Flexible Generation of Natural Language Deductions

Kaj Bostrom **Xinyu Zhao** **Swarat Chaudhuri** **Greg Durrett**
Department of Computer Science
The University of Texas at Austin
kaj@cs.utexas.edu

Abstract

An interpretable system for open-domain reasoning needs to express its reasoning process in a transparent form. Natural language is an attractive representation for this purpose — it is both highly expressive and easy for humans to understand. However, manipulating natural language statements in logically consistent ways is hard: models must cope with variation in how meaning is expressed while remaining precise. In this paper, we describe PARAPATTERN, a method for building models to generate deductive inferences from diverse natural language inputs without direct human supervision. We train BART-based models (Lewis et al., 2020) to generate the result of applying a particular logical operation to one or more premise statements. Crucially, we develop a largely automated pipeline for constructing suitable training examples from Wikipedia. We evaluate our models using out-of-domain sentence compositions from the QASC (Khot et al., 2020) and EntailmentBank (Dalvi et al., 2021) datasets as well as targeted perturbation sets. Our results show that our models are substantially more accurate and flexible than baseline systems. PARAPATTERN achieves 85% validity on examples of the 'substitution' operation from EntailmentBank without the use of any in-domain training data, matching the performance of a model fine-tuned for EntailmentBank. The full source code for our method is publicly available.[1].

1 Introduction

Developing models that can make useful inferences from natural language premises has been a core goal in artificial intelligence since the field's early days (Bobrow, 1964; Winograd, 1971). Since then, there has been massive progress in automated formal reasoning (De Moura and Bjørner, 2011); in contrast, progress in automated natural language reasoning has been comparatively slow. Today, 'natural language inference' usually means recognizing textual entailment (RTE), a pairwise sentence classification task. Models have saturated RTE benchmarks (Bowman et al., 2015; Williams et al., 2018) largely through surface-level heuristics (Gururangan et al., 2018; Poliak et al., 2018); hill-climbing on these benchmarks has failed to yield robust models (Naik et al., 2018) or systems capable of more complex reasoning.

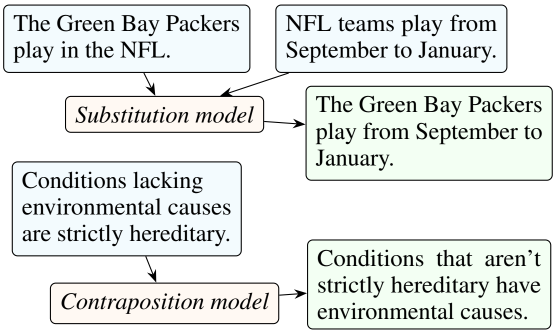

The Green Bay Packers play in the NFL.
NFL teams play from September to January.
Substitution model
The Green Bay Packers play from September to January.
Conditions lacking environmental causes are strictly hereditary.
Contraposition model
Conditions that aren't strictly hereditary have environmental causes.

Figure 1: Examples of the natural deduction operations for which we construct models. Note that conclusions involve both lexical inferences (X *plays in the NFL* $\rightarrow$ X is an *NFL team*, $\neg[X$ *lacks* $Y]$ $\rightarrow$ X *has* Y) and logical transformations.

Following a line of work on multi-hop question answering (Welbl et al., 2018; Yang et al., 2018; Chen and Durrett, 2019; Min et al., 2019), the reading comprehension community has started to make inroads in the area of reasoning. Recent datasets have been explicitly designed to test deduction ability (Liu et al., 2020; Yu et al., 2020; Holzenberger et al., 2020) and new types of models take inspiration from formal and informal reasoning (Clark et al., 2020; Saha et al., 2020; Cartuyvels et al., 2020; Betz et al., 2021). Many recent modeling efforts share a common motif of using intermediate fact chains to support their final predictions, but a major shortcoming is that these chains are either retrieved heuristically or

[1] https://github.com/alephic/ParaPattern

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6266–6278
November 7–11, 2021. ©2021 Association for Computational Linguistics

generated freely from autoregressive language models, meaning they are not necessarily sound. To enforce soundness, we envision future reasoning systems factoring the deduction process into a set of common operations, analogous to proof rules. Modeling the reasoning process in this way would grant the ability to generalize systematically to any problem that could be decomposed in terms of available operations, among other desirable properties (Rudin, 2019).

In this work, we describe a *generative* model for single-step deductive reasoning, building towards models capable of generating the range of logical transformations needed for the full reasoning process. We use a BART-based sequence-to-sequence model (Lewis et al., 2020) to represent the distribution of valid conclusion statements conditioned on one or more premise statements. To make sound inferences, the model must be fine-tuned on well-formed training data. We describe a pipeline for crafting this data based on syntactic retrieval from Wikipedia, rule-based example construction, and automatic paraphrasing to increase diversity. Our hypothesis is that the logical regularities in the constructed examples will teach models to generate correct deductions, while paraphrasing coupled with the inductive bias from pretraining will regularize models, allowing them to robustly tolerate natural lexical and syntactic variation in their inputs.

We demonstrate our method's effectiveness by using it to create models for two distinct logical operations, *substitution* and *contraposition*, examples of which are shown in Figure 1. Through experiments on manually-constructed English perturbation sets, as well as on the English Question Answering via Sentence Composition (QASC) and EntailmentBank datasets (Khot et al., 2020; Dalvi et al., 2021), we show that our proposed data generation method leads to accurate and robust operation models. While baseline methods tend to default to trivial input copying and fail to assign significant likelihood to valid novel conclusions, we show that our operation models reliably generate consistent inferences. Evaluating our substitution model on fact compositions from the QASC and EntailmentBank datasets, we find that our method produces valid conclusions at rates equivalent to models trained on in-domain, human-annotated data, indicating that our method is a viable substitute for expensive direct supervision.

2 Methods

We consider an operation G, like our substitution example (Fig. 1), to be analogous to a proof rule, allowing one or more premise statements to be combined and transformed to yield a valid conclusion statement. A model for G places a distribution $p_G(y \mid x_0, \ldots, x_n)$ over conclusions y conditioned on premises x_i.

