Flexible Operations for Natural Language Deduction

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 complex, open-domain reasoning needs an interpretable meaning representation. Natural language is an excellent candidate — it is both extremely expressive and easy for humans to understand. However, manipulating natural language statements in logically consistent ways is hard. Models have to be precise, yet robust enough to handle variation in how information is expressed. In this paper, we describe PARAPATTERN, a method for building models to generate logical transformations of diverse natural language inputs without direct human supervision. We use a BART-based model (Lewis et al., 2020) to generate the result of applying a particular logical operation to one or more premise statements. Crucially, we have a largely automated pipeline for scraping and constructing suitable training examples from Wikipedia, which are then paraphrased to give our models the ability to handle lexical variation. We evaluate our models using targeted contrast sets as well as out-of-domain sentence compositions from the QASC dataset (Khot et al., 2020). Our results demonstrate that our operation models are both accurate and flexible.

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 of relying on handwritten rules (Bobrow, 1964; Winograd, 1971). Since then, there has been massive progress in automated formal reasoning (De Moura and Bjørner, 2011); in contrast, progress towards automated natural language reasoning has been comparatively slow. At present, 'natural language inference' is most commonly used to mean recognizing textual entailment (RTE), a pairwise sentence classification task. Models can saturate performance on popular 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.

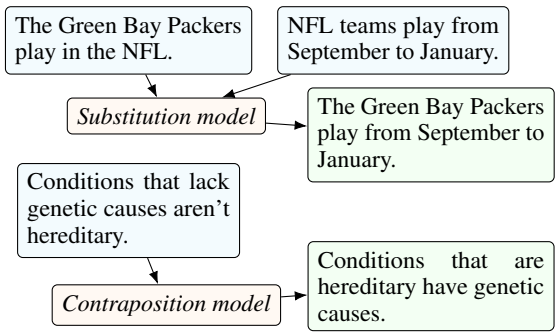

Figure 1: Examples of the natural deduction operations for which we construct models. Note that models must combine lexical inferences (X *plays in the NFL* $\rightarrow$ X is an *NFL team*, $\neg[X$ *lacks* $Y] \rightarrow X$ *has* Y) with logical operations.

Following a thread 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 natural language deduction, with the development of reading comprehension datasets explicitly designed to test deduction ability (Liu et al., 2020; Yu et al., 2020; Holzenberger et al., 2020) and models that take inspiration from formal and informal reasoning (Clark et al., 2020; Saha et al., 2020; Betz et al., 2021; Cartuyvels et al., 2020). Many of these recent modeling efforts share a common motif of using intermediate fact chains to support their final predictions, but these chains are either retrieved heuristically or generated freely from autoregressive language models. Depending on the technique, these kinds of chains are either *unsound*, being connections of facts that lack direct logical connection, or have to be retrieved from a

relatively clean domain-specific corpus where the facts naturally connect. To maintain soundness, we envision future reasoning systems factoring the deduction process into a set of common operations, analogous to proof rules. This guarantees such a system the ability to generalize systematically to any problem that can be decomposed in terms of available operations, among other desirable properties (Rudin, 2018).

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 to compose a full reasoning process. We use a BART-based sequence-to-sequence model (Lewis et al., 2020) to model the distribution of valid conclusion statements conditioned on one or more premise statements. To make sound inferences, the model naturally 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 a pretrained paraphrasing model to introduce lexical variation. Our hypothesis is that the inductive bias of the pretrained model, coupled with the logical regularities of the training examples, will teach the model to generate correct deductions while robustly tolerating lexical variation in the input.

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 contrast sets, as well as on the English QASC dataset (Khot et al., 2020), we show that our proposed data generation method leads to accurate and robust operation models. Outputs from our models judged against references from our contrast sets have BLEURT scores (Sellam et al., 2020) on average 0.79 points higher than equivalent sequence-to-sequence models fine-tuned to generate generic entailments using the Multi-genre Natural Language Inference corpus (Williams et al., 2018). This quantitative gap reflects a stark qualitative difference; whereas our models are able to generate consistent deductions, baseline methods tend to resort to trivial input copying and fail to assign significant likelihood to valid conclusions. When evaluated on out-of-domain fact compositions from the QASC dataset, we observe that even in cases where our model predicts conclusions that differ substantially from the original annotations, the majority of the resulting inferences are valid, demonstrating the flexibility of models trained with our method.

