Can We Guide a Multi-Hop Reasoning Language Model to Incrementally Learn at each Single-Hop?

Jesús Lovón-Melgarejo ♠, Jose G. Moreno ♠, Romaric Besançon ♢
Olivier Ferret ♢, Lynda Tamine ♠
♠Université Paul Sabatier, IRIT, Toulouse, France
♢Université Paris-Saclay, CEA, List, Palaiseau, France
{jesus.lovon, jose.moreno, tamine}@irit.fr
{romaric.besancon, olivier.ferret}@cea.fr

Abstract

Despite the success of state-of-the-art pre-trained language models (PLMs) on a series of multi-hop reasoning tasks, they still suffer from their limited abilities to transfer learning from simple to complex tasks and vice-versa. We argue that one step forward to overcome this limitation is to better understand the behavioral trend of PLMs at each hop over the inference chain. Our critical underlying idea is to mimic human-style reasoning: we envision the multi-hop reasoning process as a sequence of explicit single-hop reasoning steps. To endow PLMs with incremental reasoning skills, we propose a set of inference strategies on relevant facts and distractors allowing us to build automatically generated training datasets. Using the SHINRA and ConceptNet resources jointly, we empirically show the effectiveness of our proposal on multiple-choice question answering and reading comprehension, with a relative improvement in terms of accuracy of 68.4% and 16.0% w.r.t. classic PLMs, respectively.

1 Introduction

Recent developments have shown that models based on transformers (Vaswani et al., 2017; Liu et al., 2019b) have emerged as effective soft reasoners over language (Talmor et al., 2020a; Kassner et al., 2020). To teach transformers the ability to emulate reasoning, they are trained on knowledge encoded in the form of natural language statements generally built upon explicit rules (Clark et al., 2020) or symbolic facts that refer to triples in knowledge graphs (KG) (Kassner et al., 2020). In addition, the reasoning skills of these models can successfully combine explicit natural language statements with implicit knowledge acquired during pre-training (Talmor et al., 2020b). In particular, many state-of-the-art pre-trained language models (PLMs), such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019b), have been successfully used in multi-hop reasoning problems including multi-hop question-answering (QA) tasks (Weber et al., 2019; Richardson and Sabharwal, 2020; Saxena et al., 2020; Saha et al., 2021) and multi-hop reading comprehension (RC) (Min et al., 2019; Ding et al., 2019). Training multi-hop reasoning specifically implies a two-step process: 1) distinguish -within a context- the relevant facts from the distractors to be used for reasoning; both relevant facts and distractors are generally expressed as statements in natural language using linguistic patterns (Clark et al., 2020); 2) reasoning over a sequence of relevant facts leading to chains of reasoning (Das et al., 2019). A common approach to teaching PLMs to solve a multi-hop reasoning task is to convert the structural reasoning task into subtasks that model the sequence of reasoning tasks. For instance, Richardson and Sabharwal (2020) and Clark et al. (2020) rely on a multitasking training strategy (Caruana, 1997) that uses training instances mixing different depths of reasoning steps (hops). More precisely, to teach a model solving a k-hop reasoning problem, it is trained to simultaneously solve single i-hop ($1 \leq i \leq k$) inference tasks. In the same line, Min et al. (2019) and Ding et al. (2019) carry out several steps of single-hop reading comprehension to simulate multi-hop reasoning. However, while yielding impressive results, it is still unclear if PLMs endowed with multi-hop reasoning skills really leverage the learned skills at each single-hop depth level along the reasoning chain. More specifically, our work is motivated by observing that PLMs yield unpredictable results while performing multi-hop reasoning. For instance, previous studies show that the performance of PLMs degrades substantially even with a slight increase in the number of hops in the underlying reasoning tasks (Richardson and Sabharwal, 2020). This result indicates that multi-hop models at lower depths struggle to transfer information to deeper-hop models, giving rise to the *compositionality generalization* (Chaabouni et al., 2020) issue from

Proceedings of the 29th International Conference on Computational Linguistics, pages 1455–1466
October 12–17, 2022.

simpler to complex tasks.

