APOLLO: A Simple Approach for Adaptive Pretraining of Language Models for Logical Reasoning

Soumya Sanyal[1*] **Yichong Xu**[2] **Shuohang Wang**[2] **Ziyi Yang**[2]
Reid Pryzant[2] **Wenhao Yu**[3*] **Chenguang Zhu**[2] **Xiang Ren**[1]
[1]University of Southern California [2]Microsoft Cognitive Service Research
[3]University of Notre Dame
soumyasa@usc.edu

Abstract

Logical reasoning over text is an important ability that requires understanding the semantics of the text and reasoning through them to arrive at correct inferences. Prior works on pretraining language models to improve the logical reasoning ability require complex processing of training data (e.g., aligning symbolic knowledge to text), yielding task-specific solutions that are not easy to adapt to any general text corpus. In this work, we propose APOLLO, a simple adaptive pretraining approach to improve the logical reasoning skills of language models. We select a subset of Wikipedia for adaptive pretraining using a set of logical inference keywords as filter words. Further, we propose two self-supervised loss functions for training. First, we modify the masked language modeling loss to mask specific parts-of-speech words that likely require higher-order reasoning to predict them. Second, we propose a sentence-level classification loss that teaches the model to distinguish between entailment and contradiction types of sentences. The proposed pretraining paradigm is both simple and independent of task formats. We demonstrate the effectiveness of APOLLO by comparing it with prior baselines on two logical reasoning datasets. APOLLO performs comparably on ReClor and outperforms baselines on LogiQA. The code base has been made publicly available.[1]

1 Introduction

Logical reasoning is an important ability of humans that helps us in making rational decisions based on known information. It is an important ability for text understanding across various downstream tasks, e.g., in open-domain question answering (Yang et al., 2018; Zhu et al., 2021), machine reading comprehension (MRC) (Baradaran et al., 2022), etc. Recently, there has been an increasing focus on evaluating the logical reasoning abilities of language models by using MRC tasks that specifically require a significant amount of logical reasoning to obtain the correct answer (Yu et al., 2020; Liu et al., 2021). In these datasets, the model needs to understand a given context, reason logically about a question to infer new conclusions, and then select the correct answer from a set of options. With the advent of large pre-trained language models (PLMs) in NLP (Devlin et al., 2019; Radford et al., 2019; Raffel et al., 2020), understanding and improving the logical reasoning abilities of these models has become even more important as these are increasingly being used across a wide variety of real-world tasks.

Input Sentence:
Earth's temperature would drop down if it were frozen entirely, because it would be more reflective.

Random Masking:
[MASK] temperature [MASK] drop [MASK] if it were frozen entirely, because it would be [MASK] reflective.

Selective Masking:
Earth's temperature would [MASK] down if it were [MASK] entirely, [MASK] it would be [MASK] reflective.

Figure 1: **Motivation of Selective Masking.** In random masking (Devlin et al., 2019), a word is masked at random. Predicting these words often require more of language understanding than higher-order reasoning (e.g., predicting "would" at the 2^{nd} [MASK] place). In selective masking, a word is masked if its POS tag is from a specific set. These candidate words are marked in the blue box in the input sentence. Filling these words requires more reasoning (e.g., to predict "more" at the 2^{nd} [MASK] place instead of "less", which is also grammatically valid, the model needs a better understanding of the semantics of the sentence).

There have been some recent works on improving the logical reasoning abilities of PLMs (Wang et al., 2022; Ouyang et al., 2022; Jiao et al., 2022). These works typically generate a dataset containing symbolic structures such as logical graphs from

[*] Work done during internship at Microsoft Cognitive Service Research group

[1] https://github.com/INK-USC/APOLLO

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6308–6321
July 9-14, 2023 ©2023 Association for Computational Linguistics

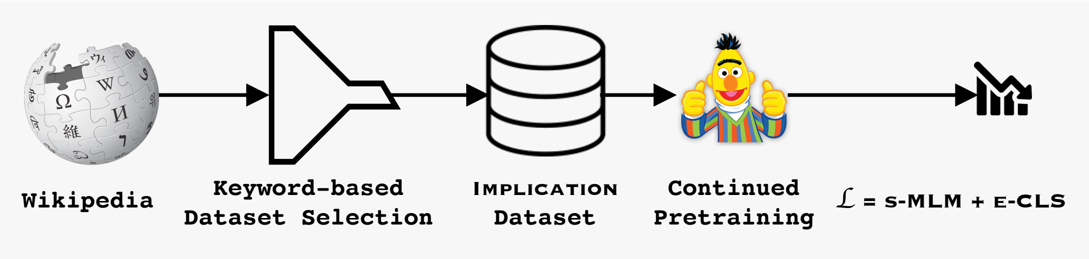
Wikipedia
Keyword-based Dataset Selection
IMPLICATION Dataset
Continued Pretraining
$\mathcal{L}$ = S-MLM + E-CLS