2 Methods

An operation model for operation G places a distribution $p_G(y \mid x_0, \ldots, x_n)$ over output sentences y conditioned on one or more input sentences x_i.

We would like operation models to satisfy the following 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 large amount of manual effort should not 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). Pretrained language models are state-of-the-art building blocks for sequence transduction tasks. 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 this 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., 2020), this is not the case for most types of reasoning. Betz et al. (2020) use templates to generate logically consistent text for training language models; however, in their

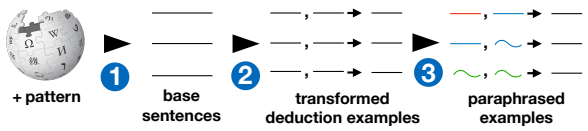

Figure 2: Schematic overview of our data collection process, broken down into three phases: retrieval of sentences from Wikipedia, expanding these into reasoning examples, and paraphrasing.

setting there is little need for diversity or naturalism since it is exclusively used during pretraining for the purposes of transfer learning. Since our aim is to be able to apply our operation models directly, our training data must be naturalistic and diverse. Significant manual effort would be needed to create enough template variants slot-fillers to achieve sufficient diversity purely through templating. 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.

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 Figure 2.

Phase 1: Source Scraping A set of dependency patterns are 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 *Y* and *X*. 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 dump 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. 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 blacklist that disallows 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. A full list of the dependency patterns we use is included in Appendix A.

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 expansion into a pair of premise and conclusion statements.

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 in order 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., 2019) fine-tuned for paraphrasing.[1] We sample two paraphrases for each original input using nucleus sampling with $p = 0.9$. See Figure 3 for samples of input sentences once this paraphrasing is applied.

We observe that the resulting paraphrases tend to introduce a noticeable amount of noise (e.g. substituting 'Hibiscus' for 'bing' in Figure 3), but we hypothesize that since we only paraphrase input sentences, this effectively adds a denoising component to the fine-tuning objective, resulting in more robust models. This is similar to the motivation behind backtranslation in machine translation (Sennrich et al., 2016). Crucially, this input paraphrasing process introduces lexical variation in the inputs that is not necessarily captured by the template matches.

[1]Model available at `https://huggingface.co/tuner007/pegasus_paraphrase`

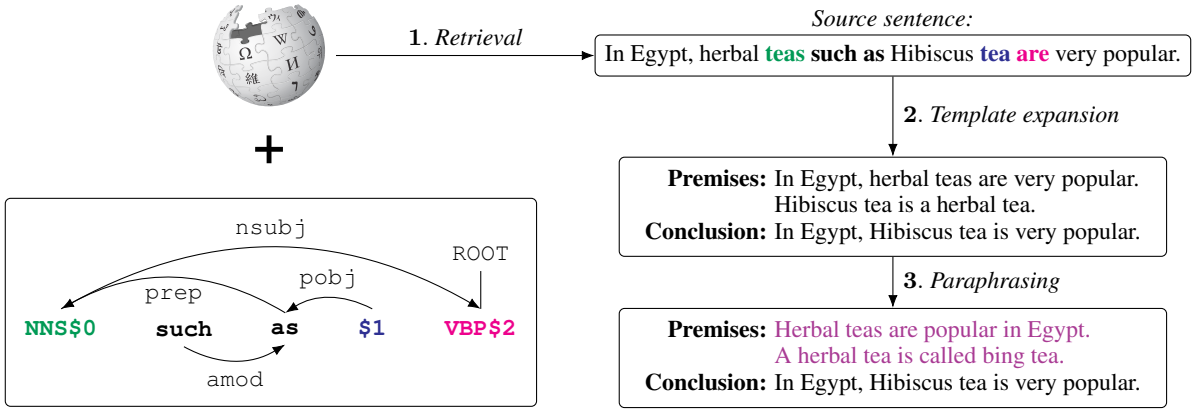

Figure 3: An example of the steps involved in our data generation process for the substitution operation. Words in the source sentence participating in the syntactic pattern match are colored according to the pattern component they align with.

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). 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. 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.

In the interest of reproducibility, the full source code for our data collection and model training pipeline is publicly available[2].

3 Experiments

3.1 Baselines

We compare models trained using our proposed method against two baseline models.