In this work, we advocate that a better understanding of the inherent relationships between the different single-hop reasoning models allows the design of more predictable models. We seek to answer three main questions. First, grounded in previous findings in the literature (Richardson and Sabharwal, 2020) showing the compositionality generalization issue, *do single-hop reasoning models incrementally learn?* **(RQ1)**. We construct large probe datasets using SHINRA (Sekine et al., 2018), ConceptNet (Speer et al., 2017), and RuleTakers (Clark et al., 2020) using single-depths of inference to train and probe single-hop models and compare their performance. Overall, our findings confirm the prevalence of the compositionality generalization issue from complex to less complex multi-hop reasoning tasks. Second, inspired by the human reasoning style to solve complex problems based on simpler ones (Anderson, 1980), *can PLMs be guided toward incremental reasoning?* **(RQ2)**. Specifically, we propose a generic and automatic methodology for generating training probe datasets that endow PLMs with reasoning capabilities over a sequence of single-hop steps. We particularly investigate the impact of using distractor generation strategies. Our empirical results show that we can guide PLMs to incrementally reason by leveraging classic approaches with a gain of up to 7.98 accuracy. These training datasets are publicly available[1]. Finally, grounded on previous findings revealing that PLMs trained on one specific reasoning task improve their performance on different and unrelated reasoning tasks (Talmor et al., 2020b), *do QA tasks leverage incrementally trained reasoning models?* **(RQ3)**. For the multi-hop QA task, in particular, the results show that our approach quickly adapts to obtain an accuracy of 54.74 compared to 52.03 from state-of-the-art

2 Methodology

In this section, we first introduce the basic definitions and notations used in our proposal and then present the data probe generation methodology.

2.1 Task Definition

We focus on the multi-hop symbolic reasoning task over explicit knowledge. Following previous work, our setting includes the following:

1) a **knowledge graph** (KG) $G = (\mathcal{E}, \mathcal{R})$ with entities as nodes $(e \in \mathcal{E})$, inference relationships as edges $(r \in \mathcal{R})$, and a set of real relation *facts* f_{ij} as positive triples (e_i, r, e_j) denoted F^+ among all the possible ones in $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$;

2) a **hypothesis** $\mathcal{H}_{ij}^k$ about the relationship $r*$ between two target entities $(e_i, r*, e_j)$ separated by k hops in graph G;

3) a **hypernym inference path** of depth k on G, referred to as $\mathcal{I}_{ij}^k$, allowing to build a new relation fact $f_{ij}^* \notin F^+$ by combining $k+1$ relation facts along the reasoning chain $< \ (e_i, r_0, e_1)\,(e_1, r_1, e_2)\ \dots (e_k, r_k, e_j)\ >$, such as $\forall 0 \leq n \leq k$, $(e_n, r_n, e_{n+1}) \in F^+$, $(e_i, r_1, e_1), (e_k, r_k, e_j) \in F^+$;

4) a **context**, composed of *relevant facts* $\mathcal{F}_{ij}^* \subset F^+$, defined by the facts that form the hypernym inference chain $\mathcal{I}_{ij}^k$ and distractors $\mathcal{D}_{ij}$, defined by triplets that do not form the hypernym inference in $\mathcal{I}_{ij}^k$.

Given a hypothesis in context $< \mathcal{H}_{ij}^k, (\mathcal{F}_{ij}^*, \mathcal{D}_{ij}) >$, the task consists in inferring its truth value. A hypothesis $\mathcal{H}_{ij}^k$ is either *true* if it deductively follows a hypernym inference $\mathcal{I}_{ij}^k$ from the context $(\mathcal{F}_{ij}^*, \mathcal{D}_{ij})$, or *false* if it does not (under the close-world assumption).

2.2 Data Probe Generation

Given a knowledge graph G, we propose a generic dataset generation methodology to probe multi-hop reasoning PLMs in a single-hop setup. We define two generation functions to construct the input $< \mathcal{H}_{ij}^k, (\mathcal{F}_{ij}^*, \mathcal{D}_{ij}) >$: i) $\textsc{Hyp}(G, k)$, to generate both the hypothesis $\mathcal{H}_{ij}^k$ and the related inference path $\mathcal{I}_{ij}^k$; and ii) $\textsc{Distr}(G, \mathcal{I}_{ij}^k, \mathcal{H}_{ij}^k)$, to generate a set of distractors $\mathcal{D}_{ij}$ with respect to the inference path $\mathcal{I}_{ij}^k$.