Figure 2: **Overview of APOLLO**. We filter Wikipedia using specific logical keywords to create the IMPLICATION dataset. This is then used for continued pretraining of a model using two loss objectives: selective masked language modeling (S-MLM) loss and entailment classification (E-CLS) loss. Please refer to Section 2 and Figure 3 for more details on the data selection process and loss function designs.

text, logical contrast sets, etc., and then train the LM using custom loss objectives to learn logical reasoning abilities. While the performance improvements achieved by these methods are encouraging, the proposed solutions generally require complex data processing to generate the additional structural information (graphs, contrast data, etc.) required for training the model. For example, Jiao et al. (2022) constructs synthetic context-answer pairs using the entity-level graph from Wikipedia for training the model. Further, the loss functions proposed in these works are very specifically designed in accordance with their respective data augmentation technique and widely differs from the typical masked language modeling loss used for LM pretraining (Devlin et al., 2019). Additionally, some of these works usually require task-specific design choices, which are not necessarily learning generalizable logical reasoning ability that is reusable across different task formats. For example, Wang et al. (2022) parses symbolic logical structures from the training data of a specific dataset, which might not generalize to a new dataset or task. Overall, it is unclear if these highly specific inductive biases are indeed essential for improving the logical reasoning abilities in language models, or if a simpler approach is possible.

On the other hand, prior works (Gururangan et al., 2020) have shown that continual domain-adaptive pretraining of PLMs leads to performance gains on downstream tasks. Inspired by this, we propose APOLLO, a continual pretraining-based approach to inject logical reasoning abilities in language models that requires minimal data processing and loss function modifications.

Firstly, we present a simple way of selecting sentences for training a model that is more likely to involve logical implications. We achieve this by defining a set of logical inference keywords and selecting a subset of sentences from a large text corpus, each containing at least one of these keywords. We hypothesize that PLMs can learn logical reasoning capabilities more easily using such sentences since the premise/conclusions are explicitly stated. We note that in contrast to previous works (Gururangan et al., 2020), our method can select sentences from any general text corpus, eliminating the need for any domain-specific corpus.

Secondly, we modify the masked language modeling (MLM) loss (Devlin et al., 2019) to selectively mask specific words in the sentence, based on their parts-of-speech tags. Prior works (Lad et al., 2022) have shown the benefit of selective masking of words on task-guided fine-tuning. We hypothesize that masking words with parts-of-speech (POS) tags that are related to higher-order reasoning (such as adverbs, conjunctions, etc.) present more challenging masked positions for the PLM to predict. For instance, in Figure 1, we observe that the words marked in blue boxes are more related to reasoning compared to the non-highlighted words that mainly involve knowledge about specific nouns or English grammar.

Lastly, we design a sentence-level classification loss to predict if the reasoning in the sentence describes an entailment in the reasoning process or a contradiction. This enables the model to better understand the differences between positive and negative implications in a sentence, thus improving logical reasoning.

To test APOLLO, we evaluate it on two downstream logical reasoning tasks: ReClor (Yu et al., 2020) and LogiQA (Liu et al., 2021), and compare it with other baselines. We achieve state-of-the-art performance on LogiQA and comparable performance on ReClor. We demonstrate that our method generalizes across different model types. Further, we show that using our proposed loss functions does not induce any catastrophic forgetting (Kirkpatrick et al., 2017) of the original language modeling skills. This demonstrates that our simple, continual pretraining approach is generalizable to different datasets and enables the PLM to acquire strong logical reasoning abilities.

Overall, compared to prior works, our proposed pretraining paradigm for APOLLO 1) Uses sentences from text corpus for training instead of complex data structures such as entity graphs, etc. 2) Uses simple learning objectives that are closer to language modeling compared to the contrastive loss. 3) Is agnostic to both task format and downstream datasets. 4) Achieves state-of-the-art performance on LogiQA.

2 Method

In this section, we describe the details of our proposed approach. In APOLLO, we use a keyword-based selection strategy to collect a dataset of reasoning-related sentences called IMPLICATION (§2.1) and then continue training a pretrained model checkpoint jointly using two loss functions (§2.2). This model is then fine-tuned on the training dataset of each task separately for evaluation. A detailed overview of the pipeline is shown in Figure 2.

2.1 Dataset Selection

PLMs are typically trained on web data which helps them to learn general language modeling capability. Then, PLMs are finetuned on downstream datasets to specialize on target tasks (Devlin et al., 2019; Radford et al., 2018; Raffel et al., 2020). Here, instead of focusing on a specific task, we want to teach the PLM generalizable logical reasoning abilities. We hypothesize that using training data that contains more logical sentences, rather than generic internet data, should help in improving the reasoning ability of the PLM.

