performance, on the Logical Contrast set without distractors. Please refer to Sections 6.3 and 6.4 for more details.

7 Related Works

Reasoning in natural language has been a prevalent problem in NLP. There are multiple reasoning datasets, studying different aspects of reasoning over textual inputs. Natural Language Inference (NLI) (Dagan et al., 2006) is a prominent dataset that requires reasoning over text to answer if a statement is entailed, contradicted, or neutral given a hypothesis. HotpotQA (Yang et al., 2018b) tests multi-hop reasoning abilities that require comparisons and inferring missing bridge between sentences. QuaRTz (Tafjord et al., 2019) focuses on qualitative comparisons between everyday properties such as distance, etc. CLUTRR (Sinha et al., 2019) tests whether models can infer biological relationships between entities in a context. RICA (Zhou et al., 2021) requires the model to employ commonsense reasoning to answer questions based on a context.

Recently, there has been an increasing focus on evaluating the logical reasoning abilities of language models. ReClor (Yu et al., 2020) is a MRC-style dataset derived from graduate admissions examinations that involves logical reasoning. LogiQA (Liu et al., 2021) is another similar dataset involving logical reasoning. RuleTaker (Clark et al., 2020) proposes deductive reasoning datasets that require logical reasoning using only the knowledge present in the context. There are very limited works that probe the logical reasoning abilities of language models (LMs). FaiRR (Sanyal et al., 2022) tests the robustness of logical reasoning models when the subjects and attributes in the context are altered to out-of-distribution terms. To the best of our knowledge, ROBUSTLR is the first dataset that

tests how robust these LMs are to different logical perturbations. Inspired by the application of contrast sets (Gardner et al., 2020) in understanding model's decision boundary, we propose multiple Logical Contrast sets to evaluate the robustness of language models to different logical operators. Similarly, we propose Logical Equivalence sets to test if models learn different equivalence conditions defined for symbolic logic.

8 Conclusion

In this paper, we proposed ROBUSTLR, a suite of evaluation datasets to test the robustness of deductive reasoning models to logical perturbations. In ROBUSTLR, we propose two evalutaion sets, Logical Contrast and Logical Equivalence, each probing different aspects of the logical reasoning process. Overall, we find that fine-tuning LMs such as RoBERTa and T5 on deductive reasoning datasets is not sufficient to learn the semantics of the logical operators: conjunction, disjunction, and negation. Although well-aligned training dataset significantly helps with performance, the models still find it challenging to understand negations, both in the context of contrast sets and contraposition equivalences. We hope that this work demonstrates some interesting shortcoming of LMs designed for logical reasoning, that can eventually motivate towards building better reasoning models.

References

Anonymous. 2022. Can bert conduct logical reasoning? on the difficulty of learning to reason from data. *ACL Rolling Review - January 2022*.

Jifan Chen and Greg Durrett. 2019. Understanding dataset design choices for multi-hop reasoning. 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 4026–4032, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3882–3890. International Joint Conferences on Artificial Intelligence Organization. Main track.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. 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, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2021. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Yinhan Liu, Myle Ott, Naman Goyal, and Jingfei Du an. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Robin Manhaeve, Sebastijan Dumancic, A. Kimmig, T. Demeester, and L. D. Raedt. 2019. Deepproblog: Neural probabilistic logic programming. In *BNAIC/BENELEARN*.

John W. McCarthy. 1959. Programs with common sense. In *Proc. Tedding Conf. on the Mechanization of Thought Processes*, pages 75–91.

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*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Tim Rocktäschel and S. Riedel. 2017. End-to-end differentiable proving. In *NeurIPS*.

Swarnadeep Saha, Sayan Ghosh, Shashank Srivastava, and Mohit Bansal. 2020. PRover: Proof generation for interpretable reasoning over rules. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 122–136, Online. Association for Computational Linguistics.

Swarnadeep Saha, Prateek Yadav, and Mohit Bansal. 2021. multiPRover: Generating multiple proofs for improved interpretability in rule reasoning. In *NAACL*.

Soumya Sanyal and Xiang Ren. 2021. Discretized integrated gradients for explaining language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10285–10299. Association for Computational Linguistics.

Soumya Sanyal, Harman Singh, and Xiang Ren. 2022. FaiRR: Faithful and robust deductive reasoning over natural language. *arXiv preprint arXiv:2203.10261*.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. 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 4506–4515, Hong Kong, China. Association for Computational Linguistics.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. QuaRTz: An open-domain dataset of qualitative relationship questions. 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 5941–5946, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018a. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018b. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations (ICLR)*.

Pei Zhou, Rahul Khanna, Seyeon Lee, Bill Yuchen Lin, Daniel Ho, Jay Pujara, and Xiang Ren. 2021. RICA: Evaluating robust inference capabilities based on commonsense axioms. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7560–7579, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

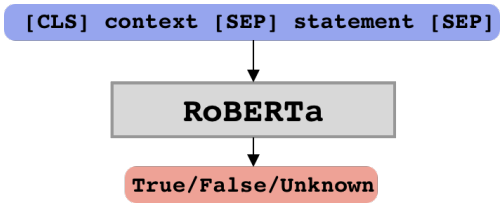

Figure 4: **Overview of the RoBERTa-Large model** - The *context* (containing the facts and rules) and the *statement* are concatenated together as input and passed into a RoBERTa-Large model. The model is trained on cross entropy loss for a 3-class classification task.