Hypothesis Generation $\textsc{Hyp}(G, k)$. First, we apply the Depth First Search (DFS) algorithm to visit all entities of knowledge graph G, generating a set of paths of length $k+1$, excluding the root, used as inference paths. For the true hypothesis, we create $\mathcal{H}_{ij}^k$ with the form (e_i, r_k, e_j) using the first and last facts from $\mathcal{I}_{ij}^k$ (see Figure 1, which illustrates examples of 1-hop and 2-hop hypothesis). Unlikely, for the false hypothesis, we simply generate a hypothesis $\mathcal{H}_{iz}$ replacing the last real fact of the inference path by $(e_k, r_k, e_z) \notin F^+$.

[1] https://github.com/jeslev/incremental_reasoning

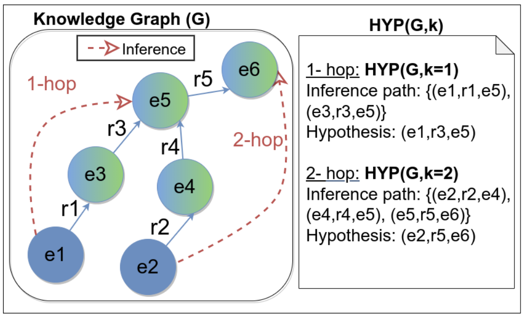


Figure 1: Hypothesis and inference path generation from a knowledge graph for 1-hop and 2-hop reasoning depths.

Distractor Generation $\textsc{Distr}(G, \mathcal{I}_{ij}^k, \mathcal{H}_{ij}^k)$. We generate object, relationship, and inference distractors for each hypothesis $\mathcal{H}_{ij}^k$ (as shown in Figure 3b).

An *object distractor* is generated by sampling a fact with the form $(., ., e_j)$, e_j referencing the last entity in the hypothesis. Similarly, a *relationship distractor* is a sampled fact with the form $(., r_k, .)$.

Inference distractors. Finally, we generate *inference distractors* in such a way that they exploit evidence from the structure of the inference path $\mathcal{I}_{ij}^k$ by linking two of its facts with a pivot element. Having in mind the goal of guiding a k-hop model to perform incremental inference over single-hops, we propose distractor strategies that either improve the entity representations or bridge between entities by transferring information along with *intermediate hops* necessary to complete the reasoning. More precisely, based on a recent finding (Kassner and Schütze, 2020) showing that fine-tuned PLMs are good for recognizing false facts, we assume that distractors have a hidden impact on the reasoning task. While most common approaches attempt to improve PLMs entity representations by enriching the context-based relevant facts, we believe distractors can significantly leverage PLMs entity representations and thus the reasoning performance. Thus, we investigate the rationale behind this assumption by designing the following strategies for generating inference distractors: **shared (s-inf)**, which uses the same distractor entity ($x = y$) of the two consecutive facts from the inference path, and the **individual (i-inf)**, that uses different distractor entities ($x \neq y$). Figure 2 shows the implementation P-INF for both inference distractors, with the variable shared$= True$ for (s-inf) and shared$= False$ for (i-inf).

Additionally, we explicitly guide a k-hop model to perform incremental inference using a **guided distractor (g-inf)** that connects the two consecutive facts in the inference path (see Figure 2 algorithm G-INF). The key underlying idea is to drive the PLM to *incrementally reason* over the inference path by transferring information between entities along *intermediate hops* of a multi-hop reasoning path. Figure 3b illustrates the different distractors generated for a specific example.