Although creating such a dataset automatically is a challenging task by itself, in APOLLO, we explore a simple and intuitive way to create such a dataset. First, we select specific keywords that are typically encountered in sentences with logical implications. Broadly, we categorize these keywords into two types[2]:

- **Positive implication (Entailment)**: These keywords are present in sentences where the reason generally entails the inference. Examples of such keywords would be "therefore", "accordingly", etc.
- **Negative implication (Contradiction)**: The keywords in this category are usually present in sentences where the reason contradicts the inference. For example, keywords such as "but", "although", etc., come under this category.

Next, we select sentences from Wikipedia such that they contain at least one of the keywords. We name this filtered version of Wikipedia as the IMPLICATION dataset. While this keyword-based filtering does not necessarily ensure that the sentence has a logical implication, the retained data contains a higher portion of logically rich sentences than the general data. We argue that pretraining on this data helps the PLM to improve logical reasoning skills. Please refer to Appendix A for more details on the list of keywords used to build the IMPLICATION dataset.

2.2 Learning objectives

Selective masked language modeling (S-MLM) is a modified version of the masked language modeling (MLM) loss used in BERT (Devlin et al., 2019). In the MLM loss, tokens in a sentence are masked at random and the model learns to predict the masked tokens. While this helps in learning a good language model, not all masked tokens require a similar degree of reasoning to predict them. In the example shown in Figure 3, words such as "were", "the", etc. are decided more by the structure of the English language than any form of reasoning. In contrast, predicting logical words such as "more", "and" and "hence" would require more logical reasoning. Thus, we hypothesize that masking these logical words would likely teach the model to perform reasoning more effectively than masking a word at random.

While finding these exact logical words for a given sentence is a hard problem, in APOLLO we simplify this by using a heuristic approach to consider words that belong to a specific set of parts-of-speech (POS) tags. More concretely, in S-MLM loss, we only randomly mask words with these 7 SpaCy POS tags (Honnibal and Montani, 2017): ADJ, ADV, CONJ, CCONJ, PART, SCONJ, and VERB. Please refer to Section 4.4 for more empirical results that further justify this choice.

[2] Appendix A presents the comprehensive list of keywords used to build the IMPLICATION dataset.

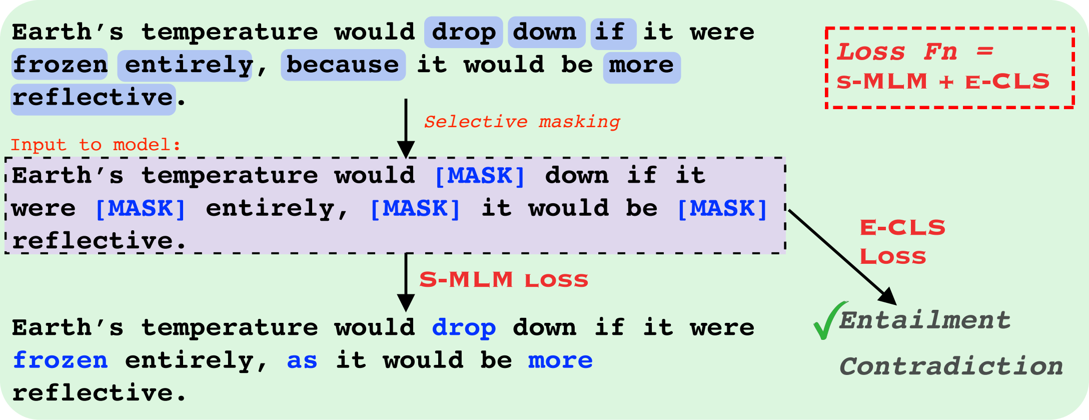
Earth's temperature would drop down if it were frozen entirely, because it would be more reflective.
Selective masking
Input to model:
Earth's temperature would [MASK] down if it were [MASK] entirely, [MASK] it would be [MASK] reflective.
S-MLM Loss
Earth's temperature would drop down if it were frozen entirely, as it would be more reflective.
Loss Fn = S-MLM + E-CLS
E-CLS Loss
✓Entailment
Contradiction

Figure 3: **Learning objectives in APOLLO.** The selective masking step masks out words from a specific set of POS tags (the candidate words are shown in blue boxes). The S-MLM loss then predicts these masked words. The E-CLS loss classifies the masked sentence into one of two categories: entailment or contradiction. The overall loss function used in APOLLO is the sum of both these objectives.

Entailment classification (E-CLS) Prior works have shown that semantic-aware sentence-level classification loss can be useful to learn the semantic information (Sun et al., 2020). Inspired by this, in addition to S-MLM, we use another auxiliary loss function that predicts whether a masked sentence contains some reasoning aspects that portray a sense of entailment or contradiction within the sentence. For example, in Figure 3, the sentence is classified as "Entailment", because the phrase "more reflective" is entailed by the phrase "frozen entirely". We note that the input to the model is the same sentence with masked words that is used for S-MLM loss. A model would ideally require strong logical reasoning abilities to understand the sentence and then predict if it refers to an entailment or contradiction. The labels for this loss are bootstrapped using the heuristic of checking the type of implication keyword present in the sentence (refer to Section 2.1 for details). We note that although the keyword is a correlated feature that can be used to predict the label, on average the keyword would be masked out due to our selective masking policy, forcing the model to learn some logical semantics to minimize the loss. Additionally, even if the model predicts a wrong keyword in the sentence, it may still get the relationship between the sentences correctly. Therefore, the classification loss adds a stronger inductive bias specifically about the reasoning semantics in the sentence than the S-MLM loss.