We would like models to satisfy three criteria:

Consistency: Predicted outputs should be valid deductions from the model's inputs.

Robustness: Models should be robust to linguistic variation present in their inputs.

Supervision economy: A minimal amount of manual effort should be needed to construct a model for a new operation.

We choose to parameterize p_G by fine-tuning pretrained sequence-to-sequence language models (Lewis et al., 2020; Raffel et al., 2020). Fine-tuning pretrained models allows the resulting operations to successfully handle a wider variety of inputs by leveraging general linguistic knowledge gained during pretraining.

The three desired model criteria we have identified lead to two data collection balancing acts:

- Model consistency and robustness improve with increased data quantity, quality, and diversity, but collecting a large amount of diverse, high-quality data presents a challenge.
- Variation in the data and even noise will improve model robustness, but too much noise will cause the trained model to be inconsistent.

Directly annotating such data is possible, but requires significant manual labor, either in the form of expert annotation or careful prompting and filtering of crowd annotations. While annotated resources already exist for certain domains (Khot et al., 2020; Hwang et al., 2021), this is not the case for most types of reasoning. Scraping data from free text only works if examples of the desired operation appear in the wild, which is generally not the case for concise well-formed deduction steps. Betz et al. (2020) use templates to generate logically consistent text for training language models; however, there is little need for diversity or naturalism in their data as it is exclusively used

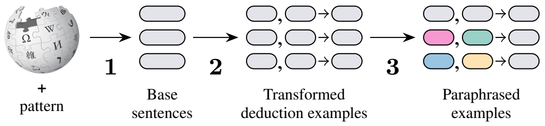


Figure 2: Schematic overview of the three phases of our data collection process: retrieval of base sentences from Wikipedia, expansion of these into reasoning examples, and paraphrasing.

during pretraining for the purposes of transfer learning. Tafjord et al. (2021) use template-based natural language proofs to fine-tune transformer language models for reasoning; we include a model trained on their data as one of our baseline systems.

2.1 Data Collection

Our proposed method, ParaPattern, combines scraping, template-based generation, and automatic paraphrasing in order to achieve sufficient data diversity and quality with very little manual effort.

ParaPattern consists of three phases, as shown in Figures 2 and 3.

Phase 1: Source scraping A set of dependency patterns is used to retrieve source sentences suitable for template expansion from a dependency-parsed free text corpus. An example of one of the dependency patterns we use is shown in Figure 3. This template finds sentences exhibiting the Hearst pattern (Hearst, 1992) *X such as Y* indicating a hypernymy relationship between *X* and *Y*. Note that the retrieved sentences do not constitute complete training examples; such examples of logical reasoning are hard to find in the wild. These sentences need to be reshaped in the next step, but they are *lexically diverse* and *semantically suitable* as inputs to our templates in terms of the relations they express.

We perform syntactic search over a corpus of cleaned English Wikipedia article text comprising 112M sentences. We use the off-the-shelf spaCy `en-core-web-sm` dependency parser (Honnibal et al., 2020), and index the resulting trees by bottom-up dependency chain prefixes in chunks of 160K sentences in order to accelerate the search process. Dependency parsing and index construction for English Wikipedia takes approximately 24 hours on a single CPU core.

We use six pattern variations to gather source sentences for the substitution template and two patterns for the contraposition template. Potential matches are filtered based on a list of disallowed subject modifiers that would result in semantically invalid examples. After filtering, the substitution patterns yield ∼44,000 source sentences and the contraposition patterns yield ∼23,000 source sentences. All dependency patterns we use are listed in Figure 7 in the appendix. Dependency search over the indexed trees takes 30-45 minutes depending on pattern complexity.

Phase 2: Template expansion Source sentences are expanded into generated examples through the application of an operation-specific template. Figure 3 shows an example of a source sentence and its rule-based expansion into a pair of premise statements and a conclusion.

Template outputs are expressed in terms of the source pattern's match variables. The template expansion algorithm produces examples by breaking out dependency subtrees rooted at each match variable and rearranging them according to the template structure. We also apply simple heuristics for verb reinflection and noun number adjustment during the reconstruction process to maximize the fluency of the resulting text.

Phase 3: Paraphrase augmentation Data from template expansion is augmented by adding copies of each example with paraphrased input sentences. Paraphrases are generated using a version of the PEGASUS model (Zhang et al., 2020) fine-tuned for paraphrasing.[2] We sample two paraphrases for each original input using nucleus sampling with $p = 0.9$. See Figure 4 for samples of input sentences after paraphrasing has been applied. These values were determined heuristically in the course of our preliminary experiments; we found that using a higher sampling cutoff or more paraphrases critically reduced the consistency of model inferences, and lowering p or using only a single paraphrase per source example increased the rate of premise copying for examples not matching a training template.

We observe that the resulting paraphrases tend to include a noticeable amount of noise (e.g. the replacement of 'Hibiscus' with 'bing' in Figure 3), but we hypothesize that since we only paraphrase premises, this effectively adds a denoising component to the fine-tuning objective similar to the motivation behind backtranslation