A Model implementation details

In this section, we describe the implementation details of the language models used to evaluate ROBUSTLR.

- **RoBERTa-Large**: Following RuleTaker (Clark et al., 2020), we use a pre-trained RoBERTa-Large (Liu et al., 2019) model to perform the classification task. Specifically, we input in the format $[CLS]$ T $[SEP]$ s $[SEP]$ to the RoBERTa-Large model, and extract the $[CLS]$ embedding to predict the label. Here, T is the theory which is the concatenation of the facts and rules, and s is the statement. We use Cross Entropy loss to fine-tune the model on the training dataset.

- **T5-Large**: Following ProofWriter (Tafjord et al., 2021), we train a T5-Large (Raffel et al., 2020) model for the deductive reasoning task. For this, we add a prefix to the T5-Large's input and generate the output in a fixed format. Specifically, we give the input in the format: $\$answer\$$; $\$question\$ = s$; $\$context\$ = T$. Here, T is the theory which is the concatenation of the facts and rules, and s is the statement. And the output is defined to be in format: $\$answer\$ = True/False/Unknown$. The model is trained on the default language modeling loss to match the output format. At evaluation time, we match the output template with the above description and generate the model's predicted label accordingly.

B Hyperparameter Details

Here we use RoBERTa-Large (Liu et al., 2019) and T5-Large (Raffel et al., 2020) models for the 3-class deductive reasoning classification task. We train the pre-trained checkpoints available in the Hugging Face (Wolf et al., 2020) Transformers library. For RoBERTa-Large model, we use AdamW (Loshchilov and Hutter, 2019) with learning rate $1e$-5. For T5-Large, we use AdamW with learning rate $1e$-4. Both models are trained with batch size 8 on Nvidia Quadro RTX 8000 GPUs. Training a single task on one GPU costs nearly 8 hours on average.

Training Dataset	Train	Dev	Test
NO LOGIC OP	51093	10948	10950
NEG	69669	14929	14930
AND, NEG	78370	16793	16795
OR, NEG	53691	11505	11506
AND, OR, NEG	86682	18574	18576

Table 7: Training dataset statistics. Please refer to Section C for more details.

Evaluation Set	Number of instances
Logical Contrast w/o distractor Conjunction	44361
Logical Contrast w/o distractor Disjunction	48761
Logical Contrast w/ distractor Conjunction	45602
Logical Contrast w/ distractor Disjunction	43271
Logical Equivalence Contrapositive	112154
Logical Equivalence Distributive 1	5336
Logical Equivalence Distributive 2	72418

Table 8: Evaluation dataset statistics. Refer to Section C for more details.

C Dataset Statistics

In this section, we describe the training and evaluation dataset statistics. We first train the model on the datasets in Table 7. Each dataset comprises of different types of logical operators to help us in understanding the effect of different logical operators. Then we evaluate the trained models on evaluation datasets mentioned in Table 8. In evaluation, we test the model on two variants of Logical Contrast set: with and without distractors. We further divide each of them into conjunction and disjunction datasets based on the type of perturbation. Lastly, we evaluate the model on three types of Logical Equivalence datasets.

D Contrast Perturbations

Following Section 3.3, we show the conjunction contrast and disjunction contrast perturbations for the case when base theory's label is *False* in Tables 9 and 10, respectively.

Modified Rule	Facts	Statement	Label	Group
$p \implies \neg q$	$\{p\}$	q	*False*	BASE
$p \wedge t \implies \neg q$	$\{p\}$	q	*Unknown*	CONJ
$p \wedge t \implies \neg q$	$\{p, t\}$	q	*False*	CONJ
$p \wedge t \implies \neg q$	$\{p, \neg t\}$	q	*Unknown*	CONJ+NEG
$p \wedge t \implies q$	$\{p\}$	q	*Unknown*	CONJ+NEG
$p \wedge t \implies q$	$\{p, t\}$	q	*True*	CONJ+NEG
$p \wedge t \implies q$	$\{p, \neg t\}$	q	*Unknown*	CONJ+NEG

Table 9: **Conjunction Contrast Perturbations**. These are perturbations for testing conjunction and negation reasoning abilities. First row is the base theory being perturbed. Please refer to Appendix D for more details.

Modified Rule	Facts	Statement	Label	Group
$p \implies q$	$\{p\}$	q	*False*	BASE
$p \vee t \implies \neg q$	$\{p\}$	q	*False*	DISJ
$p \vee t \implies \neg q$	$\{p, t\}$	q	*False*	DISJ
$p \vee t \implies \neg q$	$\{\neg p, \neg t\}$	q	*Unknown*	DISJ+NEG
$p \vee t \implies q$	$\{p\}$	q	*True*	DISJ+NEG
$p \vee t \implies q$	$\{p, t\}$	q	*True*	DISJ+NEG
$p \vee t \implies q$	$\{\neg p, \neg t\}$	q	*Unknown*	DISJ+NEG

Table 10: **Disjunction Contrast Perturbations**. These are perturbations for testing disjunction and negation reasoning abilities. First row is the base theory being perturbed. Please refer to Appendix D for more details.