2.3 Continual Pretraining

In APOLLO, we combine both S-MLM and E-CLS objectives as a joint loss function to continually train a pretrained model checkpoint (Figure 2). Unlike prior works (Jiao et al., 2022), we don't need to add MLM loss to avoid catastrophic forgetting, as S-MLM is quite close to the standard MLM objective in format.

2.4 Finetuning

As our loss functions are task-format agnostic, we follow Devlin et al. (2019) and add a randomly initialized MLP layer on top of the continually pretrained model. Then, we finetune the combined model on downstream datasets.

3 Experimental Setup

In this section, we describe the details of the datasets on which we evaluate APOLLO, the baselines we compare it with, and some implementation details of our training procedure.

3.1 Datasets

Following prior works (Jiao et al., 2022), we evaluate APOLLO on two logical reasoning datasets:

ReClor (Yu et al., 2020) is a reading comprehension dataset created from the logical reasoning questions from standardized graduate admission examinations. The test set is divided into two subsets: EASY (test-E) and HARD (test-H), where the EASY set contains instances whose options can be selected correctly without knowing the context and question. The train/dev/test split consists of 4,638/500/1,000 instances, respectively.

LogiQA (Liu et al., 2021) is developed using publicly available logical examination papers for reading comprehension. The train/dev/test split consists of 7,376/651/651 instances, respectively.

3.2 Baselines

We compare the accuracy of APOLLO with the following baselines: LRReasoner (Wang et al., 2022), DAGN (Huang et al., 2021), FOCAL REASONER (Ouyang et al., 2022), and MERIt (Jiao et al., 2022).

3.3 Implementation Details

For creating the IMPLICATION dataset, we use the Wikipedia version provided under HuggingFace Datasets (Wolf et al., 2020) as the main corpus.[3] The list of keywords we use for filtering sentences from Wikipedia are listed in Appendix A. We experiment with RoBERTa-Large (Liu et al., 2019a), DeBERTa-v3 (He et al., 2021), and DeBERTa-v2-xxlarge (He et al., 2020) as the base models for APOLLO. We pretrain the last two layers of the Transformer (Vaswani et al., 2017) layer for 3 epochs, using a batch size of 4096. Please refer to Appendix B for more details on training and finetuning hyperparameters.

4 Results

4.1 Overall Results

In this section, we compare the performance of APOLLO with prior baselines on the two logical reasoning datasets for different base architectures. The results of using pretrained Roberta-Large as the starting checkpoint for our method are shown in Table 1. We observe that APOLLO outperforms all baselines on LogiQA and performs lower on ReClor than three baselines, although consistently outperforming the RoBERTa baseline. Overall, this demonstrates that our simple continual pretraining approach is indeed strong enough to perform well on logical reasoning tasks as compared to the prior models that depend on much more complex training data and loss function designs.

To test the generality of our approach across different architectures, we use pretrained DeBERTa-v3 and DeBERTa-v2-xxlarge as the base models for continued training. The results of using these models are shown in Table 2. We find that APOLLO outperforms both the baselines on both datasets. Further, we observe that APOLLO performs 1.5% worse compared to MERIt on ReClor test set. This shows that our continual pretraining process can improve performance across different LM architectures.

4.2 Performance on GLUE Benchmark

While improving the logical reasoning abilities of a PLM is important, it is equally important to retain the natural language understanding skills learned during pretraining. To demonstrate that our proposed approach does not lead to catastrophic forgetting, we finetune APOLLO on each dataset of the GLUE benchmark (Wang et al., 2019) and evaluate the finetuned checkpoint on the Dev set. The results are compared with the Dev set results for the RoBERTa model (Liu et al., 2019b) in Table 3. Following Devlin et al. (2019), we omit the evaluation on the problematic WNLI set. Overall, we observe that APOLLO can slightly improve the overall performance on the GLUE benchmark. This demonstrates that our proposed continued pretraining strategy is able to learn better logical reasoning abilities without any catastrophic forgetting of general-purpose language modeling skills, and these logical reasoning capabilities are also beneficial for general natural language understanding.

4.3 Qualitative Analysis

In this section, we analyze the effect of continued pretraining on the model's overall faithfulness. Post-hoc interpretability methods such as Integrated Gradients (Sundararajan et al., 2017), are algorithms to determine the importance of words in the input towards predicting a particular class. These importance scores are also referred to as *attribution scores*. To approximate the impact of continued pretraining, we compute the overall change in attribution scores for the implication keywords, before and after pretraining the model using our proposed datasets and loss functions. Specifically, we compute the sum of the attribution scores for the keywords present in each instance of the validation set. The results are shown in Figure 4. We observe that our proposed pretraining increases the overall attribution score by a significant margin, indicating that the model intrinsically learns these important logical keywords, which is desirable.

