Critical Thinking for Language Models

Gregor Betz
KIT
Karlsruhe, Germany
gregor.betz@kit.edu

Christian Voigt
KIT
Karlsruhe, Germany
christian.voigt@kit.edu

Kyle Richardson
Allen Institute for AI
Seattle, WA, USA
kyler@allenai.org

Abstract

This paper takes a first step towards a critical thinking curriculum for neural auto-regressive language models. We introduce a synthetic corpus of deductively valid arguments, and generate artificial argumentative texts to train CRiPT: a critical thinking intermediarily pre-trained transformer based on GPT-2. Significant transfer learning effects can be observed: Trained on three simple core schemes, CRiPT accurately completes conclusions of different, and more complex types of arguments, too. CRiPT generalizes the core argument schemes in a correct way. Moreover, we obtain consistent and promising results for NLU benchmarks. In particular, CRiPT's zero-shot accuracy on the GLUE diagnostics exceeds GPT-2's performance by 15 percentage points. The findings suggest that intermediary pre-training on texts that exemplify basic reasoning abilities (such as typically covered in critical thinking textbooks) might help language models to acquire a broad range of reasoning skills. The synthetic argumentative texts presented in this paper are a promising starting point for building such a "critical thinking curriculum for language models."

1 Introduction

Pre-trained autoregressive language models (LM) such as GPT-2 and GPT-3 achieve, remarkably, competitive results in a variety of language modeling benchmarks without task-specific fine-tuning (Radford et al., 2019; Brown et al., 2020). Yet, it is also widely acknowledged that these models struggle with reasoning tasks, such as natural language inference (NLI) or textual entailment (Askell, 2020). Actually, that doesn't come as a surprise, given the tendency of humans to commit errors in reasoning (Kahneman, 2011; Sunstein and Hastie, 2015), their limited critical thinking skills (Paglieri, 2017), and the resulting omnipresence of fallacies and biases in texts and the frequently low argumentative quality of online debates (Hansson, 2004; Guiaşu and Tindale, 2018; Cheng et al., 2017): Neural language models are known to pick up and reproduce *normative* biases (e.g., regarding gender or race) present in the dataset they are trained on (Gilburt and Claydon, 2019; Blodgett et al., 2020; Nadeem et al., 2020), as well as other *annotation artifacts* (Gururangan et al., 2018); no wonder this happens with *argumentative* biases and reasoning flaws, too (Kassner and Schütze, 2020; Talmor et al., 2020). This diagnosis suggests that there is an obvious remedy for LMs' poor reasoning capability: make sure that the training corpus contains a sufficient amount of exemplary episodes of sound reasoning.

In this paper, we take a first step towards the creation of a "critical thinking curriculum" for neural language models. Critical thinking can be loosely defined as "reasonable reflective thinking that is focused on deciding what to believe or do." (Norris and Ennis, 1989) Generally speaking, our study exploits an analogy between teaching critical thinking to students and training language models so as to improve their reasoning skill. More specifically, we build on three key assumptions that are typically made in critical thinking courses and textbooks: First, there exist fundamental reasoning skills that are required for, or highly conducive to, a large variety of more specific and advanced critical thinking skills (e.g., Fisher, 2001, p. 7). Second, drawing deductive inferences is one such basic ability (e.g., Fisher, 2001, pp. 7–8). Third, reasoning skills are not (just) acquired by learning a theory of correct reasoning, but by studying lots of examples and doing "lots of good-quality exercises" (Lau and Chan, 2020), typically moving from simple to more difficult problems (e.g., Bowell and Kemp, 2014).

These insights from teaching critical thinking translate, with respect to our study, as follows (see Fig. 1). First of all, we design and build 'lots of good-quality exercises': a synthetic corpus of de-

Proceedings of the 14th International Conference on Computational Semantics, pages 63–75
June 17–18, 2021.

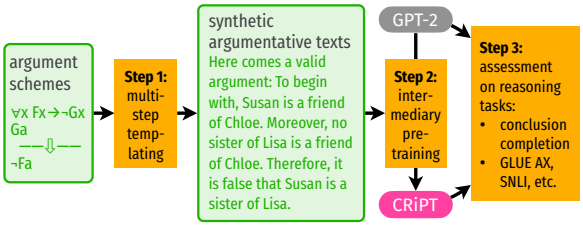

Figure 1: Training and testing of CRiPT language models (critical thinking intermediarily pre-trained transformer) with synthetic argumentative texts.

ductively valid arguments which instantiate a variety of (syllogistic) argument schemes, and which are rendered as text paragraphs (Section 3). Next, we use our synthetic argument text corpus to train and to evaluate GPT-2 (Section 4). The training, which maximizes a causal language modeling objective, can be conceived of as a generic, intermediary pre-training in the spirit of STILTS (Phang et al., 2018) and yields models we term CRiPT (critical thinking intermediarily pre-trained transformer).

Evaluating CRiPT's ability to correctly complete conclusions of arguments, we observe strong transfer learning effects/generalization (Section 5): Just training CRiPT on a few central core schemes (generalized modus ponens, contraposition and chain rule) allows it to accurately complete conclusions of different types of arguments, too (e.g., complex argumentative forms that involve dilemma and de Morgan). The language models appear to connect and generalize the core argument schemes in a correct way. In addition, CRiPT is equally able to apply learned argument patterns beyond the training corpus' domain.

Moreover, we test CRiPT on different reasoning benchmarks. Because we are particularly interested in transfer learning effects, we do so in a zero-shot set-up (i.e., evaluating our argumentation models on entirely unrelated NLU tasks, which follows recent work by Mitra et al. (2019); Shwartz et al. (2020); Ma et al. (2020)). We obtain consistent and promising results for the GLUE diagnostics (Wang et al., 2018) and SNLI (Bowman et al., 2015) benchmarks (Section 5), finding that training on core schemes clearly improves the NLU skills of pre-trained models.