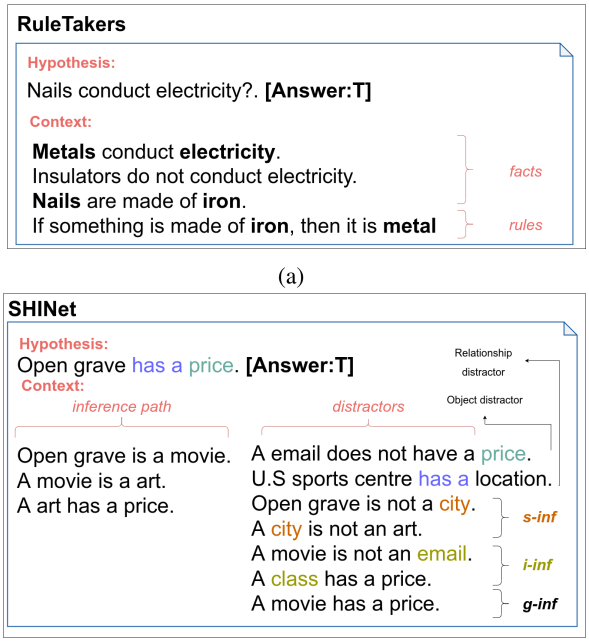
```
P-INF ($\mathcal{I}_{ij}^k$,k, shared)
$AD = \{\mathcal{E} \times \mathcal{R} \times \mathcal{E}\} \smallsetminus F^+$
$D = \emptyset$
for $i \in 1 \dots k+1$ do
    $(e_x, \_, \_)$= $\mathcal{I}_{ij}^k$[i]
    $(\_, \_, e_{x+2})$= $\mathcal{I}_{ij}^k$[i+1]
    $L1 = \emptyset, L2 = \emptyset$
    for $(e_x, r_a, p) \sim AD(e_x)$ do
        $L1 = L1 \cup \{(e_x, r_i, b_j)\}$
    end for
    for $(q, r_b, e_{x+2}) \sim AD(e_{x+2})$ do
        $L2 = L2 \cup \{(e_i, r_j, e_{x+2})\}$
    end for
    for $(d_1, d_2) \sim L1 \times L2$ do
        $(e_x, r_a, x)$=$d_1$
        $(y, r_b, e_{x+2})$=$d_2$
        if $x = y$ and shared then
            $D = D \cup \{d_1, d_2\}$
            break
        else if $x \neq y$ and  not shared then
            $D = D \cup \{d_1, d_2\}$
            break
        end if
    end for
end for
return $D$

G-INF ($\mathcal{I}_{ij}^k$,k)
D = P-INF($\mathcal{I}_{ij}^k$,k, True)
for $i \in 1 \dots k+1$ do
    $(e_x, \_, \_)$= $\mathcal{I}_{ij}^k$[i]
    $(\_, r, e_{x+2})$= $\mathcal{I}_{ij}^k$[i+1]
    $D = D \cup \{(e_x, r, e_{x+2})\}$
end for
return $D$

S-INF ($\mathcal{I}_{ij}^k$,k)
D = P-INF($\mathcal{I}_{ij}^k$,k, True)
return $D$

I-INF ($\mathcal{I}_{ij}^k$,k)
D = P-INF($\mathcal{I}_{ij}^k$,k, False)
return $D$
```

Figure 2: Pseudocode for distractors generation in a k-hop dataset. D, L1, and L2 are lists used to stock the generated distractors. AD stands for “available distractors”, considering all the possible triples in the KG not used in the inference. AD(e) represents a filtered AD where e is present.

3 Experimental Setup

3.1 Generated Dataset Probes

We present here the dataset probes, namely Single-RuleTakers (S-RT) and SHINet, we automatically constructed using the previously presented generation functions (see Section 2.2). These datasets are based on three different publicly available resources: the RuleTakers dataset (Clark et al., 2020),


RuleTakers
Hypothesis:
Nails conduct electricity?. **[Answer:T]**
Context:
Metals conduct **electricity**.
Insulators do not conduct electricity.
Nails are made of **iron**.
If something is made of **iron**, then it is **metal**
facts
rules

(a)

SHINet
Hypothesis:
Open grave has a price. **[Answer:T]**
Context:
Relationship distractor
Object distractor
inference path
distractors
Open grave is a movie.
A movie is a art.
A art has a price.
A email does not have a price.
U.S sports centre has a location.
Open grave is not a city.
A city is not an art.
A movie is not an email.
A class has a price.
A movie has a price.
s-inf
i-inf
g-inf

(b)

Figure 3: RuleTakers (a) and SHINet 2-hop (b) examples.

SHINRA (Sekine et al., 2018), a knowledge graph manually built upon a structured taxonomy, and ConceptNet (CN) (Speer et al., 2017), another KG widely used in NLP tasks (Talmor et al., 2020b; Ma et al., 2021).

3.1.1 The Single RuleTakers Dataset (S-RT)

In the original RuleTakers dataset, each entry has a small theory representing the context ($\mathcal{F}^{*}_{ij}$), and a True/False question representing the hypothesis $\mathcal{H}^{k}_{ij}$, mainly grouped on five variations $k = 0$, and $D \leq k$ with $k = \{1, 2, 3, 5\}$ with questions requiring reasoning up to depths $0, 1, 2, 3, 5$ respectively. An example of a true hypothesis in the RuleTakers dataset is presented in Figure 3(a). We filtered these datasets to construct our probes, single k-hop datasets with $k \in \{0, 1, 2, 3, 5\}$ for train and test splits, called *S-RT* dataset.