[2] Model weights from https://huggingface.co/tuner007/pegasus_paraphrase

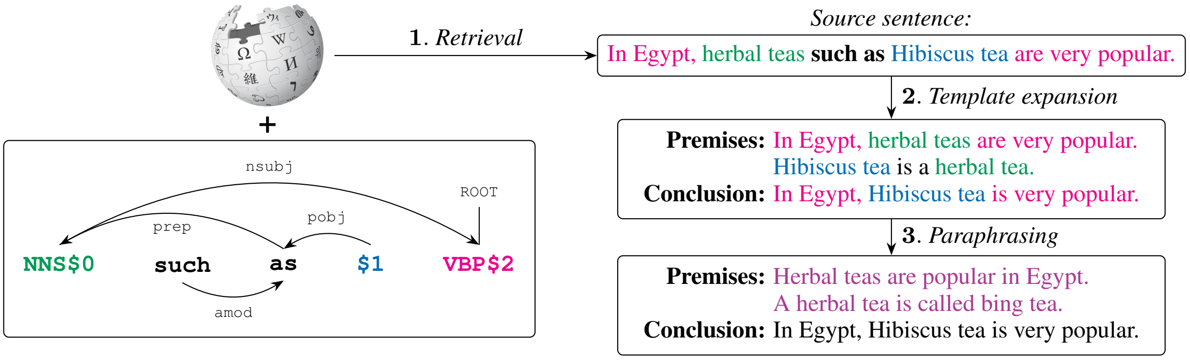

1. Retrieval
Source sentence:
In Egypt, herbal teas such as Hibiscus tea are very popular.
2. Template expansion
Premises: In Egypt, herbal teas are very popular.
Hibiscus tea is a herbal tea.
Conclusion: In Egypt, Hibiscus tea is very popular.
3. Paraphrasing
Premises: Herbal teas are popular in Egypt.
A herbal tea is called bing tea.
Conclusion: In Egypt, Hibiscus tea is very popular.
+
nsubj
ROOT
prep
pobj
NNS$0
such
as
$1
VBP$2
amod

Figure 3: An example of the steps involved in our data generation process for the substitution operation. Phrases in the source sentence and expanded template are colored according to their corresponding pattern variable.

Substitution

Premises: Staphylococcus epidermis is a microorganism.
Microorganisms colonize the skin surface.
Paraphrased: Staphylococcus epidermidis is a microorganism.
Microbiological colonization of the skin surface.
“Staphylococcus Epidermidis is a Microorganism.”
The skin surface is colonized by micro organisms.
Conclusion: Staphylococcus epidermis colonizes the skin surface.

Premises: During the undergraduate years, seminarians learn the ancient language courses.
Latin is an ancient language course.
Paraphrased: The seminars know the ancient language courses.
Latin is an old language course.
Seminarians learn ancient language during their undergraduate years.
Latin is a language.
Conclusion: During the undergraduate years, seminarians learn Latin.

Contraposition

Premise: As such, rivers that have headwaters in the mountains provide water for irrigation in the surrounding lands.
Paraphrased: In order for water to be used in the surrounding lands, the rivers in the mountains must have their headwaters there.
Conclusion: As such, rivers that do not provide water for irrigation in the surrounding lands do not have headwaters in the mountains.

Premise: Dogs that are especially dirty or hungry are not able to participate in contests.
Paraphrased: To participate in a contest, dogs that are dirty or hungry, must be turned away.
Conclusion: Dogs that are able to participate in contests are not especially dirty or hungry.

Figure 4: Examples produced by our data generation pipeline.

in machine translation (Sennrich et al., 2016). Additional samples of the output of our data generation pipeline are shown in Figure 4. These examples demonstrate the ability of automatic paraphrasing to reduce both lexical and syntactic regularities in the original template outputs that can lead models to overfit to the template. In our subsequent experiments, we examine this overfitting by ablating Phase 3 of our data generation pipeline.

2.2 Model Training

Once data for an operation has been generated, we use it to fine-tune an instance of BART-Large (Lewis et al., 2020). Premise sentences $x_0 \ldots x_n$ are concatenated in a random order and provided as input to the model's encoder, and the conclusion sentence y is used as the target sequence for the decoder.

We use model and training algorithm implementations from the `transformers` library (Wolf et al., 2020). We fine-tune models for a single epoch using the AdamW optimizer (Loshchilov and Hutter, 2019) with initial learning rate 3e-5 and triangular learning rate decay. In our preliminary experiments, we found that fine-tuning models for more than a single epoch always produced detrimental overfitting. Models are trained using a total batch size of 16 split across two NVidia Titan RTX GPUs; with this configuration, training takes no more than an hour of wall clock time per model.

3 Experiments

3.1 Baselines

We compare models trained using our proposed method against three baselines.

Our first baseline system is an unmodified instance of the pretrained autoregressive GPT2-Large language model (Radford et al., 2019), prompted with operation premises followed by the elicitation prefix “Therefore,” (**Zero-shot GPT2**). This baseline, inspired by the zero-shot premise elaboration employed by Betz et al. (2021), is intended to assess the likelihood of making consistent deductions under a general model of language with no logical specialization.

Our second baseline model is an instance of BART-Large fine-tuned to generate hypotheses from the MNLI dataset (Williams et al., 2018)

Substitution - Control

Premises: RSA is a cryptographic system.
Cryptographic systems let people exchange messages securely.
Conclusion: RSA lets people exchange messages securely.
Predicted: RSA lets people exchange messages securely.

Link NP mismatch

Premises: RSA is a cryptographic system.
Encryption protocols let people exchange messages securely.
Conclusion: RSA lets people exchange messages securely.
Predicted: RSA allows people to exchange messages securely.

Identity VP mismatch

Premises: Dominant cryptographic systems include RSA.
Cryptographic systems let people exchange messages securely.
Conclusion: RSA lets people exchange messages securely.
Predicted: RSA allows people to exchange messages securely.

NP + VP mismatch

Premises: Dominant encryption protocols include RSA.
Cryptographic systems let people exchange messages securely.
Conclusion: RSA lets people exchange messages securely.
Predicted: RSA allows people to exchange messages securely.

Number agreement

Premises: RSA is a cryptographic system.
Cryptographic systems shield web traffic from surveillance and let people communicate securely.
Conclusion: RSA shields web traffic from surveillance and lets people communicate securely.
Predicted: RSA shields web traffic from surveillance and let people communicate securely.

Contraposition - Control

Premise: Pesticides that contain DDT have harmful effects on birds.
Conclusion: Pesticides that do not have harmful effects on birds do not contain DDT.
Predicted: Pesticides that do not have harmful effects on birds do not contain DDT.

Postnominal modifier mismatch

Premise: Pesticides containing DDT have harmful effects on birds.
Conclusion: Pesticides that do not have harmful effects on birds do not contain DDT.
Predicted: Pesticides that do not have harmful effects on birds do not contain DDT.