All these transfer learning effects observed strengthen the analogy between teaching critical thinking and training language models: A variety of reasoning skills are improved by generic, intermediary pre-training on high-quality texts that exemplify a basic reasoning skill, namely simple deductive argumentation. Obviously, drawing correct inferences is just one of the elementary skills typically covered in critical thinking courses (Fisher, 2001). Critical thinking involves more than deduction. And it would hence, by analogy, be unreasonable to expect that intermediary pre-training on the synthetic argument corpus suffices to turn language models into accomplished reasoners. However, we have shown that argumentative texts (with valid syllogistic arguments) are certainly a good starting point when building a more comprehensive dataset for initial or intermediary pre-training that might help language models to acquire a broad range of reasoning skills. Or, to put it differently, the synthetic argumentative texts might belong to the core of a "critical thinking curriculum for language models." In the final section, we advance some ideas for complementing the artificial argument corpus so as to further improve the performance of LMs with regard to different reasoning benchmarks.

2 Related Work

To our knowledge, this paper is, together with Gontier et al. (2020), among the first to show that autoregressive language models like GPT-2 can learn to reason by training on a *text corpus* of correct natural language arguments. By contrast, previous work in this field, described below, has typically modeled natural language reasoning problems as classification tasks and trained neural systems to accomplish them. For example, Schick and Schütze (2021); Schick and Schütze (2020) find that a *masked* language model with classification head achieves remarkable NLU performance by pre-structuring the training data. This paper explores the opposite route: We start with highly structured (synthetic) data, render it as unstructured, plain text and train a *uni-directional* language model on the synthetic text corpus.

Over and above the methodological novelty of our approach, we discuss, in the following, related reasoning benchmarks and explain what sets our synthetic argument corpus apart from this work.

Rule reasoning in natural language Various datasets have been developed for (deductive) rule reasoning in natural language. One-step rule application (cf. Weston et al., 2016; Richardson et al., 2020; Tafjord et al., 2019; Lin et al., 2019) closely resembles the conclusion completion task for *gen-*

eralized modus ponens and *generalized modus tollens* schemes described below. However, we go beyond previous work in investigating the ability of LMs to infer conclusions that have a more complex logico-semantic structure (e.g., existential or universal statements). RuleTaker, arguably the most general system for rule reasoning in natural language so far, is a transformer model for multi-hop inference (Clark et al., 2020). PRover (Saha et al., 2020) extends RuleTaker by a component for proof generation and is able to construct valid proofs and outperforms RuleTaker in terms answer accuracy in a zero-shot setting.

Benchmarks for enthymematic reasoning An 'enthymeme' is an argument whose premises are not explicitly stated, e.g.: "Jerry is a mouse. Therefore, Jerry is afraid of cats." The following studies involve such reasoning with implicit assumptions, whereas our synthetic argument corpus doesn't: all premises are transparent and explicitly given. COMET generates and extends commonsense knowledge graphs (Bosselut et al., 2019). Trained on seed data, the model is able to meaningfully relate subject phrases to object phrases (by doing the type of completion tasks we introduce in Section 4). The Argument Reasoning Comprehension (ARC) dataset (Habernal et al., 2018) comprises simple informal arguments. The task consists in identifying which of two alternative statements is the missing premise in the argument (see also Niven and Kao, 2019). CLUTRR is a task generator for relational reasoning on kinship graphs (Sinha et al., 2019). CLUTTR takes a set of (conceptual) rules about family relations as given and constructs set-theoretic possible worlds (represented as graphs) which instantiate these rules. The task consists in inferring the target fact from the base facts alone – the conceptual rules remain implicit. Gontier et al. (2020) show that Transformers do not only learn to draw the correct conclusion (given a CLUTTR task), but also seems to acquire the ability to generate valid proof chains. Finally, training on synthetic knowledge-graph data *from scratch*, Kassner et al. (2020) find that BERT (Devlin et al., 2019) is able to correctly infer novel facts implicit in the training data.

Critical thinking tasks LogiQA (Liu et al., 2020) is a collection of publicly available critical thinking questions, used by the National Civil Servants Examination of China to assess candidates' critical thinking and problem solving skills. Its scope is much broader than our highly specific and carefully designed argument corpus.

3 An Artificial Argument Corpus

This section describes the construction of a synthetic corpus of natural language arguments used for training and evaluating CRiPT.[1]

The corpus is built around eight simple, deductively valid syllogistic argument schemes (top row in Fig. 2). These eight *base schemes* have been chosen because of their logical simplicity as well as their relevance in critical thinking and argument analysis (Feldman, 2014; Bowell and Kemp, 2014; Brun and Betz, 2016). Each of these eight base schemes is manually varied in specific ways to create further deductively correct variants, which are verified for correctness using an off-the-shelf theorem prover.

Negation variants of base schemes are created by substituting a sub-formula with its negation (e.g., $Fx \rightsquigarrow \neg F_1 x$) and/or by applying *duplex negatio affirmat*. *Complex predicates* variants build on base schemes or their respective negation variants and are obtained by substituting atomic predicates with compound disjunctive or conjunctive ones (e.g., $Fx \rightsquigarrow F_1 x \vee F_2 x$). *De Morgan* variants of base schemes are finally derived by applying de Morgan's law to the respective variants created before (a de Morgan variant of modus ponens is, for instance: $\forall x : \neg(Fx \vee Gx) \rightarrow Hx; \neg Fa; \neg Ga \Rightarrow Ha$).

With 2-3 different versions for each of these variations of a base scheme (parameter *n* in Fig. 2), we obtain, in total, 71 distinct handcrafted argument schemes. In view of their simplicity and prominence in natural language argumentation, three of the eight *base schemes* are marked as *core schemes*: generalized modus ponens, generalized contraposition, hypothetical syllogism 1.