3.1.2 SHINet Dataset

The RuleTakers dataset presents the context as a paragraph, with no annotations on the relevant facts or the distractors, making it difficult to measure their impact on the inference process. To overcome this limitation, we created the SHINet dataset built upon the public SHINRA dataset. The SHINRA contains facts with the form (e_i, is-a, e_j), limited to the "is-a" relation. Having in mind that the inference task heavily relies on the range of relationships and objects that the model has seen in the training phase (Wang et al., 2021), we created *SHINet* dataset by sampling from SHINRA and ConceptNet based on a manual alignment of the intermediate nodes of SHINRA. We enrich the single "is-a" relationship with the facts from ConceptNet in the form (e_j, r', p_j). The alignment relies on manual verification of finding e_j in both datasets. An example of a true hypothesis in the SHINet dataset with related distractors *s-inf*, *i-inf*, and *g-inf* is presented in Figure 3(b). Table 1 summarizes our generated datasets.

Dataset	train	dev	test
1-hop (*s-inf*)	35,000	1,200	2,074
2-hop (*s-inf*)	35,000	1,200	5,994
2-hop (*i-inf*)	35,000	1,200	-
2-hop (*g-inf*)	35,000	1,200	-

Table 1: Number of samples for train/dev/test splits for each generated dataset.

3.2 Model Training

We used PLMs trained on the single-hop training partitions of our generated datasets. It is worth mentioning that this training protocol differs from the protocol used in previous approaches, where mixed datasets $\{0 \leq i\}$-hop datasets are simultaneously used for training multi-hop reasoning models based on a multitasking approach (Richardson and Sabharwal, 2020). More specifically, we exploited the following: 1) we used several pre-trained LMs based on BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), and RoBERTA (Liu et al., 2019b); even using similar architectures, they showed significant differences in performance results, especially on reasoning tasks (Talmor et al., 2020a); 2) as a building block, we used the training protocol proposed in previous work (Talmor et al., 2020b), removing the first fact from $\mathcal{I}^{k}_{ij}$ in 40% of the samples. Using this training protocol, we can provide insights into both the intrinsic strengths and limits of our proposed PLM since we exploit a recent state-of-the-art PLM that captures rich semantic information.

3.3 Implementation Details

Each fact (e_i, r, e_j) can be expressed as a statement in natural language using linguistic patterns referred to as *fact templates* (e.g., e_i *is a* e_j). We make use of the Hearst Patterns templates (Hearst, 1992; Roller et al., 2018) and the ones proposed in Talmor et al. (2020b) as fact templates. Then, we train a transformer-based model with a set of input sequences of tokens with the following schema: "[CLS] Context [SEP] Hypothesis [SEP]". Then, we used the output representation of the [CLS] token and projected it into a binary classifier layer to obtain the probabilities that the hypothesis is true or false. For all of the models, we used the transformers' public implementation from HuggingFace (Wolf et al., 2020). Main hyperparameters were set following standard setup or original authors' recommendations. In all the experiments where SHINet is used for training, we set a maximum word length to 256, batch size to 4, number of steps per batch to 729, number of epochs to 4, and Adam as optimizer with learning rate to 1e-5 and weight decay to 0.1. In the case of fine-tuning, each dataset uses its default hyperparameters. However, the parameters remain the same for each dataset regardless of the fine-tuning order. For the QA tasks evaluation, we opted for the same parameters as when fine-tuning the SHINet datasets, except for the loss (categorical cross entropy), and the number of steps per batch is set to 2,163. We trained and evaluated the models with 10 different random seeds and presented mean scores in our comparisons (see Appendix A for computational costs). To provide statistical significance to our results, we applied a test for Almost Stochastic Dominance (Dror et al., 2019) between test score distributions, using $\alpha = 0.05$.