Our first baseline model 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 in 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) 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 place higher likelihood on reference conclusions than a general language model would, it will place even higher likelihood 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).

3.2 Contrast sets

In order to evaluate the accuracy of our models and the degree to which they generalize to input variations that deviate from their training patterns, we manually construct controlled contrast sets for each operation.

Our substitution contrast set consists of 75 examples evenly split across five test conditions (15 examples per test condition), examples of which are presented in Figure 4:

Control: Contains examples that fit the substitution template the model was trained on. We include this condition to verify that a model can apply its pattern to novel sentences correctly.

Link NP mismatch: The NP targeted for substitution is distinct in each premise, but still shares the same meaning.

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

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 4: Aligned contrast set examples for substitution (left) and contraposition (right), with associated PARAPATTERN BART output samples.

Identity VP mismatch: The identity statement is not expressed using a copula.

NP + VP mismatch: This condition combines the perturbations from **Link NP mismatch** and **Identity VP mismatch**.

Number agreement: The target statement contains a verb that must be reinflected to agree with the substituted NP's number, but is not in a position that would be corrected by the training template rules.

Our contraposition contrast set consists of 60 examples evenly split across the following conditions:

Control: Contains examples that fit the contraposition template the model was trained on.

Postnominal modifier mismatch: The subject restriction is expressed using a postnominal modifier not produced by the training template.

Prenominal modifier mismatch: The subject restriction is expressed as a prenominal modifier instead of the postnominal phrases produced by the training template.

Premise negation: The premise contains a syntactic or lexical negation in a position not handled by the training template.

Contrast set examples contain aligned lexical content across test conditions in order to allow us to evaluate generalization under various perturbations while avoiding the confounding effect of lexical variation that would be present if each test condition were constructed to be completely independent. As shown in Figure 4, each example in a given test condition is a perturbation of a control example. The combination of this controlled structure and the wide performance gaps between models gives us confidence in our results despite the small size of these datasets.

3.3 Results on Contrast Sets

We measure model performance on the contrast sets using the perplexity of reference conclusions under the model distribution, the BLEURT score (Sellam et al., 2020) of generated conclusions with respect to reference conclusions, and the proportion of generated conclusions we manually rated as valid and non-redundant. We additionally report the perplexity of re-emitting the premise statements, as we found this to be a common failure mode for both autoregressive language models and sequence-to-sequence models trained without paraphrase data augmentation; in the case of autoregressive models, repetition is a well-known type of degenerative behavior (Holtzman et al., 2020), while for sequence-to-sequence language models such as BART, repeating inputs