Prenominal modifier mismatch

Premise: DDT-containing pesticides have harmful effects on birds.
Conclusion: Pesticides that do not have harmful effects on birds do not contain DDT.
Predicted: Pesticides that do not have harmful effects on birds do not contain DDT.

Premise negation

Premise: Pesticides that contain DDT aren't safe for birds.
Conclusion: Pesticides that are safe for birds do not contain DDT.
Predicted: Pesticides that are safe for birds do not contain DDT.

Figure 5: Aligned perturbation set examples for substitution (left) and contraposition (right), with corresponding predicted PARAPATTERN BART output samples. Perturbed portions of each example are shown in turquoise. Grammatical errors are shown in orange.

conditioned on their respective premises (**MNLI BART**). We train on all instances for which the gold label indicates entailment (≈103K examples) with the same training configuration as our other models, detailed in 2.2. We hypothesize that while this model may assign higher likelihood to valid conclusions than a general language model would, it will place much more probability mass on re-emitting premise statements due to the fact that high word overlap tends to be a common feature of RTE examples labeled as 'entailment' (Zhou and Bansal, 2020).

Our third baseline model is an instance of BART-Large fine-tuned to generate proof steps from the ProofWriter dataset (Tafjord et al., 2021) (**ProofWriter BART**). While this dataset contains a large number of English proof steps (≈135K), the language used in its proofs is automatically generated from a limited template library and is thus highly constrained. We hypothesize that this model will be unable to generalize as a result.

On the QASC and EntailmentBank datasets, we additionally compare to BART-Large models fine-tuned on inference steps from each dataset's respective training split (**QASC BART** and **Ent. Bank BART**). The QASC training set contains ∼8K crowd-annotated fact compositions, while the EntailmentBank training set contains ∼3K expert-annotated premise-conclusion steps.

3.2 Perturbation sets

First, in order to evaluate the accuracy of our models on the operations they were designed for, and to understand the degree to which they generalize when input statements deviate from their training patterns, we manually construct controlled perturbation sets for each operation.

Our substitution perturbation set consists of 75 examples evenly split across a control condition and four test conditions, and our contraposition perturbation set consists of 60 examples evenly split across a control condition and three test conditions (15 examples per condition).

Each example in a given test condition is constructed by perturbing a corresponding control example. This aligned structure allows us to evaluate the impact of a particular perturbation on model performance without the confounding effect of content variation that would be present if each condition were constructed independently. Samples of each perturbation condition are presented in Figure 5.

3.3 QASC and EntailmentBank

Our perturbation sets are not necessarily "in-domain" for our models, but they still neatly fit the reasoning patterns we are targeting. To test our approach's applicability to data outside its training, we apply our substitution model to the

	Substitution			Contraposition		
Model	Ref. PPL ↓	BLEURT ↑	Valid% ↑	Ref. PPL ↓	BLEURT ↑	Valid% ↑
Zero-shot GPT2	3.52	-0.88 ± 0.35	1	6.04	-0.89 ± 0.31	1
MNLI BART	2.00	-0.06 ± 0.13	6	4.50	-0.16 ± 0.04	2
ProofWriter BART	3.86e2	-1.18 ± 0.14	4	5.83e3	-1.39 ± 0.12	0
Pattern-only BART	3.55	0.49 ± 0.01	55	3.16	0.31 ± 0.00	38
ParaPattern BART	**1.54**	**0.66 ± 0.05**	**87**	**1.57**	**0.69 ± 0.07**	**80**

Table 1: Results for each perturbation set, averaged across perturbation conditions. Ref. PPL refers to the perplexity of the reference conclusion under the model distribution. BLEURT scores are averaged across 10 samples per example; ± indicates the standard deviation of the samples. Valid% refers to the proportion of generated conclusions rated as valid and non-redundant following manual review. Separate results for each perturbation condition can be found in Table 4 in the appendix.

fact compositions in the validation splits of the QASC and EntailmentBank datasets (Khot et al., 2020; Dalvi et al., 2021).

QASC fact compositions were annotated by crowd workers as rationales for multiple-choice question answering problems. Since annotators combined facts with a certain question in mind, there is some amount of missing context for many QASC fact combinations.

The EntailmentBank dataset consists of a set of expert-annotated natural language proofs for elementary science question-answer pairs involving multi-step reasoning. Thanks to its trained annotators, EntailmentBank contains fewer spurious fact combinations than QASC.

3.4 Evaluation Criteria

We evaluate model performance on each dataset primarily through a manual assessment of conclusion validity. The first author placed generated conclusions into one of six categories:

Valid: Conclusion is logically consistent with premises but does not trivially repeat them.

Valid with minor grammar errors: Conclusion is valid but includes minor syntactic errors such as bad verb inflection that do not hinder comprehension.

Repeats premises: Conclusion is a near-verbatim copy of one or more premise sentences.

Unsupported: Conclusion is technically true but does not logically follow from premises.

Incompatible: Conclusion contradicts premises or is inherently false.

Incomprehensible: Conclusion is difficult to interpret due to major ungrammaticality.

Model outputs were shuffled and annotated without knowledge of model identity to prevent rating bias. The last author reannotated a subset of QASC examples to validate the first author's annotations; there were minor differences in interpretation of the divisions between non-valid categories, but the relative proportion of conclusions rated as valid remained consistent between annotators.