4 Results and Discussion

To address **RQ1** and **RQ2**, we used the k-hop S-RT with $k \in \{0, 1, 2, 3, 5\}$ and the $\{D \leq 2\}$-hop, 1-hop, and 2-hop SHINet dataset. All the SHINet datasets are composed of the object, relationship, and inference distractors. Regarding the inference distractors, we evaluated with the three strategies: *i-inf, s-inf, g-inf*. The default setting uses the *(s-inf)* strategy unless it is explicitly mentioned otherwise. Note that for each SHINet strategy-based dataset, we have train and test partitions. As recommended in the literature (Elazar et al., 2021; Sakaguchi et al., 2020), to avoid biases in these datasets that lead to an overestimation of the reasoning capabilities of PLMs, we applied on the test partitions the AFLITE algorithm (Sakaguchi et al., 2020) that finds machine-detectable embeddings associations to reduce biases. We used optimal parameters after grid-search: n (classifiers) = 64, m(samples) = 1000, top-k = 200, threshold = 0.75. Comparing the original and filtered datasets, we approximately filtered 45% of the total samples. To address **RQ3**, we used the MCQA (Richardson and Sabharwal, 2020), and RACE (Lai et al., 2017). MCQA is composed of 193,000 entries. Each entry is composed of a question and five possible answers, including reasoning tasks such as hypernymy, hyponymy, synonymy detection, and word sense disambiguation. RACE (Lai et al., 2017) consists of nearly 28,000 passages and 100,000 questions divided into Middle and High School sets and up to four possible answers.

Test (k-hop)	**k=0**	**k=1**	**k=2**	**k=3**	**k=5**
Train ↓ Models →			*RoBERTa*		
0-**hop**	**99.99**	*43.51*	*26.52*	*22.94*	*12.78*
1-**hop**	**90.11**	**98.16**	50.64	*37.30*	*23.07*
2-**hop**	66.92	64.62	**88.54**	**91.65**	**96.16**
3-**hop**	68.36	64.44	**88.47**	**91.35**	**96.11**
5-**hop**	63.32	63.11	**87.09**	**89.92**	**95.06**

Table 2: Accuracy performance (in %) for a RoBERTa model trained on k-hop S-RT training set (rows) and tested on k-hop S-RT test set (columns). For a better reading, scores worse than random ($< 50\%$) are in *italic* and good results ($> 80\%$) are in **bold**.

Test (k-hop)	**k=1**	**k=2**	**k=1**	**k=2**	**k=1**	**k=2**
Train ↓ Models →	*XLNet*		*BERT*		*RoBERTa*	
Mixed	98.89	89.06	**99.23**	**94.63**	99.71	93.44
1-hop	**99.71**	86.00	98.96	89.75	**99.80**	96.23
2-hop	96.31	**99.82**	77.75	87.25	98.77	**99.99**

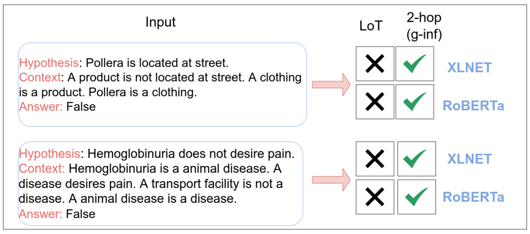
Table 3: Accuracy performances (in %) for mixed, 1-hop and 2-hop models using the SHINet dataset. The highest values are in **bold**.

4.1 Do Single-Hop Reasoning Models Incrementally Learn? (RQ1)

To answer **RQ1**, we train separately single i-hop reasoning models using the S-RT dataset ($i \in \{0, 1, 2, 3, 5\}$)[2], and we train PLMS using the SHINet dataset on single 1-hop and 2-hop, and the *Mixed* models trained on the $\{k \leq 2\}$-hop SHINet dataset.

Table 2 and Table 3 report, respectively, the accuracy scores for the different single i-hop models using the S-RT dataset, and the accuracy scores of PLMs trained on 1-hop and 2-hop SHINet dataset, as well as the *Mixed* model. We take an empirical approach by assuming that incremental learning is observed when the models generalize from complex to less complex tasks. Overall, we can observe

[2] We keep the model, hyperparameters, and setting as proposed in (Clark et al., 2020).