Natural language instances of the argument schemes can be created by means of a first-order-logic domain (with names and predicates) and natural language templates for the formal schemes. In order to obtain a large variety of realistic natural language arguments, we have devised (i) a

[1]The corpus as well as the source code used to generate it are available at https://github.com/debatelab/aacorpus. Selected example texts which illustrate, in particular, the multiple domains covered by the corpus are presented in Appendix A.

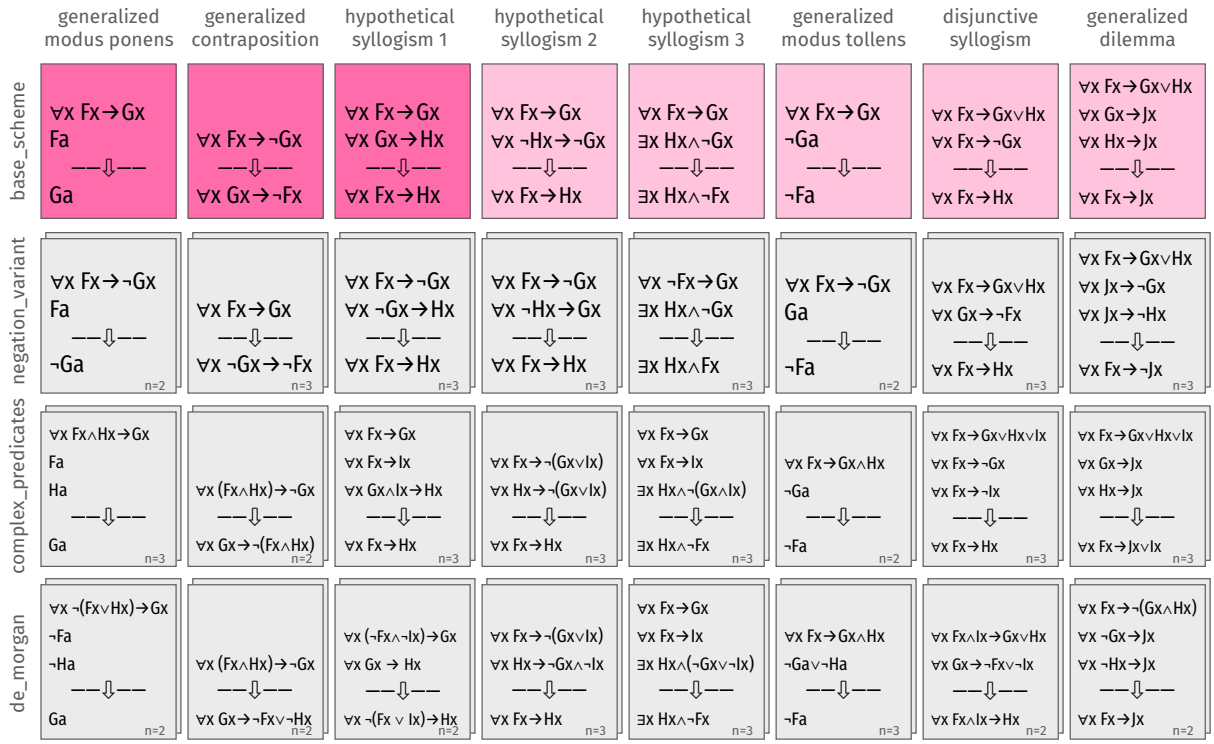

Figure 2: Syllogistic argument schemes used to create an artificial argument corpus with eight base schemes (upper row), three of which are core schemes (left). Parameter n indicates the number of different schemes belonging to one and the same base scheme group (column) and variant (row).

multi-stage templating process with (ii) alternative templates at each stage and (iii) multiple domains.

This process can be split into five consecutive steps.

In *step 1*, the argument scheme, which serves as formal template for the natural language argument, is chosen at random.

In *step 2*, each sentence in the formal scheme (premises and conclusion) is individually replaced by a natural language pattern in accordance with a randomly chosen template. For example, the formula "$\forall x Fx \rightarrow Gx$" might be replaced by any of the following natural language sentence schemes: "Every F is a G", "Whoever is a F is also a G", "Being a G is necessary for being a F", "If someone is a F, then they are a G". Some of these patterns (e.g., the fourth one in the above list) are reserved for generating an out-of-domain test dataset, and are not used for training.

In *step 3*, the entity- and property-placeholders in the resulting argument scheme are replaced argument-wise with names and predicates from a domain. We hence obtain an instance of the formal argument scheme as premise-conclusion list. Each domain provides hundreds of entity-names, which can be paired with different binary predicates to create thousands of different unary predicates. For example, the text in Fig. 1 is obtained by substituting predicates from the domain *female relatives*, which includes predicates like being a "sister of Anna", "granddaughter of Elsa", "cousin of Sarah", ... Once more, some domains are used for testing only, and not for training (see below and Section 4.2).

In *step 4*, the premises of the natural language argument are randomly re-ordered.

In *step 5*, the premise-conclusion list is packed into a text paragraph by adding an argument intro, framing the premises, and adding an inference indicator. Again, multiple templates are available for doing so, which yields a large variety of textual renderings of an argument.

Following this pipeline, we generate natural language instances of each formal argument scheme, thus creating:

1. a training set of argumentative texts, based on the default domains and templates (TRAIN);
2. an evaluation set of argumentative texts, based on the default domains and templates, which are used for development (DEV);
3. a test set of argumentative texts, based on the default domains and templates and used for

final tests (TEST_OUT-OF-SAMPLE);

4. a test set of argumentative texts, based on the domains and templates reserved for testing (TEST_OUT-OF-DOMAIN).

This represents the artificial argument text corpus we use to train and evaluate CRiPT.