4.4 Ablation Studies

In this section, we ablate various design choices in constructing the IMPLICATION dataset, and our proposed method. For the ablations involving APOLLO, we use RoBERTa-Large as the base

[3] https://huggingface.co/datasets/wikipedia

Model	ReClor				LogiQA	
	Dev	Test	Test-E	Test-H	Dev	Test
RoBERTa	62.6	55.6	75.5	40.0	35	35.3
DAGN	65.2	58.2	76.1	44.1	35.5	38.7
LRReasoner	66.2	**62.4**	**81.4**	**47.5**	38.1	40.6
Focal Reasoner	66.8	58.9	77.1	44.6	41.0	40.3
MERIt	**67.8**	60.7	79.6	45.9	**42.4**	41.5
Apollo	67.2	58.2	76.8	43.6	41.6	**42.1**

Table 1: Comparison of Apollo with other baselines on ReClor and LogiQA. All the models are based on the RoBERTa-large model. The results for all the baselines are reported from Jiao et al. (2022). Please refer to Section 4.1 for more details.

Model	ReClor				LogiQA	
	Dev	Test	Test-E	Test-H	Dev	Test
DeBERTa-v3	75.4	71.0	80.2	64.0	45.2	40.1
Apollo (DeBERTa-v3)	**76.8**	**72.8**	**81.8**	**65.7**	**48.4**	**44.4**
DeBERTa-v2-xxlarge	78.3	75.3	84.0	68.4	45.9	49.8
MERIt (DeBERTa-v2-xxlarge)	80.6	**78.1**	84.6	**72.9**	-	-
Apollo (DeBERTa-v2-xxlarge)	**81.8**	76.5	**85.2**	69.6	**49.6**	**51.0**

Table 2: Comparison of Apollo with other baselines on ReClor and LogiQA with DeBERTa as the base architecture. Results for MERIt are reported from Jiao et al. (2022), which is missing results on LogiQA. Other baselines are reproduced by ourselves. The base models are shown in brackets. Please refer to Section 4.1 for more details.

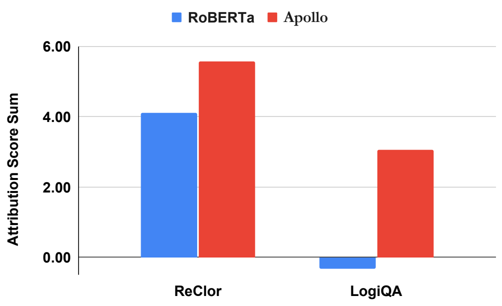

Figure 4: Comparison plot of the keyword attribution scores between RoBERTa-large and Apollo. Please refer to 4.3 for more details.

model and the Implication dataset, if not mentioned separately. All the reported numbers are on the validation set of the downstream task, since we used these ablation studies in our model's design choices.

Effect of datasets and loss functions To study the effect of using Implication for continued pretraining along with the proposed loss functions, we first create Random, a random subset of Wikipedia of similar size as that of Implication, and also consider using the standard masked language modeling (MLM) loss (Devlin et al., 2019), where any token can be masked at random. The results of the ablation are shown in Table 4. We observe that using the Implication dataset leads to consistent improvements on both datasets when compared to the Random dataset. Additionally, we find that both the s-MLM and e-CLS loss lead to improvements over MLM loss. Thus, this empirically justifies our choice of the dataset and loss functions proposed here.

Effect of keyword category In this ablation, we study the effect of the keyword categories that we use for filtering Wikipedia. For this, we create two different pretraining datasets Implication-Positive and Implication-Negative using the positive and negative implication keywords, respectively (refer to Section 2.1). The total number of sentences in these datasets is 7.5M and 11.3M, respectively. Our complete dataset Implication thus has a total of 18.3M sentences. The results of the ablation are shown in Table 5, under the section "Keyword Category". We observe that Implication-Positive, although smaller in size, leads to better performance on both downstream tasks, com-

Model	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	**Avg**
RoBERTa-Large	90.2	94.7	92.2	86.6	96.4	90.9	68.0	92.4	88.9
APOLLO	90.3	94.9	92.1	88.1	96.2	92.2	68.6	91.9	**89.3**

Table 3: Performance on the dev set of GLUE benchmark. Following Devlin et al. (2019), we do not report performance on the WNLI dataset. Please refer to Section 4.2 for further details.

Model (Dataset, Loss functions)	**ReClor**	**LogiQA**
RoBERTa (RANDOM, MLM)	60.2	35.0
RoBERTa (RANDOM, S-MLM)	**63.8**	**36.4**
RoBERTa (IMPLICATION, MLM)	64.8	36.6
RoBERTa (IMPLICATION, S-MLM)	65.4	41.5
RoBERTa (IMPLICATION, S-MLM + E-CLS)	**67.2**	**41.6**

Table 4: Effect of IMPLICATION dataset and the loss functions on the dev set performance of ReClor and LogiQA.