Hypothesis: Pollera is located at street. Context: A product is not located at street. A clothing is a product. Pollera is a clothing. Answer: False

Hypothesis: Hemoglobinuria does not desire pain. Context: Hemoglobinuria is a animal disease. A disease desires pain. A transport facility is not a disease. A animal disease is a disease. Answer: False

Figure 4: Two examples from the 2-*hop* test set. Both examples are negative (false hypothesis), the first with a positive phrase and the second one with a negative phrase. The guided model correctly predicted both.

that when trained with larger hop depths, models struggle to solve even slightly less large reasoning tasks. Regarding specifically the *S-RT* dataset, it can be seen from Table 2 (green area), that the performance of the model trained and tested on the 2-hop (88.54) decreases to 66.92 and 64.62 respectively when tested on the 0-hop and 1-hop data. Similar behavior is observed for the model trained on 3-hop. Looking at Table 3, obtained using the SHINet dataset, we can see that the results on the 2-hop test show a similar trend with the *S-RT* dataset: overall, the 2-hop PLMs exhibit better results when tested on 2-hop, but their performances decrease for a simpler task, 1-hop test, for all the three models while we expect at least stable performance. More precisely, we observe a performance decrease of $99.82 \downarrow 96.31$, $87.25 \downarrow 77.75$, and $99.99 \downarrow 98.77$ for XLNet, BERT, and RoBERTa, respectively. The same performance decrease trend is observed compared to the upper bound achieved when testing the 1-hop trained PLMs on the 1-hop test. This might reveal a counter-intuitive and uncontrollable behavior: having in mind the incremental human-style reasoning (Anderson, 1980), one could argue that the ability to solve a k-hop problem implies the ability to solve the $\{k\text{-}1\}$-hop one, but results indicate the contrary. These results are consistent in both datasets suggesting that *PLMs do not incrementally learn by accumulating knowledge*.

In addition, looking at the compositionality generalization from simple to complex tasks, we can see from Table 2 (S-RT) that the performance of models trained on low-depth single-hops (rows $k = 0, 1$) significantly decreases when the hop is deeper (columns $k = 2, 3, 5$) in the test set (e.g., the 1-hop model performance decreases from 90.11 to 50.64 and 37.30 for columns $k = 2, 3$, respectively). However, for depth rows $k = 2, 3, 5$, this trend is less clear. Similarly, Table 3, using the SHINet dataset, shows that 1-hop models manage to obtain strong accuracy scores, over 86.00 in all datasets, indicating that they can deal with their tasks and complex ones. This behavior can be explained by the mix of implicit knowledge (from pretraining) and explicit knowledge (from training), filling the logic gap between tasks as shown by (Talmor et al., 2020b). Moreover, we can observe that RoBERTa-based and XLNet-based PLMs are more effective in both 1-hop and 2-hop configurations, in contrast to BERT-based models, consistently with previous work (Talmor et al., 2020a).

4.2 Can Reasoning Models be Guided Toward Incremental Reasoning? (RQ2)

To answer **RQ2**, we compared our models to two different baselines: the *Mixed* model used in RQ1, and the *LoT* model from Talmor et al. (2020b). *LoT* is trained on a $\{k \leq 1\}$ using ConceptNet, WordNet, and Wikidata datasets to combine implicit knowledge acquired in pre-training with explicit rules and facts showing good performance on various types of reasoning tasks. We also trained the *hybrid* model by fine-tuning on the SHINet 2-hop (g-inf) dataset followed by the *LoT* train dataset (1-hop) to show the effect of jointly leveraging implicit knowledge, explicit knowledge (*LoT*), and incremental reasoning. We report in Table 4 the mean accuracy scores of the different distractors.

At a first glance, we can see that our proposed guided model 2-*hop (g-inf)* surpasses all its counterparts for 2 out of 3 settings, namely XLNet and RoBERTa. More precisely, by comparing the performance scores of 2-*hop (g-inf)* to 2-*hop (s-inf)*, we can fairly assess the positive impact of our proposed inference distractors presented in Figure 2 to guide the training toward incremental learning across all the models. For instance, we observed an improvement of 3.53 (0.17), 6.41 (7.98), and 1.17 (0.01) when comparing the 2-*hop (g-inf)* model with 2-*hop (s-inf)* models on the 1-*hop* (2-*hop*) tests with XLNet, BERT, and RoBERTa respectively. We further compare the accuracy scores of 2-*hop (g-inf)* in comparison to 2-*hop (i-inf)* to show the impactful role of the (s-inf) inference distractor to improve the reasoning inference. As it can be seen from Table 4, 2-*hop (g-inf)* increases the performance on both tests by a difference greater than 21.0 in all models. The comparison between the 2-*hop (g-inf)* model to a traditional multi-hop mixing strategy *Mixed* shows the advantage of the