4 Experiments with CRiPT

Our basis for training and evaluating CRiPT are three compact versions of GPT-2 with 117M, 345M and 762M parameters, as implemented by Wolf et al. (2019). We note that all of these models fall short of the full-scale model with 1542M parameters.[2]

4.1 Training

From the training items in the Artificial Argument Corpus (TRAIN) we sample three types of differently-sized training sets TRAIN01 $\subset$ TRAIN02 $\subset$ TRAIN03 as follows (see also the color pattern in Fig. 2):

- TRAIN01: all training items which are instances of a *core scheme*, i.e. generalized modus ponens, generalized contraposition, hypothetical syllogism 1 (N=4.5K, 9K, 18K, 36K)
- TRAIN02: all training items which are instances of a *base scheme* (N=4.5K, 9K, 18K, 36K)
- TRAIN03: all training items in the corpus (N=4.5K, 9K, 18K, 36K)

In an attempt to avoid over-fitting, we blend the training arguments with snippets from Reuters news stories (Lewis et al., 2004) and the standardized Project Gutenberg Corpus (Gerlach and Font-Clos, 2018), trying a mixing ratio of 1:1 and thus doubling training size to N=9K, 18K, 36K, 72K.[3] Training the BASE model (pre-trained GPT-2) on TRAIN01–TRAIN03 yields three corresponding CRiPT models (see Appendix B). For purpose of comparison, we have similarly trained three randomly initialized Transformer models (structurally identical with GPT-2) – none of these random models gains any performance through training on our critical thinking corpus.

[2]The fine-tuned models are released through https://huggingface.co/debatelab.

[3]We find that fine-tuning on the accordingly enhanced argument corpus still increases the model's perplexity on the Wiki103 dataset by a factor of 1.5 (see Appendix D), which suggests to mix a higher proportion of common texts into the training data in future work.

4.2 Testing

Conclusion Completion on Artificial Argument Corpus To test whether language models can reason correctly, we assess their ability to accurately complete conclusions of arguments in the artificial argument corpus. Here, we make use of the fact that, by construction, the conclusion of every argument in the corpus ends with a predicate (a property-term such as "sister of Chloe" or "supporter of Tottenham Hotspurs"), which is potentially preceded by a negator. First of all, as shown in Table 1, we test whether the model is able to correctly fill in the final predicate (task *split*). The second, more difficult task consists in completing the final predicate plus, if present, the preceding negator (task *extended*). With a third, adverserial task we check how frequently the model wrongly adjoins the complement of the correct completion of the *extended* task (task *inverted*).

Task	Conclusion with cloze-style prompt	Completion
split	Every *F* is a | *G*	*G*
	Some *F* is not a | *G*	*G*
	a is a *F* or not a | *G*	*G*
extended	Every *F* is | a *G*	a *G*
	Some *F* is | not a *G*	not a *G*
	a is a *F* or | not a *G*	not a *G*
inverted	Every *F* is | a *G*	not a *G*
	Some *F* is | not a *G*	~~not~~ a *G*
	a is a *F* or | not a *G*	~~not~~ a *G*

Table 1: Three conclusion completion tasks

Clearly, the higher the accuracy in the *split* and *extended* tasks, and the lower the accuracy in the *inverted* task, the stronger the model's reasoning performance.

Based on the artificial argument corpus (see Section 3), we generate and distinguish three different test datasets, each of which comprises the three tasks described above, as follows:

- *out of sample (oos)*: contains items from TEST_OUT-OF-SAMPLE, which share domain and natural language templates with the training data;
- *paraphrased (para)*: a sample of 100 items, randomly drawn from TEST_OUT-OF-SAMPLE, which have been manually reformulated so as to alter the premises' grammatical structure

imposed by the natural language templates;

- *out of domain (ood)*: contains items from TEST_OUT-OF-DOMAIN, which belong to different domains and instantiate grammatical patterns other than the training data.

Technically, conclusion completions, in all tasks and tests, are generated by the language model with nucleus sampling and top-p $= 0.9$ (Holtzman et al., 2019).

Classification for NLU Benchmarks To investigate transfer learning effects, we evaluate the trained models on standard NLU benchmarks, such as GLUE AX and SNLI. These benchmark tasks are classification problems. In the following, we describe how we use the generative language models to perform such classification.

Using simple templates, we translate each benchmark entry into alternative prompts (e.g., context and question) and/or alternative completions (e.g., answers). Consider for example a GLUE-style problem given by two sentences "The girl is eating a pizza." and "The girl is eating food" and the question whether one entails, contradicts, or is independent of the other. We can construct three prompts, corresponding to the three possible answers (entail / contradict / independent):

> *Prompt1*: The girl is eating a pizza. Therefore,
> *Prompt2*: The girl is eating a pizza. This rules out that
> *Prompt3*: The girl is eating a pizza. This neither entails nor rules out that
> *Completion*: the girl is eating food.

In this case, the correct match is obviously *Prompt1–Completion*. The ability of a language model to discern that "The girl is eating pizza" entails (and does not contradict) "The girl is eating food" will be reflected in a comparatively low conditional perplexity of *Completion* given *Prompt1* and a correspondingly high conditional perplexity of *Completion* given *Prompt2* or *Prompt3*.

Generally put, we classify a given input X by constructing N alternative prompts $p_1, \ldots p_N$ and a completion c, such that each pair (p_i, c) corresponds to a class $i \in \{1 \ldots N\}$ of the classification problem. The conditional perplexity of the completion c given prompt p_i according to the language model serves as prediction score for our classifier (as for instance in Shwartz et al., 2020).

5 Results

Conclusion Completion on Artificial Argument Corpus Does CRiPT correctly complete conclusions of natural language arguments? Fig. 3 displays the evaluation results in an aggregated way. Each subplot visualizes the accuracy of the models in the three completion tasks for a different test dataset (see Section 4.2).

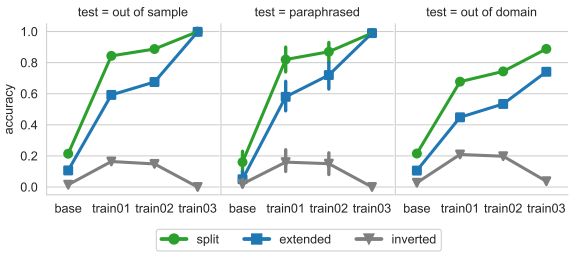

Figure 3: Accuracy of CRiPT in three conclusion completion tasks and on different test datasets (out of sample, paraphrased, out of domain).