E Logical Contrast set breakdown

In this section, we further discuss the performance of the LMs on each group of the Logical Contrast set. From Tables 11 and 12 we can say that the models generally perform worse when they need to handle more complicated compound rules (CONJ + NEG > CONJ > BASE (where > means harder)). Additionally, we find that when we add more compound rules in the training dataset, the performance is generally better, except for OR + NEG. From Table 6, we can say that OR is harder to learn no matter from scratch or from a pre-trained checkpoint. So, when training on OR + NEG, instead of using AND, the model performs worse since it cannot figure out the semantics of AND using the OR dataset. And that's why OR + NEG always perform worse than NEG in conjunction dataset or performs similarly in disjunction dataset. Also, we observe that models trained with AND perform better on conjunction and models trained with OR perform better on disjunction. Lastly, we find that the models are worse at handling disjunction than conjunction theories. Overall, it indicates that these models still do not learn the semantics of logic from language.

F Result breakdown by label

From Tables 13 and 14, we find that the model gets good results when trained on NO LOGIC OP dataset and that is likely because the model learns to predict *Unknown* more frequently. When the model is trained on the other four datasets with more compound rules, the performance are generally similar or slightly improved on the *True* and *False* labels. That means adding more compound rules during training can only help the model up to an extent. The exception in the trend is seen with OR + NEG dataset, for similar reasons as discussed in Appendix E.

Logical Contrast set breakdown	Conjunction			Disjunction		
	BASE	CONJ	CONJ + NEG	BASE	DISJ	DISJ + NEG
NO LOGIC OP	0.50	0.54	0.71	0.50	0.30	0.44
NEG	1.00	0.73	0.50	1.00	0.64	0.34
AND + NEG	1.00	0.86	0.72	1.00	0.56	0.45
OR + NEG	1.00	0.36	0.27	1.00	0.86	0.34
AND + OR + NEG	1.00	0.84	0.78	1.00	0.87	0.44

Table 11: Performance breakdown of RoBERTa-Large with different groups of Logical Contrast set. Please refer to Appendix E for more details.

Logical Contrast set breakdown	Conjunction			Disjunction		
	BASE	CONJ	CONJ + NEG	BASE	DISJ	DISJ + NEG
NO LOGIC OP	0.47	0.49	0.67	0.48	0.23	0.44
NEG	1.00	0.78	0.57	1.00	0.58	0.34
AND + NEG	1.00	0.81	0.56	1.00	0.52	0.30
OR + NEG	1.00	0.63	0.42	1.00	0.86	0.32
AND + OR + NEG	1.00	0.83	0.42	1.00	0.88	0.34

Table 12: Performance breakdown of T5-Large with different groups of Logical Contrast set. Please refer to Appendix E for more details.

Training Dataset	Without Distractors						With Distractors					
	Conjunction			Disjunction			Conjunction			Disjunction		
	False	*True*	*Unknown*	*False*	*True*	*Unknown*	*False*	*True*	*Unknown*	*False*	*True*	*Unknown*
NO LOGIC OP	0.00	0.77	0.95	0	0.6	0.95	0.00	0.74	0.96	0	0.55	0.96
NEG	0.77	0.77	0.66	0.62	0.61	0.45	0.76	0.75	0.67	0.58	0.57	0.42
AND + NEG	0.76	0.77	0.88	0.57	0.58	0.74	0.77	0.77	0.78	0.58	0.58	0.48
OR + NEG	0.78	0.79	0.28	0.77	0.78	0.25	0.79	0.78	0.31	0.76	0.75	0.28
AND + OR + NEG	0.75	0.75	0.94	0.77	0.78	0.45	0.75	0.75	0.83	0.75	0.74	0.39

Table 13: Performance breakdown of RoBERTa-Large with different labels for Logical Contrast set. Please refer to Appendix F for more details.

Training Dataset	Contrapositive			Distributive 1			Distributive 2		
	False	*True*	*Unknown*	*False*	*True*	*Unknown*	*False*	*True*	*Unknown*
NO LOGIC OP	0.00	0.70	0.98	0.00	1.00	-	0.00	1.00	-
NEG	0.66	0.68	0.97	1.00	1.00	-	1.00	1.00	-
AND + NEG	0.68	0.68	0.99	0.98	0.98	-	1.00	1.00	-
OR + NEG	0.75	0.75	0.94	1.00	1.00	-	1.00	1.00	-
AND + OR + NEG	0.73	0.73	1.00	1.00	1.00	-	1.00	1.00	-

Table 14: Performance breakdown of RoBERTa-Large with different labels for Logical Equivalence set. Please refer to Appendix F for more details.