	ReClor	**LogiQA**
Keyword Category		
IMPLICATION-Positive	65.0	38.6
IMPLICATION-Negative	64.6	37.6
IMPLICATION	**65.4**	**41.5**
POS Category		
Base	**65.4**	**41.5**
Base + Nouns	64.0	39.0
Base + Nouns + Random	64.8	36.6

Table 5: Ablation of design choices involved in keyword-based dataset selection and S-MLM loss function implementation. We report the performance on the dev set of each dataset. Please refer to Section 4.4 for more details.

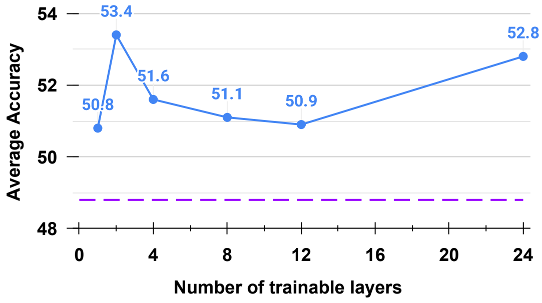

Figure 5: Average performance on the dev set of ReClor and LogiQA with increasing number of trainable layers of APOLLO. The pink dashed line shows the average performance of RoBERTa-Large when all layers are finetuned. Please refer to Section 4.4 for more details.

pared to IMPLICATION-Negative. One reason for this is that the sentences with positive keywords are more likely related to reasoning than the negative counterparts because the negative keywords are used in many diverse scenarios in the English language. For example, the word "*still*" can be used in a non-logical manner such as "*I am still waiting for the call*". Overall, we observe that the combined IMPLICATION dataset leads to the best performance, demonstrating that both the positive and negative implication keywords are essential to improve logical reasoning.

Effect of POS tag category In this, we analyze the effect of the parts-of-speech (POS) tags we use to mask tokens in our S-MLM loss. We consider the following categories:

- **Base**: This consists of the POS tags used in APOLLO, i.e., ADJ, ADV, CONJ, CCONJ, PART, SCONJ, and VERB.
- **Nouns**: Here, we consider the tags referring to nouns and pronouns, i.e., NOUN, PRON, and PROPN.
- **Random**: This consists of remaining categories such as ADP, INTJ, DET, PUNCT, etc.

To study the effect of the POS tags, we incrementally add the "Nouns" and "Random" categories to the base case and evaluate the effect of pretraining using the S-MLM loss. The results of this ablation are shown in Table 5, under the section "POS Category". We observe that masking nouns and pronouns ("Nouns") leads to a significant performance drop. We attribute this drop to the fact that predicting a correct noun in a sentence would likely require more world knowledge than logical reasoning. Using the remaining categories for selective masking ("Random"), effectively making the loss function equivalent to random MLM, leads to some drop in performance as well, indicating that our set of POS tag categories is indeed more useful to learn logical reasoning.

Effect of the number of trainable layers In order to study the effect of training different numbers of parameters of the RoBERTa model, we vary the number of trainable layers of the transformer architecture between 1 and 24 (i.e., training the complete model). The results are shown in Figure 5. The blue solid line shows the performance of

APOLLO and the purple dashed line denotes the average performance of RoBERTa-Large when all layers are finetuned. From the plot, we observe that with increasing the number of trainable layers, the performance improves till layer 2, and then continues to degrade until all the layers are being trained. Prior works (Tenney et al., 2019) have shown that PLMs learn syntactic-level information in the lower layers of the transformer and semantic-level information in the upper layers. Thus, we hypothesize that the logical reasoning task initially benefits from an increasing number of trainable layers, as the semantic information needed to understand logic is being captured. But lower layers that contain the syntactic information do not benefit as much when trained using the same data as they are less related to high-level logical reasoning. The full model finetuning surprisingly performs quite well as all the model layers along with the token embeddings are being trained specifically for the logical reasoning task. But it takes significantly larger compute to finetune such a model. Overall, we find that by training the topmost two layers of the model, we are able to achieve the best performance on both datasets and hence we follow this across all variants of APOLLO.

5 Related Works

Logical Reasoning in LMs Reasoning in natural language has been a prevalent problem in NLP. In recent years, logical reasoning in textual data has seen an increasing focus. ReClor (Yu et al., 2020) and LogiQA (Liu et al., 2021) are reading comprehension-style datasets focused on questions that require reasoning using information from a given context. Prior works have predominantly used language models (Wang et al., 2022; Jiao et al., 2022) or graph neural networks (GNNs) (Huang et al., 2021; Xu et al., 2022; Li et al., 2022; Ouyang et al., 2022) to perform logical reasoning over text. Wang et al. (2022) proposed LRReasoner, which parses symbolic logical structures from the training data of ReClor for data augmentation using logical context extensions to train a PLM. Jiao et al. (2022) proposed MERIt, that used Wikipedia to generate sentence pairs for contrastive learning that are logically related, and trained the PLM using contrastive loss. DAGN (Huang et al., 2021) uses the discourse structure of the texts to perform logical reasoning using GNNs. FOCAL REASONER (Ouyang et al., 2022) constructs logical graphs using the chain of facts present in a task instance and uses GNNs to reason on the graph. GNN-based methods are not directly in scope since our main objective is to improve the logical reasoning skills of language models. Following (Jiao et al., 2022), we compare our method with two GNN-based representative methods DAGN and FOCAL REASONER.