We may observe, first of all, that pre-training on the argument corpus effectively improves conclusion-completion-skill. In all three test datasets, the accuracy in the *split* and *extended* tasks increases as models are trained on more and more argument schemes, far exceeding the base model's performance. Once CRiPT has seen all schemes (TRAIN03), accuracy levels reach 100% for in-domain and 70%-90% for out-of-domain tests. However, the TRAIN01 and TRAIN02 models do also generate more incorrect completions than the BASE model (*inverted* task). But the frequency of such incorrect completions increases much less than the frequency of correct ones (the gap between blue and gray curve widens), and it actually falls back to almost zero with the TRAIN03 model. Out-of-domain performance of CRiPT (right-hand plot) is qualitatively similar and only slightly less strong than in-domain performance (left-hand and middle plot). CRiPT models trained on a given domain are able to effectively exercise the acquired skill in other domains, and have hence gained topic-neutral, universal reasoning ability.

The strong performance of TRAIN01 models (Fig. 3) indicates that training on a few argument schemes positively affects performance on other schemes, too. To further investigate transfer learning, Table 2 contrasts (a) CRiPT's accuracy on schemes it has not been trained on – averaged over TRAIN01 and TRAIN02 models – with (b) its accuracy on schemes present in the respective train-

Task	BASE	(A) UNSEEN SCH.			(B) SEEN SCH.		
		oos	*para*	*ood*	*oos*	*para*	*ood*
split	21.4	85.4	82.0	69.4	99.9	99.2	89.0
ext.	10.7	60.3	59.3	45.8	99.9	99.2	76.2
inv.	1.5	16.9	18.0	22.1	0.0	0.0	3.2

Table 2: Accuracy of CRiPT models in three conclusion completion tasks and on different test datasets (out of sample: oos, paraphrased: para, out of domain: ood). Columns report, separately, the performance (A) on schemes the model has not been trained on (TR01–02), and (B) on schemes that are covered by the model's training data (TR01–03). For comparison, column BASE reports the performance of pre-trained GPT-2, averaged over all schemes.

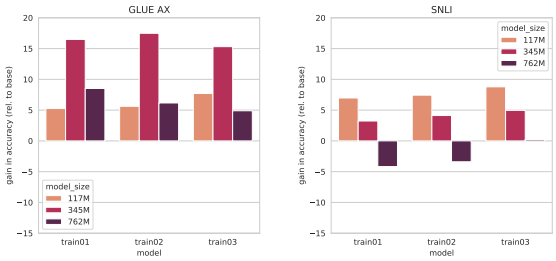

Figure 4: Gains in accuracy due to fine-tuning on the AAC (accuracy TRAIN model – accuracy BASE model) for differently sized models and different NLP benchmark tasks: the GLUE diagnostics data, and the SNLI dataset.

ing corpus – averaged over TRAIN01, TRAIN02, and TRAIN03 models. The upshot is that CRiPT performs much more strongly than the base model not only on argument schemes it has been trained on, but also on those schemes not seen yet. We take this to be a promising result as it strengthens the analogy between teaching critical thinking and training language models: intermediary pre-training on high-quality texts that exemplify a specific, basic reasoning skill – namely, simple deductive argumentation – improves other, more complex reasoning skills.

Moreover, a closer look at the scheme-specific performance suggests important variations in CRiPT's ability to generalize, for it seems to struggle with unseen schemes which involve negations (e.g., CRiPT-TRAIN02 generates more incorrect than correct completions of the *negation_variants* of generalized modus ponens, see Appendix C). This is consistent with the finding that some NLMs seemingly fail to understand simple negation (Kassner and Schütze, 2020; Talmor et al., 2020).

To further understand transfer learning effects, we next examine CRiPT's zero-shot performance in other NLP reasoning tasks (i.e., without task-specific fine-tuning).

GLUE AX The GLUE datasets (Wang et al., 2018) represent standard benchmarks for natural language understanding (NLU). We evaluate our models' NLU skill in terms of accuracy on the curated GLUE diagnostics dataset (Fig. 4).

Training on the artificial argument corpus substantially boosts accuracy on the GLUE diagnostics. Accuracy increases by at least 5 and up to 17 percentage points, depending on model size. Remarkably, training on the core scheme alone suffices to bring about these improvements.

This is a major finding and our clearest evidence so far that critical thinking pre-training involves substantial transfer learning effects.

SNLI Our assessment of CRiPT with respect to SNLI data (Bowman et al., 2015) proceeds in close analogy to the GLUE benchmark. The results (Fig. 4) are consistent with, albeit less definite than our previous findings for the GLUE benchmark: First and foremost, training on all schemes (TRAIN03) improves the performance by up to 8 percentage points. Training on fewer schemes is slightly less effective. However, only small and medium sized CRiPT profit from pre-training on the AAC; while the performance of the 762M model drops. This might be due to a coincidentally strong performance of the corresponding BASE model (see Appendix D), or suggest that large GPT-2 has already learned during general pre-training whatever is of relevance for SNLI in argumentative texts. (Further experiments, preferably involving more model versions, are required to clarify this.)

Besides GLUE AX and SNLI, we have assessed CRiPT on the semantically more demanding Argument Reasoning Comprehension task (Habernal et al., 2018) or the critical thinking assessment compiled in LogiQA (Liu et al., 2020), but found no performance increase compared to the base model.

6 Conclusion

This paper has taken a first step towards the creation of a critical thinking curriculum for neural language models. It presents a corpus of deductively valid, artificial arguments, and uses this artificial argument corpus to train and evaluate CRiPT – a Transformer language model based on GPT-2.