Model	Substitution				Contraposition			
	Ref. PPL ↓	**Repeat PPL** ↑	**BLEURT** ↑	**Valid%** ↑	**Ref. PPL** ↓	**Repeat PPL** ↑	**BLEURT** ↑	**Valid%** ↑
			Mean				*Mean*	
Zero-shot GPT2	3.52	2.15	-0.88 ± 0.35	1	6.04	3.37	-0.89 ± 0.31	1
MNLI BART	2.0	1.72	-0.06 ± 0.13	6	4.50	1.32	-0.16 ± 0.04	2
Pattern-only BART	3.55	3.51	0.49 ± 0.01	55	3.16	**4.06**	0.31 ± 0.0	38
PARAPATTERN BART	**1.54**	**3.57**	**0.66 ± 0.05**	**87**	**1.57**	3.77	**0.69 ± 0.07**	**80**
			Control				*Control*	
Zero-shot GPT2	3.28	2.36	-0.93 ± 0.33	3	5.41	3.18	-0.93 ± 0.28	3
MNLI BART	1.79	1.73	0.05 ± 0.15	13	3.81	1.3	-0.25 ± 0.02	1
Pattern-only BART	**1.01**	**7.97**	**0.89 ± 0.0**	**100**	**1.01**	**7.92**	**0.9 ± 0.0**	93
PARAPATTERN BART	1.08	5.01	0.85 ± 0.01	96	1.1	5.15	**0.89 ± 0.02**	**100**
			Link NP mismatch				*Postnominal modifier mismatch*	
Zero-shot GPT2	3.61	2.19	-0.89 ± 0.35	1	6.31	**3.43**	-0.86 ± 0.3	0
MNLI BART	1.91	1.74	-0.04 ± 0.07	0	4.79	1.36	-0.29 ± 0.02	8
Pattern-only BART	1.46	2.38	**0.70 ± 0.0**	53	2.23	2.43	0.0 ± 0.0	0
PARAPATTERN BART	**1.39**	**3.72**	**0.68 ± 0.05**	**87**	**1.39**	2.9	**0.75 ± 0.08**	**87**
			Identity VP mismatch				*Prenominal modifier mismatch*	
Zero-shot GPT2	3.74	2.12	-0.87 ± 0.36	2	6.96	**4.1**	-0.87 ± 0.28	0
MNLI BART	2.17	1.81	-0.07 ± 0.12	3	6.14	1.33	-0.3 ± 0.04	0
Pattern-only BART	4.39	1.65	0.09 ± 0.0	13	7.08	1.74	-0.37 ± 0.0	0
PARAPATTERN BART	**1.59**	**3.04**	**0.52 ± 0.14**	**86**	**1.79**	3.63	**0.48 ± 0.15**	**58**
			NP + VP mismatch				*Premise negation*	
Zero-shot GPT2	4.15	2.04	-0.89 ± 0.35	0	5.5	2.77	-0.87 ± 0.37	3
MNLI BART	2.17	1.70	-0.18 ± 0.17	2	3.24	1.29	0.2 ± 0.07	0
Pattern-only BART	8.37	1.21	0.0 ± 0.0	7	2.32	**4.17**	**0.7 ± 0.0**	60
PARAPATTERN BART	**1.71**	**2.51**	**0.46 ± 0.08**	**75**	**2.01**	3.39	0.64 ± 0.04	**75**
			Number agreement					
Zero-shot GPT2	2.83	2.05	-0.81 ± 0.33	1				
MNLI BART	1.97	1.61	-0.03 ± 0.16	11				
Pattern-only BART	2.53	**4.34**	**0.77 ± 0.0**	**100**				
PARAPATTERN BART	**1.93**	3.58	**0.75 ± 0.02**	93				

Table 1: Results for each contrast set. **Ref. PPL** refers to the perplexity of the reference conclusion under the model distribution. **Repeat PPL** refers to the perplexity of re-emitting the premises. **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 not redundant following manual review. The *Mean* header indicates the section containing aggregate metrics across all test conditions within a contrast set.

is a "default" behavior acquired from their mask-filling pretraining objective.

Results for both contrast sets are presented in Table 1.

Our first question is whether or not we **have a good generative model of natural language deductions.** As the *Mean* section of Table 1 shows, PARAPATTERN BART outperforms both baseline methods by a wide margin in terms of the likelihood it assigns to desired conclusions (**Ref. PPL**), the accuracy of its generated outputs with respect 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). In order to understand the advantage conferred by training on paraphrased inputs, we refer to Table 1's individual test condition sections. Within each contrast set, we can clearly see that training only on template-generated data leads models to significantly underestimate the likelihood of correct conclusions when confronted with perturbed examples that lie off the template data manifold.

Table 1 also confirms our hypothesis that a model trained on RTE data will tend to assign higher likelihood (and thus lower perplexity) to repeating premises than it will to nontrivial conclusions.

Evaluating Model Generations on Contrast Sets Beyond perplexity, we want to see whether decoded samples from the model faithfully model the desired reasoning.

As shown by their mean BLEURT scores, outputs from the PARAPATTERN BART model are the most faithful across the contrast sets as a whole. Note that the model trained without paraphrase data is marginally more confident for control examples, which are a good fit for the original training templates. In more difficult conditions, such as *identity VP mismatch*, *NP + VP mismatch*, and *prenominal modifier mismatch*, the pattern-only model tends to fall back on repeating its inputs; this is reflected in its lower Repeat PPL for these conditions. The relatively high standard deviation of BLEURT scores between samples for both baseline methods reflect their inconsistent generation outputs; the MNLI BART model tends to repeat premises or emit slightly compressed close paraphrases of them while GPT2 produces widely varying hallucinated elaborations of its inputs; as is to be expected from a general language model, these elaborations almost never follow logically from the premises. In contrast, as can be seen in the generation samples in Figure 4, PARAPATTERN outputs are extremely stable and reflect direct logical inferences based on their inputs.