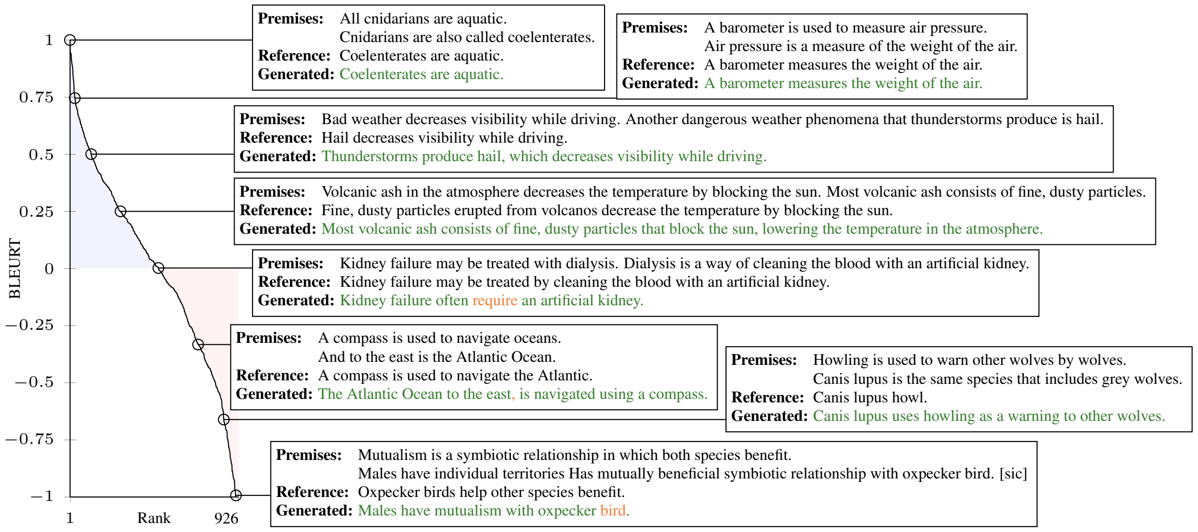
We additionally compute the perplexity of reference conclusions under each model in order to assess the likelihood assigned to desired conclusions by each model's output distribution.

For the perturbation sets, we also report the BLEURT score (Sellam et al., 2020) of generated conclusions with respect to reference conclusions.

4 Results

4.1 Results on Perturbation Sets

Our first question is whether or not we **have good generative models of natural language deductions.** As Table 1 shows, ParaPattern BART outperforms all baselines by a wide margin in terms of the likelihood of desired conclusions (Ref. PPL), the similarity of its outputs to desired conclusions, and its overall rate of valid inference. Additionally, there is a substantial gap in performance between models trained with and without paraphrastic data augmentation (ParaPattern vs. Pattern-only, an ablation of our method). We observe that **ParaPattern allows models to reliably produce valid inferences** when given inputs that lie both on and off the training template manifold. In contrast, models trained using generic entailments (MNLI BART) or purely template-derived inferences (ProofWriter BART) are *almost never* able to produce valid, non-redundant inferences.



Premises: All cnidarians are aquatic. Cnidarians are also called coelenterates.
Reference: Coelenterates are aquatic.
Generated: Coelenterates are aquatic.

Premises: A barometer is used to measure air pressure. Air pressure is a measure of the weight of the air.
Reference: A barometer measures the weight of the air.
Generated: A barometer measures the weight of the air.

Premises: Bad weather decreases visibility while driving. Another dangerous weather phenomena that thunderstorms produce is hail.
Reference: Hail decreases visibility while driving.
Generated: Thunderstorms produce hail, which decreases visibility while driving.

Premises: Volcanic ash in the atmosphere decreases the temperature by blocking the sun. Most volcanic ash consists of fine, dusty particles.
Reference: Fine, dusty particles erupted from volcanos decrease the temperature by blocking the sun.
Generated: Most volcanic ash consists of fine, dusty particles that block the sun, lowering the temperature in the atmosphere.

Premises: Kidney failure may be treated with dialysis. Dialysis is a way of cleaning the blood with an artificial kidney.
Reference: Kidney failure may be treated by cleaning the blood with an artificial kidney.
Generated: Kidney failure often require an artificial kidney.

Premises: A compass is used to navigate oceans. And to the east is the Atlantic Ocean.
Reference: A compass is used to navigate the Atlantic.
Generated: The Atlantic Ocean to the east, is navigated using a compass.

Premises: Howling is used to warn other wolves by wolves. Canis lupus is the same species that includes grey wolves.
Reference: Canis lupus howl.
Generated: Canis lupus uses howling as a warning to other wolves.

Premises: Mutualism is a symbiotic relationship in which both species benefit. Males have individual territories Has mutually beneficial symbiotic relationship with oxpecker bird. [sic]
Reference: Oxpecker birds help other species benefit.
Generated: Males have mutualism with oxpecker bird.

Figure 6: BLEURT score profile of ParaPattern BART substitution outputs for fact combinations from the QASC development set. Sampled substitution model outputs and corresponding QASC annotations for a range of scores are shown to the right. Minor grammatical errors are indicated in orange. Note that generated conclusions remain semantically coherent despite diverging from annotated references as BLEURT scores decrease.

Model	Ref. PPL↓	QASC Valid%↑	QASC Gram.%↑
Zero-shot GPT2	7.03	0	0
MNLI BART	**3.83**	8	7
ProofWriter BART	1.69e2	7	1
Ent. Bank BART	6.61	**72**	62
Pattern-only BART	39.7	16	10
PARAPAT. BART	4.82	**73**	**68**
QASC BART	2.71	77	69

Table 2: Results on the QASC development set. Valid% indicates the proportion of predictions for 100 uniformly sampled examples that were rated as valid, non-redundant inferences following manual review. Gram.% indicates the proportion of predictions rated both valid and free of grammatical errors.

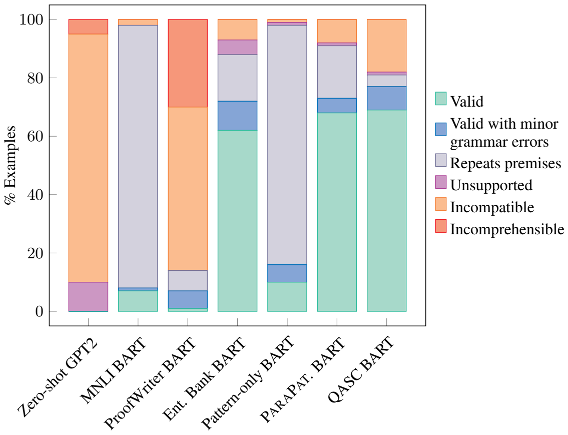


Figure 7: Detailed results of manual evaluation of QASC inferences for each model.