As our main finding, we observe strong transfer learning effects/generalization: Training CRiPT on a few central core schemes allows it to accurately complete conclusions of different types of arguments, too. The language models seem to connect and to generalize the core argument schemes in a correct way. Moreover, CRiPT is equally able to apply learned argument patterns beyond the domain it has been trained on, and there is evidence that generic language modeling skill facilitates the successful generalization of learned argument patterns as randomly initialized models fail to acquire any inference skill by critical thinking pre-training. (Accordingly, we expect our approach to scale to even larger versions of GPT-2.) These findings are consistent with previous work on rule reasoning (Clark et al., 2020). Moreover, CRiPT has been tested on different reasoning benchmarks. We obtain clear and promising results for the GLUE AX and SNLI benchmarks. All this suggests that there exist (learning-wise) fundamental reasoning skills in the sense that generic intermediary pre-training on texts which exemplify these skills leads to spill-over effects and can improve performance on a broad variety of reasoning tasks. The synthetic argumentative texts might be a good starting point for building such a "critical thinking curriculum for language models."

There are different directions for advancing the approach adopted in this paper and further improving the general reasoning skill of neural language models:

- The syllogistic argument text corpus might be complemented with corpora of arguments that instantiate *different kinds of correct schemes*, e.g., propositional inference schemes, modal schemes, argument schemes for practical reasoning, complex argument schemes with intermediary conclusions or assumptions for the sake of the argument, etc. (Technically, we provide the infrastructure for doing so, as all this might be achieved through adjusting the argument corpus configuration file.)
- To succeed in NLI tasks, it doesn't suffice to understand 'what follows.' In addition, a system needs to be able to explicitly discern contradictions and *non sequiturs* (relations of logical independence). This suggests that the artificial argument corpus might be fruitfully supplemented with corpora of correctly identified aporetic clusters (Rescher, 1987) as well as corpora containing correctly diagnosed fallacies.
- In addition, the idea of curriculum learning for ML (Bengio et al., 2009) might be given a try. Accordingly, a critical thinking curriculum with basic exemplars of good reasoning would not only be used to fine-tune a pre-trained model, but would be employed as starting point for training a language model from scratch.

In conclusion, designing a critical thinking curriculum for pre-training neural language models seems to be a promising and worthwhile research program to pursue.

Acknowledgments

An earlier version of this work has been presented at Allen AIs Aristo Group, we profited from critical and constructive feedback.

References

Amanda Askell. 2020. Gpt-3: Towards renaissance models. In *Daily Nous Blog: Philosophers On GPT-3*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA. ACM.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Tracey Bowell and Gary Kemp. 2014. *Critical Thinking: A Concise Guide*, 4th edition edition. Routledge, London.

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Georg Brun and Gregor Betz. 2016. Analysing practical argumentation. In Sven Ove Hansson and Gertrude Hirsch-Hadorn, editors, *The Argumentative Turn in Policy Analysis. Reasoning about Uncertainty*, pages 39–77. Springer, Cham.

J. Cheng, M. Bernstein, C. Danescu-Niculescu-Mizil, and J. Leskovec. 2017. Anyone can become a troll: Causes of trolling behavior in online discussions. *CSCW: Proceedings of the Conference on Computer-Supported Cooperative Work. Conference on Computer-Supported Cooperative Work, 2017*, page 1217–1230.

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.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Richard Feldman. 2014. *Reason and Argument*. Pearson, Harlow.

Alec Fisher. 2001. *Critical Thinking: An Introduction*. Cambridge University Press, Cambridge.

Martin Gerlach and Francesc Font-Clos. 2018. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *CoRR*, abs/1812.08092.

Ben Gilburt and Mark Claydon. 2019. Examining gender bias in OpenAI's GPT-2 language model. *towardsdatascience.com*.

Nicolas Gontier, Koustuv Sinha, Siva Reddy, and Christopher Pal. 2020. Measuring systematic generalization in neural proof generation with transformers.

Radu Cornel Guiaşu and Christopher W Tindale. 2018. Logical fallacies and invasion biology. *Biology & philosophy*, 33(5-6):34.

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.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1930–1940. Association for Computational Linguistics.

Sven Ove Hansson. 2004. Fallacies of risk. *Journal of Risk Research*, 7(3):353–360.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Daniel Kahneman. 2011. *Thinking, fast and slow*, 1st edition. Farrar, Straus and Giroux, New York.

Nora Kassner, Benno Krojer, and Hinrich Schütze. 2020. Are pretrained language models symbolic reasoners over knowledge? In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564, Online. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Joe Lau and Jonathan Chan. 2020. Critical thinking web. *https://philosophy.hku.hk/think*.

D. D. Lewis, Y. Yang, T. Rose, and F. Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. *Proc. MRQA Workshop (EMNLP'19)*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3622–3628. ijcai.org.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2020. Knowledge-driven self-supervision for zero-shot commonsense question answering. *CoRR*, abs/2011.03863.

Arindam Mitra, Pratyay Banerjee, Kuntal Kumar Pal, Swaroop Mishra, and Chitta Baral. 2019. Exploring ways to incorporate additional knowledge to improve natural language commonsense question answering. *CoRR*, abs/1909.08855.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. Stereoset: Measuring stereotypical bias in pretrained language models. *CoRR*, abs/2004.09456.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

SP Norris and RH Ennis. 1989. What is critical thinking. *The practitioner's guide to teaching thinking series: Evaluating critical thinking*, pages 1–26.

Fabio Paglieri. 2017. A plea for ecological argument technologies. *Philosophy & Technology*, 30(2):209–238.

Jason Phang, Thibault Févry, and Samuel R. Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *CoRR*, abs/1811.01088.

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

Nicholas Rescher. 1987. Aporetic method in philosophy. *The Review of metaphysics*, 41(2):283–297.

Kyle Richardson, Hai Hu, Lawrence S. Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8713–8721. AAAI Press.