3.4 Results on QASC

Our probe sets are not necessarily "in-domain" for our model, but they still neatly fit the reasoning patterns we are targeting. To test our approach's applicability to data outside its training, we additionally evaluate our substitution model on the fact compositions in the validation split of the QASC dataset (Khot et al., 2020). These fact compositions were annotated by crowd workers as rationales for multiple-choice question answering problems. Workers were prompted with a single background fact from a set of high-quality manually filtered facts from the WorldTree corpus (Jansen et al., 2018) and a corpus provided by the CK-12 Foundation; annotators then selected an additional fact from a retrieved shortlist of filtered web-text sentences and wrote both a question and a combined background fact. Since annotators combined facts with a particular question in mind, there is a certain amount of missing context in many QASC fact combinations. Additionally, annotators were free to combine facts in any way they wanted so long as the result contained some material from each background fact.

Evaluating Model Generations on QASC Figure 5 depicts the curve formed by ranking all PARAPATTERN BART outputs for QASC validation set fact combinations according to their BLEURT scores with respect to the original annotations. Examining the portion of this distribution above 0 BLEURT, we see very close agreement between the content of generated outputs and reference fact combinations, trending towards exact matches above 0.75 BLEURT. On the opposite end of the spectrum, as BLEURT scores become negative, we can see the structure of model outputs diverge from that of the reference fact combinations. However, even as our model's predictions grow further from the annotated reference conclusions, they remain semantically consistent combinations of the premise facts. The prediction for the final example in Figure 5 is still a