4.2 Results on QASC

Table 2 shows that PARAPATTERN BART generates valid, grammatical inferences at a rate comparable to that of a model with identical parameter budget and pretraining fine-tuned on in-domain data (QASC BART) as well as a model fine-tuned on inferences from EntailmentBank (Ent. Bank BART), an expert-annotated dataset in an adjacent domain. Thanks to our automated data collection pipeline, we are able to achieve this level of fidelity without any direct annotation.

For a full profile of model behaviors on QASC, refer to Figure 7. We note that MNLI BART's preference for repeating premises results in a lower reference perplexity than PARAPATTERN BART in Table 2 despite MNLI BART behaving poorly during generation due to the fact that reference conclusions exhibit high lexical overlap with premises. Pattern-only BART also tends to repeat inputs, but in this case it is a low-confidence 'fallback' behavior, as evidenced by its high reference perplexity.

Visualizing Model Generations on QASC Figure 6 depicts a range of PARAPATTERN BART outputs for QASC validation set fact combinations ranked according to their BLEURT scores with respect to the reference combined fact. In the portion of this distribution above 0 BLEURT, we see very close agreement between the content of generated outputs and references. On the

Model	EntailmentBank		
	Ref. PPL↓	Valid% (All)↑	Valid% (Subst.)↑
ParaPat. BART	4.70	57	85
Ent. Bank BART	3.37	69	85

Table 3: Results on the EntailmentBank development set. Valid% (All) indicates the proportion of predictions for 100 uniformly sampled examples manually rated as valid inferences. Valid% (Subst.) indicates the proportion of valid predictions for the subset of examples classified as "substitution", as defined in Dalvi et al. (2021). This set includes 41% of the sampled examples, in agreement with the statistic reported by the dataset's authors. We omit out-of-domain baselines due to their non-competitive performance on QASC.

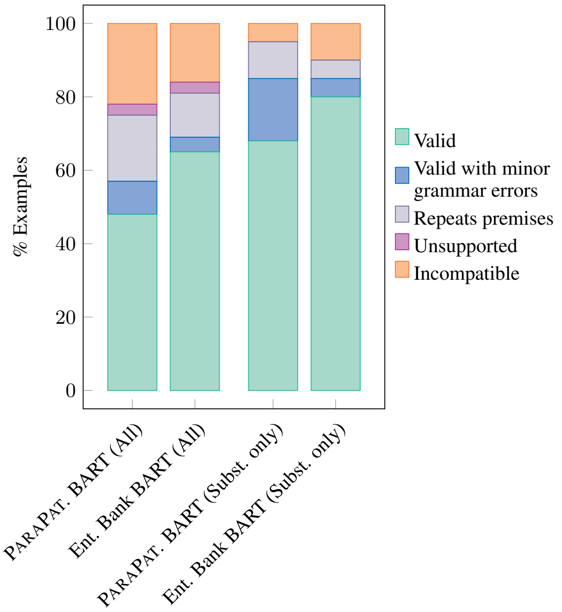


Figure 8: Detailed results of manual evaluation of EntailmentBank inferences for our proposed method (ParaPat. BART) and an in-domain fine-tuned model (Ent. Bank BART).

opposite end of the spectrum, we can see the structure of model outputs diverges from that of the reference fact combinations. However, even as our model's predictions grow farther from the reference conclusions, they remain semantically consistent combinations of the premise facts. The prediction for the final example in Figure 6 is a valid inference in spite of an ungrammatical premise, exemplifying one of the benefits of training on data augmented with noisily paraphrased inputs. In agreement with our quantitative results, these outputs confirm that **ParaPattern generates sound inferences even under domain shift**.

4.3 Results on EntailmentBank

We capitalize on the known taxonomy of reasoning types present in EntailmentBank to better understand how well our substitution model aligns with the definition of 'substitution' adopted by the dataset's authors. In Table 3 we demonstrate that ParaPattern BART matches the performance of the in-domain fine-tuned BART model on the subset of examples where the inputs specify a well-formed substitution as defined in Dalvi et al. (2021). This indicates that there is agreement between their definition of substitution, our model's representation of the operation, and the aspects of the operation captured by the EntailmentBank training set.

Furthermore, this result shows that **ParaPattern can circumvent the need for manual supervision for a given reasoning skill** without sacrificing performance in that skill.

Figure 8 provides a breakdown of model behaviors on EntailmentBank. Of the 22% of *non-substitution* examples for which our substitution model is able to produce valid inferences, we note that the majority are instances of 'property inheritance', 'sequential inference', or 'inference from rule', according to the taxonomy of Dalvi et al. (2021). The following are examples of some of these inferences generated by the ParaPattern BART substitution model:

Inference from rule

If fossils are destroyed in rock transition, then there will be gaps in the fossil record. Fossils are lost / destroyed when sedimentary rock changes to metamorphic rock.
→ *Fossils are lost / destroyed when sedimentary rock changes to metamorphic rock, leaving gaps in the fossil record.*

Property inheritance

United states is located in the northern hemisphere. New york / new york state is a state located in the united states of america.
→ *New york / new york state is in the northern hemisphere.*

Sequential inference

Humans changing ecosystems usually has a negative impact on an ecosystem / organisms living in an ecosystem. Humans building homes in an ecosystem causes that ecosystem to change.
→ *Humans building homes in an ecosystem*

> *usually has a negative impact on an ecosystem / organisms living in an ecosystem.*

We hypothesize that these inferences reflect generalizations of NP substitution to other phrase categories, most likely learned as a side effect of paraphrastic data augmentation.

5 Related Work

Natural Logic (Bernardi, 2002; Zamansky et al., 2006; MacCartney and Manning, 2009; Angeli et al., 2016) is related to our approach in that it provides a framework for logical reasoning about statements in natural language. Such systems recognize that *there is a cat on the dresser* entails *there is an animal on the dresser* because of the hypernymy relationship between *cat* and *animal*. These relationships can be formalized into a monotonicity calculus (Icard et al., 2017) and past work has grounded lexical inference tasks into such a formalism (Angeli et al., 2016; Hu et al., 2020). Instead of decomposing entailment into relationships between words, our models learn to map premises to conclusions at the sentence level, allowing our approach to handle relationships not captured by such a formalism.