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 2020, Online, November 16-20, 2020*, pages 122–136. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 255–269. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners.

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 2020, Online, November 16-20, 2020*, pages 4615–4629. Association for Computational Linguistics.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4505–4514. Association for Computational Linguistics.

Cass R Sunstein and Reid Hastie. 2015. *Wiser: getting beyond groupthink to make groups smarter*. Harvard Business Review Press, Boston.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. Quartz: An open-domain dataset of qualitative relationship questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5940–5945. Association for Computational Linguistics.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. olmpics - on what language model pre-training captures. *Trans. Assoc. Comput. Linguistics*, 8:743–758.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

J. Weston, A. Bordes, S. Chopra, and T. Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. *ICLR*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.

A Appendix: Illustrative Examples of Synthetic Argumentative Texts

The following items are drawn from the artificial argument corpus and illustrate the synthetic

texts used to train and test CRiPT – specifically the various *domains* covered in the corpus. Links to the entire dataset and source code for generating synthetic arguments are released at https://github.com/debatelab/aacorpus.

Domain: **female_relatives**. *Base scheme group:* Generalized modus tollens. *Scheme variant:* base scheme. *Text:* It is not always easy to see who is related to whom – and in which ways. The following argument pertains to this question: To start with, Daisy is not a sister of Melissia. Now, being an ancestor of Kerstin is sufficient for being a sister of Melissia. Hence, it is false that Daisy is an ancestor of Kerstin.

Domain: **male_relatives**. *Base scheme group:* Hypothetical Syllogism 1. *Scheme variant:* negation_variant. *Text:* Is Fred a cousin of Robert? Is Joe related to Bob? In large families, it is sometimes difficult to keep track of all one's relatives. The following argument seeks to clarify some such relations: First of all, no schoolmate of Erik is a classmate of Andy. Next, whoever is not a classmate of Andy is a schoolmate of Marvin. We may conclude that every schoolmate of Erik is a schoolmate of Marvin.

Domain: **consumers_personalcare**. *Base scheme group:* Disjunctive Syllogism. *Scheme variant:* negation_variant. *Text:* Consumer research aims at understanding whether users of some products also tend to consume other ones, or not. The following argument seeks to clarify some such relations: Everyone who is an occasional purchaser of Bio Ionic shampoo is a rare consumer of The Body Shop soap, too. Every occasional purchaser of Bio Ionic shampoo is not a rare consumer of The Body Shop soap or a frequent consumer of Shiseido shampoo. It follows that everyone who is an occasional purchaser of Bio Ionic shampoo is a frequent consumer of Shiseido shampoo, too.

Domain: **chemical_ingredients**. *Base scheme group:* Generalized Contraposition. *Scheme variant:* complex_predicates. *Text:* Here comes a perfectly valid argument: No ingredient of Eyeshadow Quad is an ingredient of Midnight Black or an ingredient of Bubble Gum Laquer. We may conclude that no ingredient of Bubble Gum Laquer and no ingredient of Midnight Black is an ingredient of Eyeshadow Quad.

Domain: **football_fans**. *Base scheme group:* Generalized Dilemma. *Scheme variant:* base scheme. *Text:* Is Fred a fan of Liverpool? Are supporters of Real Madrid devotees of PSG? In European football, it is sometimes difficult to keep track of the mutual admiration and dislike. The following argument seeks to clarify some such relations: Every friend of FC Olexandriya is either a backer of The New Saints FC or an ex-fan of Olympique Lyonnais, or both. Everyone who is an ex-fan of Olympique Lyonnais is a devotee of RC Celta de Vigo, too. Everyone who is a backer of The New Saints FC is a devotee of RC Celta de Vigo, too. In consequence, being a devotee of RC Celta de Vigo is necessary for being a friend of FC Olexandriya.

Domain: **dinos**. *Base scheme group:* Modus barbara. *Scheme variant:* base scheme. *Text:* Consider the following argument: If someone is a predator of Iguanodon, then they are a prey of Stegosaurus. Parasaurolophus is a predator of Iguanodon. Thus, Parasaurolophus is a prey of Stegosaurus.

Domain: **philosophers**. *Base scheme group:* Hypothetical Syllogism 3 *Scheme variant:* negation_variant *Text:* Here comes a perfectly valid argument: If someone is not a teacher of Diodorus of Adramyttium, then they are a teacher of Dexippus. Moreover, someone is a student of Alexicrates and not a teacher of Dexippus. Thus, someone is a student of Alexicrates and a teacher of Diodorus of Adramyttium.

B Appendix: Training Parameters

We train differently sized versions of GPT-2 with causal language modeling objective (using default training scripts by Wolf et al. (2019)) on each of the 12 enhanced, differently sized training sets. This gives us 36 fine-tuned CRiPT models plus the three BASE models to evaluate. Unless explicitly stated otherwise, the main article reports results of the 762M parameter model trained on 72K items. We train the models on 8 GPUs for 2 epochs with batch size = 2, learning rate = 5×10^{-5}, gradient accumulation steps = 2, and default parameters of the HuggingFace implementation otherwise (Wolf et al., 2019).

C Appendix: Performance Metrics on Different Argument Schemes

Fig. 5 displays CRiPT's accuracy on conclusion completion tasks on specific argument schemes. Its subplots are arranged in a grid that mirrors the organisation of argument schemes as presented in the main article. Each subplot visualizes the abil-

ity of CRiPT to correctly complete arguments of the corresponding scheme (given the out-of-sample test dataset). Reported accuracy values that fall within gray background areas are attained by models which have seen the corresponding scheme during training. Vice versa, thick lines on white background visualize model performance on unknown schemes. Fig. 5 reveals, first of all, that even the BASE models (only pre-training, no fine-tuning) display a significant ability to correctly complete conclusions of some kinds of arguments. For example, GPT-2-762M achieves 50% accuracy (*split* task) in completing contrapositions, 30% accuracy in completing generalized modus ponens, and still 20% accuracy in completing disjunctive syllogism and dilemma arguments. These findings further corroborate the hypothesis that NLMs learn (basic) linguistic and reasoning skills "on the fly" by training on a large generic corpus (Radford et al., 2019).