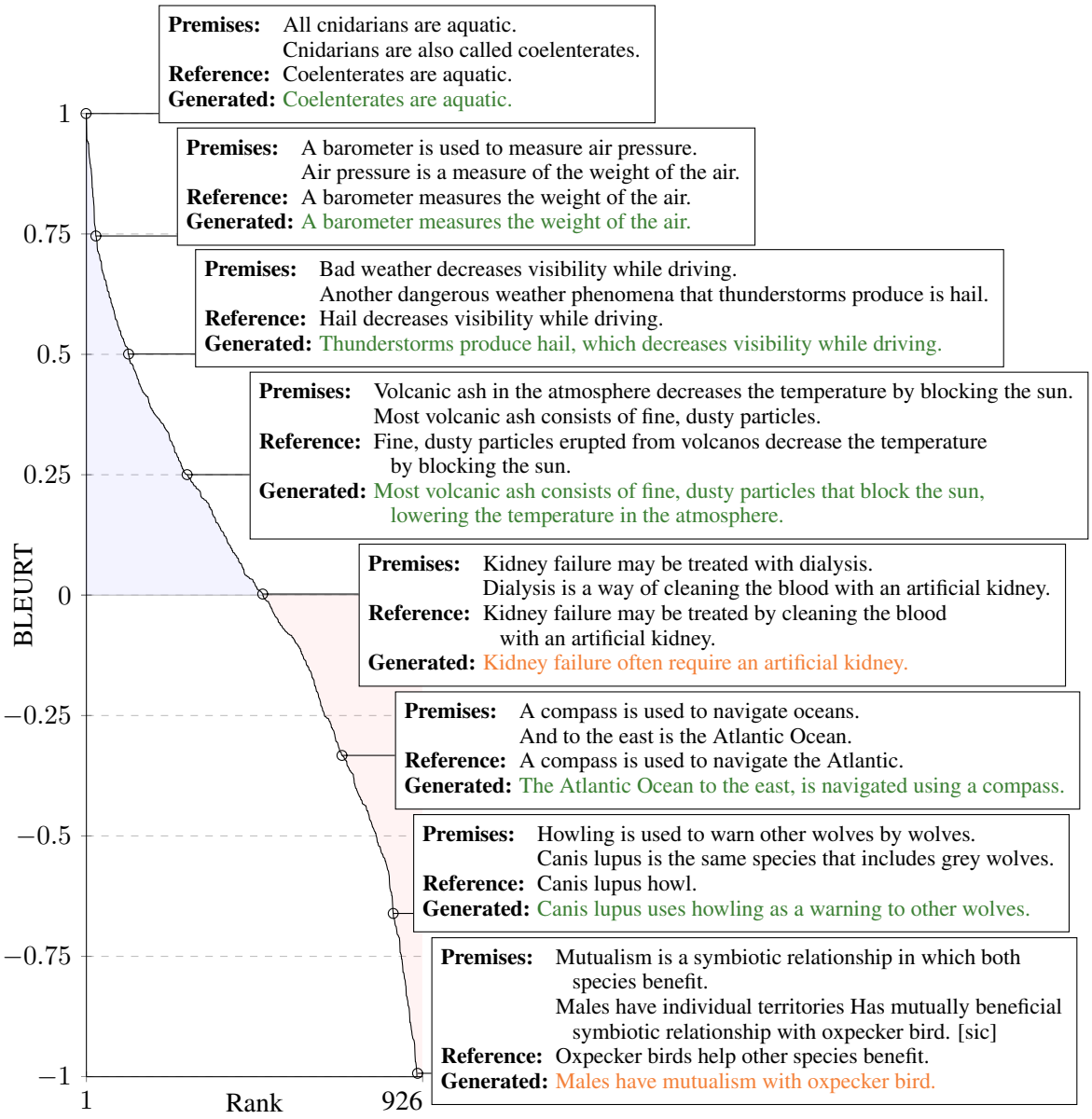

Figure 5: BLEURT score profile of ParaPattern BART substitution outputs for fact combinations from the QASC development set. Sampled substitution model outputs and corresponding QASC annotations for a range of scores are shown to the right. Outputs with minor grammatical errors are indicated in orange. Note that generated conclusions remain semantically coherent despite diverging from annotated references as BLEURT scores decrease.

valid semantic combination of its premises in spite of an ungrammatical input, exemplifying another benefit of training on data augmented with noisily paraphrased inputs.

4 Related Work

Natural Logic (Bernardi, 2002; Zamansky et al., 2006; MacCartney and Manning, 2009; Angeli et al., 2016) is similar 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*, while *there is no cat on the dresser* does not entail *there is no animal on the dresser* because of the negated context. These monotonicity 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). Our work generalizes this reasoning: we do not decompose logical relations between sentences in terms of fine-grained relationships between words, but instead learn relationships between sentences in an end-to-end way.

Multi-hop reasoning Combining multiple facts to form a conclusion overlaps heavily with the idea of multi-hop reasoning, which has been explored in reading comprehension settings (Welbl et al., 2018; Yang et al., 2018). However, methods can achieve high performance on such benchmarks without truly exhibiting multi-hop reasoning (Chen and Durrett, 2019; Min et al., 2019); training end-to-end models on these datasets does not necessarily teach our models the requisite skills. Systems like NLProlog attempt to make this reasoning explicit (Weber et al., 2019); in contrast, by grounding reasoning directly in natural language, a system based on natural deduction operations gains inherent faithful natural language explanations and congruence with 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). Systems like PRover (Saha et al., 2020) and Leap-of-Thought (Talmor et al., 2020) have some broadly similar goals as ours, but only *retrieve* facts and do not generate novel conclusions.

Generative Reasoning The generative capability of our models resembles those used for commonsense inference (Latcinnik and Berant, 2020; Shwartz et al., 2020). However, these models do not strongly constrain the nature of what is generated; we believe our approach is more scalable and can lead to sound inferences over longer reasoning chains in future work. Arabshahi et al. (2021) explored generative reasoning in commonsense scenarios, but the domain of this approach is limited. Khot et al. (2021) use generative models to decompose the decision procedure for a complex question-answering problem into a series of elementary steps that can be delegated to simpler models; this idea parallels our notion of decomposing reasoning into elementary steps to be performed by individual generative operation models.

5 Conclusion

Building systems that use natural language as a medium for reasoning will require operations that can reliably generate logical combinations and transformations of natural language statements. In this work, we present 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 our technique yields operation models capable of generating consistent logical transformations over a diverse range of natural language inputs.

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. *arXiv*, abs/2006.10022.

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 preprint*, 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.

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*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th Annual International Conference on Learning Representations, ICLR 2020*. OpenReview.net.

Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. 2020. A dataset for statutory reasoning in tax law entailment and question answering. *arXiv*, abs/2005.05257.

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. 2020. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. *arXiv*, abs/2010.05953.

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*.

Peter Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton Morrison. 2018. WorldTree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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. *arXiv*, abs/2009.00751.

Veronica Latcinnik and Jonathan Berant. 2020. Explaining question answering models through text generation.

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*.

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.

Cynthia Rudin. 2018. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *arXiv*, abs/1811.10154.

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.

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 J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint*, abs/1912.08777.

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 A

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
```