Both LRReasoner and FOCAL REASONER use data augmentation that is specific to the task being solved, making the pretraining process specific to the downstream dataset, and thus not generalizable across tasks. While MERIt addresses this issue by using Wikipedia to generate logical graphs, their contrastive loss formulation requires counterfactual data augmentation, which potentially distorts the factual knowledge present in the pretrained model. Additionally, their approach is restricted to using Wikipedia as the data source since they heavily rely on forming entity graphs from Wikipedia texts. In contrast, we propose a simple continued pretraining strategy by modifying the masked language modeling loss (Devlin et al., 2019) and sentence classification loss to improve the logical reasoning ability of language models. Our approach is simple to integrate during pretraining, is not dependent on any data processing, and generalizes well across different datasets.

Along a related line, Clark et al. (2020) used synthetically generated data to teach PLMs to perform logical deductions over a given rule base to predict the entailment of a hypothesis. This led to some recent developments in trying to build systems that can generate step-by-step reasoning chains that demonstrate the model's reasoning process (Saha et al., 2020; Tafjord et al., 2021; Sanyal et al., 2022b). While this progress is encouraging, the use of synthetic data for training the models limits the generality of the logical reasoning skills learned by these models. Recent works have questioned if these models are indeed learning to perform logical reasoning in a robust manner or just learning some shortcuts from training data (Zhang et al., 2022; Sanyal et al., 2022a). In contrast, our method uses real-world sentences which alleviates the issue of using synthetic datasets for reasoning.

Selective masking A key step in the processing of masked language modeling loss (Devlin et al., 2019) is to determine which tokens to mask. Originally, Devlin et al. (2019) *randomly* mask 15% of tokens. Prior works have tried different techniques to select which tokens to mask. For example, ERNIE (Zhang et al., 2019) and EntityBERT (Lin et al., 2021) mask named entities to perform better knowledge-driven tasks. Other prior works (Gu et al., 2020; Lad et al., 2022) calculate the importance of words for a specific task and selectively mask the most important words. In this work, we explore the use of selective masking in the context of logical reasoning, using a novel heuristic of selecting specific POS-tagged words.

6 Conclusion

In this paper, we proposed APOLLO, an adaptive pre-trained language model with logical reasoning abilities. We use a subset of Wikipedia sentences for continued pretraining of the model using two self-supervised loss functions. The choice of the training dataset and loss functions are guided by the goal to include more reasoning-related sentences and training signals, respectively. Through experiments on two logical reasoning datasets and ablation studies, we demonstrate the effectiveness of our proposed approach. Overall, we show that APOLLO is a generalized solution to improving logical reasoning in language models.

A key advantage of APOLLO is that the pretraining steps are independent of the dataset used to train the model and the downstream task format. This opens the scope to use a larger text corpus for training such as C4 (Raffel et al., 2020). Additionally, expanding on the keywords beyond positive and negative implications (for example, conditionals such as “if-then”, “either-or”, etc.) can also benefit the training pipeline.

7 Limitation

A limitation of this approach is the trade-off between completeness and noise in the training data. While our method using keywords to extract text from Wikipedia is effective, IMPLICATION likely contains redundant sentences that cannot improve the model’s logical reasoning capability. A better rule-based or neural model might be able to extract a better corpus with potentially higher computational costs. Additionally, using POS tagging limits the application of this approach to languages with well-defined POS taggers. Switching to a more universal semantic tagging system (Abzianidze and Bos, 2017) can potentially alleviate this.

Acknowledgements

This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity(IARPA), via Contract No. 2019-19051600007, the DARPA MCS program under Contract No.N660011924033, the Defense Advanced Research Projects Agency with award W911NF-19-20271, NSF IIS 2048211, NSF SMA 1829268, and gift awards from Google, Amazon, JP Morgan, and Sony. We would like to thank all the collaborators in the USC INK research lab for their constructive feedback on the work.

References

Lasha Abzianidze and Johan Bos. 2017. Towards universal semantic tagging. In *IWCS 2017 — 12th International Conference on Computational Semantics — Short papers*.