D Appendix: Performance Metrics for Differently Sized Training Sets

Fig. 6 displays accuracy values on conclusion completion tasks for models trained on differently sized datasets.

Fig. 7 reports perplexity and NLU accuracy metrics for models trained on differently sized datasets.

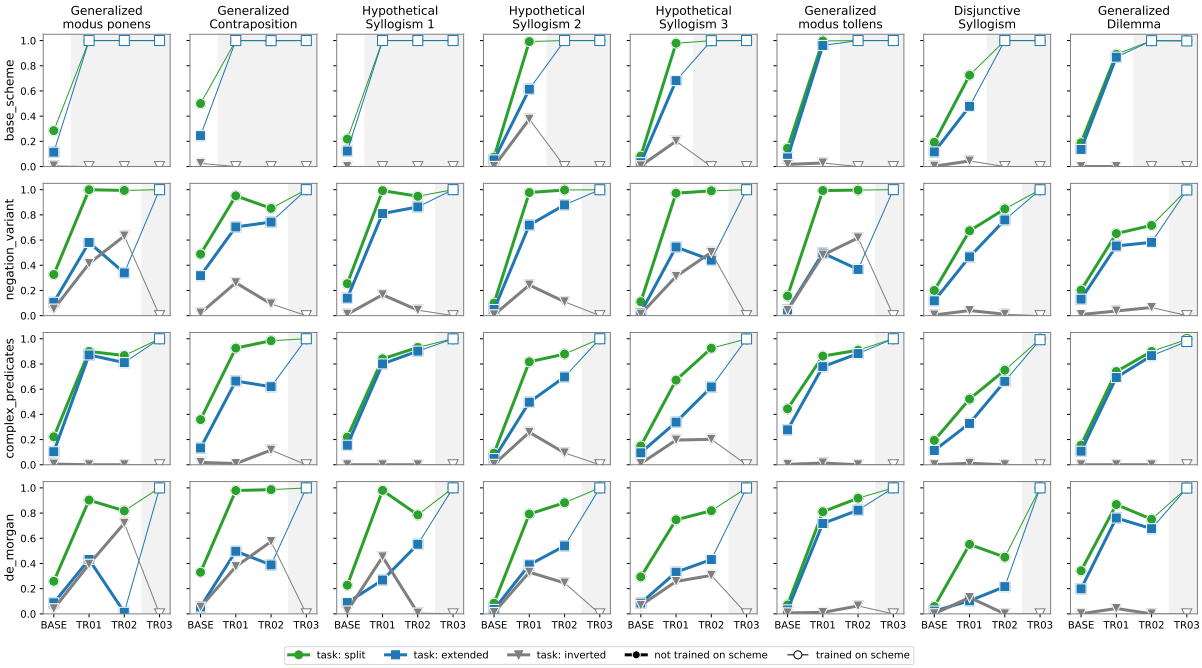

Figure 5: Accuracy of CRiPT in three conclusion completion tasks and on different test datasets (out of sample, paraphrased, out of domain) by argument scheme.

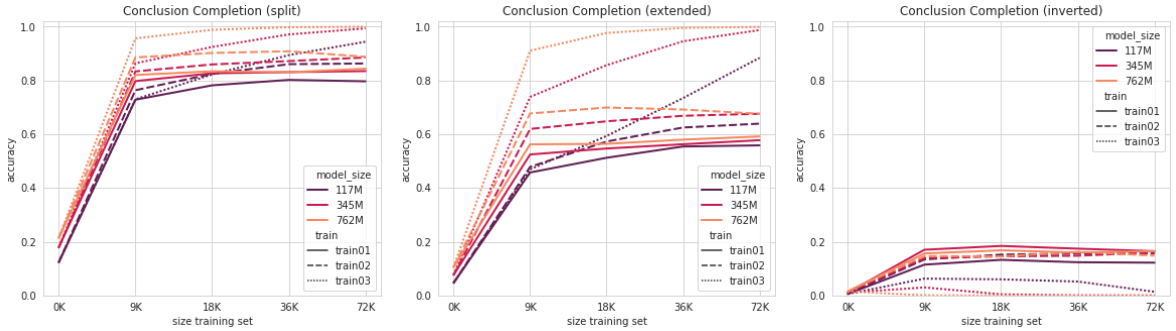

Figure 6: Accuracy on three conclusion completion tasks as a function of training corpus size.

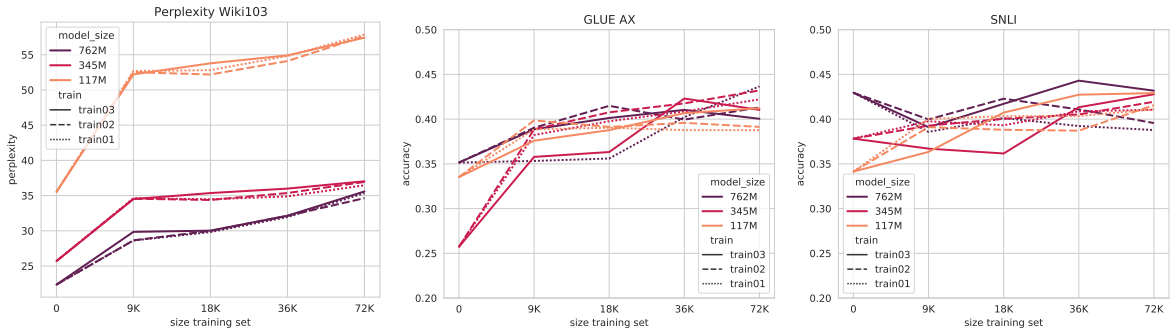

Figure 7: Perplexity and NLI metrics as a function of training corpus size.