Multi-Hop Reasoning Combining facts to form a conclusion overlaps with the idea of multi-hop reasoning, which has been explored in reading comprehension settings (Welbl et al., 2018; Yang et al., 2018). However, training end-to-end models on these datasets does not necessarily teach models to combine facts (Chen and Durrett, 2019; Min et al., 2019). Systems like NLProlog (Weber et al., 2019) attempt to explicitly ground reasoning in logic, but this process still heavily relies on latent representations; in contrast, by grounding reasoning directly in natural language, a system based on natural deduction operations like ours gains inherent faithful natural language explanations and can build on the strengths of pretrained language models.

More recent datasets emphasize the ability to actually exhibit correct reasoning chains and form explanations (Clark et al., 2020; Xie et al., 2020; Dalvi et al., 2021). Systems like PRover (Saha et al., 2020) and Leap-of-Thought (Talmor et al., 2020) have some broadly similar goals to ours, but only *retrieve* facts and do not generate novel conclusions.

Generative Reasoning The generative nature of our models resembles generative models used for commonsense inference (Rajani et al., 2019; Latcinnik and Berant, 2020; Shwartz et al., 2020). However, these models do not strongly constrain the nature of what is generated. In contrast, our models reliably perform specific logical transformations, indicating that they can support sound inferences over longer reasoning chains in future work. Arabshahi et al. (2021) also explore generative reasoning in commonsense scenarios, but the domain of their approach is limited. Khot et al. (2021) use generative models to decompose a complex QA problem into a series of elementary steps that can be delegated to simpler models; this idea parallels the notion of decomposing reasoning into simple steps to be performed by generative operation models. Their results support the idea that such decomposition aids systematic generalization by enforcing separation of concerns.

6 Conclusion

Building systems that use natural language as a medium for reasoning will require operations to logically combine and transform natural language statements. In this work, we present ParaPattern, a method for creating such models with minimal manual effort by fine-tuning pretrained sequence-to-sequence language models on data generated through a three-step process of syntactic retrieval, template expansion, and automatic paraphrasing. Our experimental results show that ParaPattern yields operation models capable of generating consistent logical transformations over a diverse range of natural language inputs, matching the performance of models trained with in-domain human supervision.

Acknowledgments

Thanks to Peter Clark for valuable discussion and to the anonymous reviewers for their helpful comments. This work was partially supported by NSF awards # IIS-1814522 and # CCF-1918651, by DARPA KAIROS award # FA8750-19-2-1003, and by ARO award # W911NF-21-1-0009.

References

Gabor Angeli, Neha Nayak, and Christopher D. Manning. 2016. Combining natural logic and shallow reasoning for question answering. In *Proceedings of the 54th Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), pages 442–452, Berlin, Germany. Association for Computational Linguistics.

Forough Arabshahi, Jennifer Lee, Mikayla Gawarecki, Kathryn Mazaitis, Amos Azaria, and Tom Mitchell. 2021. Conversational neuro-symbolic commonsense reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6):4902–4911.

Raffaella Bernardi. 2002. *Reasoning with Polarity in Categorial Type Logic*. Ph.D. thesis, University of Utrecht.

Gregor Betz, Kyle Richardson, and Christian Voigt. 2021. Thinking Aloud: Dynamic Context Generation Improves Zero-Shot Reasoning Performance of GPT-2. *arXiv*, abs/2103.13033.

Gregor Betz, Christian Voigt, and Kyle Richardson. 2020. Critical thinking for language models. *arXiv*, abs/2009.07185.

Daniel G. Bobrow. 1964. Natural language input for a computer problem solving system. Technical report, Massachusets Institute of Technology.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Ruben Cartuyvels, Graham Spinks, and Marie-Francine Moens. 2020. Autoregressive reasoning over chains of facts with transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6916–6930, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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.

Bhavana Dalvi, Peter A. Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. *arXiv*, abs/2104.08661.

Leonardo De Moura and Nikolaj Bjørner. 2011. Satisfiability modulo theories: Introduction and applications. *Commun. ACM*, 54(9):69–77.

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.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. 2020. A dataset for statutory reasoning in tax law entailment and question answering. In *Proceedings of the 2020 Natural Legal Language Processing (NLLP) Workshop, 24 August 2020, San Diego, US*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Hai Hu, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S. Moss, and Sandra Kuebler. 2020. MonaLog: a lightweight system for natural language inference based on monotonicity. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 334–344, New York, New York. Association for Computational Linguistics.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. COMET-ATOMIC 2020: On Symbolic and Neural Commonsense Knowledge Graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):6384–6392.

Thomas Icard, Lawrence Moss, and William Tune. 2017. A monotonicity calculus and its completeness. In *Proceedings of the 15th Meeting on the Mathematics of Language*, pages 75–87, London, UK. Association for Computational Linguistics.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. QASC: A Dataset for Question Answering via Sentence Composition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8082–8090.

Tushar Khot, Daniel Khashabi, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2021. Text modular networks: Learning to decompose tasks in the language of existing models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1264–1279.

Veronica Latcinnik and Jonathan Berant. 2020. Explaining question answering models through text generation. *arXiv*, abs/2004.05569.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Hanmeng Liu, Leyang Cui, Jian Liu, and Yue Zhang. 2020. Natural language inference in context – investigating contextual reasoning over long texts. *arXiv*, abs/2011.04864.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net.

Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *Proceedings of the Eight International Conference on Computational Semantics*, pages 140–156, Tilburg, The Netherlands. Association for Computational Linguistics.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4249–4257, Florence, Italy. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215.

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.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4615–4629, Online. 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.

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. In *Advances in Neural Information Processing Systems*, volume 33, pages 20227–20237. Curran Associates, Inc.