Razieh Baradaran, Razieh Ghiasi, and Hossein Amirkhani. 2022. A survey on machine reading comprehension systems. *Natural Language Engineering*, 28(6):683–732.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuxian Gu, Zhengyan Zhang, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. 2020. Train no evil: Selective masking for task-guided pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6966–6974, Online. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don’t stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Yinya Huang, Meng Fang, Yu Cao, Liwei Wang, and Xiaodan Liang. 2021. DAGN: Discourse-aware graph network for logical reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5848–5855, Online. Association for Computational Linguistics.

Fangkai Jiao, Yangyang Guo, Xuemeng Song, and Liqiang Nie. 2022. MERIt: Meta-Path Guided Contrastive Learning for Logical Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3496–3509, Dublin, Ireland. Association for Computational Linguistics.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Tanish Lad, Himanshu Maheshwari, Shreyas Kottukkal, and Radhika Mamidi. 2022. Using selective masking as a bridge between pre-training and fine-tuning. *arXiv preprint arXiv:2211.13815*.

Xiao Li, Gong Cheng, Ziheng Chen, Yawei Sun, and Yuzhong Qu. 2022. AdaLoGN: Adaptive logic graph network for reasoning-based machine reading comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7147–7161, Dublin, Ireland. Association for Computational Linguistics.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2021. EntityBERT: Entity-centric masking strategy for model pretraining for the clinical domain. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 191–201, Online. Association for Computational Linguistics.

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

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: a robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Siru Ouyang, Zhuosheng Zhang, and hai zhao. 2022. Fact-driven logical reasoning.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

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.

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.

Soumya Sanyal, Zeyi Liao, and Xiang Ren. 2022a. Robustlr: Evaluating robustness to logical perturbation in deductive reasoning.

Soumya Sanyal, Harman Singh, and Xiang Ren. 2022b. FaiRR: Faithful and robust deductive reasoning over natural language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1075–1093, Dublin, Ireland. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8968–8975.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

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

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Siyuan Wang, Wanjun Zhong, Duyu Tang, Zhongyu Wei, Zhihao Fan, Daxin Jiang, Ming Zhou, and Nan Duan. 2022. Logic-driven context extension and data augmentation for logical reasoning of text. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1619–1629, Dublin, Ireland. Association for Computational Linguistics.

Thomas Wolf, Quentin Lhoest, Patrick von Platen, Yacine Jernite, Mariama Drame, Julien Plu, Julien Chaumond, Clement Delangue, Clara Ma, Abhishek Thakur, Suraj Patil, Joe Davison, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angie McMillan-Major, Simon Brandeis, Sylvain Gugger, François Lagunas, Lysandre Debut, Morgan Funtowicz, Anthony Moi, Sasha Rush, Philipp Schmidd, Pierric Cistac, Victor Muštar, Jeff Boudier, and Anna Tordjmann. 2020. Datasets. *GitHub. Note: https://github.com/huggingface/datasets*.

Fangzhi Xu, Jun Liu, Qika Lin, Yudai Pan, and Lingling Zhang. 2022. Logiformer: A two-branch graph transformer network for interpretable logical reasoning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 1055–1065, New York, NY, USA. Association for Computing Machinery.

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 (ICLR)*.

Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang, and Guy Van den Broeck. 2022. On the paradox of learning to reason from data.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering.

A List of Keywords

In this section, we list the set of keywords that we use to filter the entire WikiPedia data. Any sentence that contains one of the keywords is considered as part of our filtered dataset IMPLICATION. The keywords are divided into two types as described below:

- **Positive implication (Entailment)**: These keywords are present in sentences where the reason generally entails the inference. Examples of such keywords would be "therefore", "accordingly", etc. We consider the following keywords in this category: "therefore", "accordingly", "so", "thus", "consequently", "hence", "thence", "and so", "for this reason", "in consequence", "on account of", "on the "grounds", "since", "therefrom", "thereupon", "to that end", "whence", and "wherefore".
- **Negative implication (Contradiction)**: The keywords in this category are usually present in sentences where the reason contradicts the inference. For example, keywords such as "but", "although", etc., come under this category. Here, we consider the following keywords: "but", "although", "however", "nevertheless", "on the other hand", "still", "though", and "yet".

B Hyperparameter Details

In continual pretraining, we select the learning rate from the set $\{7e-6, 1e-5, 7e-5\}$, batch size 4, gradient accumulation step size from the set $\{64, 128\}$, warmup ratio 0.1, and train the model on a cluster of 8 A100 GPUs. To fine-tune a continually pretrained checkpoint, we use the training data of each dataset separately. We select learning rate from the set $\{8e-6, 1e-5, 5e-5\}$, batch size of 4, and gradient accumulation step size 1. To train the models we use a cluster of 8 A100 GPUs, which typically takes around 20 hours for the largest model.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7

☑ A2. Did you discuss any potential risks of your work?
7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B □ Did you use or create scientific artifacts?

Not applicable. 1

□ B1. Did you cite the creators of artifacts you used?
Not applicable. 1

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We use Wikipedia as the data source which is a standard practice in language model pretraining

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3.1

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix B

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix B

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4.1

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
2

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.