Leon Weber, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, and Tim Rocktäschel. 2019. NLProlog: Reasoning with weak unification for question answering in natural language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6151–6161, Florence, Italy. Association for Computational Linguistics.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Terry Winograd. 1971. Procedures as a representation of data in a computer program for understanding natural language. Technical report, Massachusets Institute of Technology.

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.

Zhengnan Xie, Sebastian Thiem, Jaycie Martin, Elizabeth Wainwright, Steven Marmorstein, and Peter Jansen. 2020. WorldTree v2: A corpus of science-domain structured explanations and inference patterns supporting multi-hop inference. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5456–5473, Marseille, France. European Language Resources Association.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. 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*.

Anna Zamansky, Nissim Francez, and Yoad Winter. 2006. A 'natural logic' inference system using the Lambek calculus. *J. of Logic, Lang. and Inf.*, 15(3):273–295.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 11328–11339.

Xiang Zhou and Mohit Bansal. 2020. Towards robustifying NLI models against lexical dataset biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8759–8771, Online. Association for Computational Linguistics.

A Appendix

Substitution source dependency patterns:

```
[nsubj:NNS$0 <[amod:‘such’ > prep:IN‘as’ < pobj:$1]]> ROOT:VBP$2
[nsubj:NNS$0 < prep:IN‘like’ < pobj:$1]> ROOT:VBP$2
[nsubj:NNS$0 < prep:VBG‘include’ < pobj:$1]> ROOT:VBP$2
ROOT:VBP$2 <[dobj:NNS$0 <[amod:‘such’ > prep:IN‘as’ < pobj:$1]]
ROOT:VBP$2 <[dobj:NNS$0 < prep:IN‘like’ < pobj:$1]
ROOT:VBP$2 <[dobj:NNS$0 < prep:VBG‘include’ < pobj:$1]
```

Contraposition source dependency patterns:

```
[nsubj:NNS$0 <[nsubj:WDT‘that’ > relcl:VBP$1]] > ROOT:VBP$2
[nsubj:NNS$0 <[prep:IN‘with’ < pobj:$1]] > ROOT:VBP$2
```

Figure 9: All syntactic patterns used for data scraping. Pattern heads are written as `arclabel:POS‘lemma’$i`, where `arclabel` constrains the label on the arc to the matching token's head, `POS` constrains the part-of-speech tag of the matching token, and `‘lemma’` constrains the lemmatized form of the matching token. `$i` indicates that a matching token and the subtree under it will be exposed as a match variable for use in template expansion.

	Substitution			Contraposition		
Model	**Ref. PPL** ↓	**BLEURT** ↑	**Valid%** ↑	**Ref. PPL** ↓	**BLEURT** ↑	**Valid%** ↑
		Control			*Control*	
Zero-shot GPT2	3.28	-0.93 ± 0.33	3	5.41	-0.93 ± 0.28	3
MNLI BART	1.79	0.05 ± 0.15	13	3.81	-0.25 ± 0.02	1
ProofWriter BART	1.72e2	-1.22 ± 0.16	10	3.01e3	-1.36 ± 0.12	0
Pattern-only BART	**1.01**	**0.89 ± 0.00**	**100**	**1.01**	**0.90 ± 0.00**	93
ParaPattern BART	1.08	0.85 ± 0.01	96	1.10	**0.89 ± 0.02**	**100**
		Link NP mismatch			*Postnominal modifier mismatch*	
Zero-shot GPT2	3.61	-0.89 ± 0.35	1	6.31	-0.86 ± 0.30	0
MNLI BART	1.91	-0.04 ± 0.07	0	4.79	-0.29 ± 0.02	8
ProofWriter BART	2.46e2	-1.21 ± 0.20	9	3.25e3	-1.42 ± 0.14	0
Pattern-only BART	1.46	**0.70 ± 0.0**	53	2.23	0.00 ± 0.00	0
ParaPattern BART	**1.39**	**0.68 ± 0.05**	**87**	**1.39**	**0.75 ± 0.08**	**87**
		Identity VP mismatch			*Prenominal modifier mismatch*	
Zero-shot GPT2	3.74	-0.87 ± 0.36	2	6.96	-0.87 ± 0.28	0
MNLI BART	2.17	-0.07 ± 0.12	3	6.14	-0.30 ± 0.04	0
ProofWriter BART	2.68e2	-1.25 ± 0.14	0	7.05e3	-1.51 ± 0.12	0
Pattern-only BART	4.39	0.09 ± 0.00	13	7.08	-0.37 ± 0.00	0
ParaPattern BART	**1.59**	**0.52 ± 0.14**	**86**	**1.79**	**0.48 ± 0.15**	**58**
		NP + VP mismatch			*Premise negation*	
Zero-shot GPT2	4.15	-0.89 ± 0.35	0	5.50	-0.87 ± 0.37	3
MNLI BART	2.17	-0.18 ± 0.17	2	3.24	0.20 ± 0.07	0
ProofWriter BART	2.81e2	-1.31 ± 0.08	0	1.00e4	-1.26 ± 0.11	0
Pattern-only BART	8.37	0.00 ± 0.00	7	2.32	**0.70 ± 0.00**	60
ParaPattern BART	**1.71**	**0.46 ± 0.08**	**75**	**2.01**	0.64 ± 0.04	**75**
		Number agreement				
Zero-shot GPT2	2.83	-0.81 ± 0.33	1			
MNLI BART	1.97	-0.03 ± 0.16	11			
ProofWriter BART	9.63e2	-0.93 ± 0.14	0			
Pattern-only BART	2.53	**0.77 ± 0.00**	**100**			
ParaPattern BART	**1.93**	**0.75 ± 0.02**	93			

Table 4: Results for each perturbation set, broken down by test condition. Ref. PPL refers to the perplexity of the reference conclusion under the model distribution. BLEURT scores are averaged across 10 samples per example; ± indicates the standard deviation between samples. Valid% refers to the proportion of generated conclusions rated as valid and non-redundant following manual review.