Test (k-hop)	**k=1**	**k=2**	**k=1**	**k=2**	**k=1**	**k=2**
Models →	*XLNet*		*BERT*		*RoBERTa*	
Train ↓						
LoT	98.37∗	99.17	86.28∗	**95.33**∗	99.15∗	98.96
Mixed	98.89	89.06	**99.23**	94.63	99.71	93.44
2**-hop**						
(i-inf)	60.81	56.77	62.37	57.16	60.48	56.23
(s-inf)	96.31	99.82†	77.75	87.25	98.77	99.99†
(g-inf)	**99.84**∗†	**99.99**∗†	84.16∗	95.23∗	**99.94**∗†	**100**∗†
hybrid	99.18∗†	99.67∗†	86.32∗†	93.85∗	99.31∗†	99.76†

Table 4: Accuracy performances (%) for 2-hop models by varying the inference distractor in the SHINet dataset. † and ∗ indicates statistical significance according to the Almost Stochastic Dominance test over *LoT* and 2-*hop (s-inf)*, respectively.

Train ↓ / Test(k-hop) →		$k = 0$	$k = 1$	$k = 2$
S-RT	2-hop	66.92	64.62	88.54
	3-hop	68.36	64.44	88.47
LoT	2-hop	72.61	65.37	88.40
	3-hop	72.46	64.81	88.56
2**-hop** *(s-inf)*	2-hop	67.15	64.88	88.42
	3-hop	67.06	64.58	88.25
2**-hop** *(g-inf)*	2-hop	68.06	65.18	88.05
	3-hop	67.89	64.8	88.11
hybrid	2-hop	71.75	65.49	88.44
	3-hop	**72.47**	**65.14**	**88.60**

Table 5: Accuracy comparing different reasoning models on the S-RT dataset. The best result for each test in underlined and **bold** for 2-hop and 3-hop models, respectively.

incremental inference for most of the settings. Finally comparing our proposed guided model 2-*hop (g-inf)* to the fine-tuned multi-hop strategy *hybrid*, we can observe that our guided model surpasses the hybrid model and *LoT* in 2 out of 3 models, namely (*XLNet* and *RoBERTa*). It is worth noting that this performance is achieved using fewer computational resources; the model can address both tests 2-hop and the simpler 1-hop in an incremental reasoning fashion.

In Figure 4, we show some hand-picked difficult examples from the 2-*hop* test set for the *LoT* model that are especially helped by the guided model 2-*hop (g-inf)*, using XLNet and RoBERTa. Specifically, we observed a positive impact of the distractors to solve false hypothesis examples using a negative phrase.

Additionally, we compare our proposed models with those trained on the *S-RT* dataset used in **RQ1**. We particularly examine if the proposed models still exhibit the observed phenomenon highlighted in the green area from Table 2. Table 5 shows results with mean values after 3 runs and under the *inoculation* technique. The *inoculation technique* from Liu et al. (2019a) was used to avoid overriding the knowledge acquired in our models. The inoculation consists of using a small amount of training data to solve new tasks, overcoming the mismatch between the datasets used in training and fine-tuning (Jiang et al., 2020).

Model	**Def**	**Hype**	**Hypo**	**Syn**	**DQA**	**Avg (%Imp)**
Rand	19.90	19.90	20.20	19.80	20.00	19.96 (-38.6%)
RoB	40.10	32.65	23.15	34.41	32.26	32.51 (-)
LoT	72.41	43.83	40.54	51.85	51.53	52.03 (+60.0%)
1**-hop**						
(s-inf)	62.65	52.69	38.58	45.80	43.68	48.68 (+49.7%)
2**-hop**						
(s-inf)	63.65	46.50	36.82	50.52	47.69	49.04 (+50.8%)
(g-inf)	70.86	**51.86**	43.03	**55.29**	52.65	**54.74** (+68.4%)
hybrid	**73.09**	44.70	**46.43**	51.87	**54.72**	*54.16* (+66.6%)

Table 6: Accuracy (%) scores for baselines and multi-hop reasoning models using the validation set of the MCQA dataset. Improvement percentages (%Imp) are given w.r.t. **RoB**ERTa (inoculated).

We did a preliminary analysis of the learning curves for each task to determine the right amount of data to use (see Appendix C).

We can see from Table 5 that even when the inoculation is used, the models relying on incremental reasoning (2-*hop (g-inf)* and *hybrid*) overpass the baseline results (*S-RT* and 2-*hop (s-inf)*). Particularly, we see that guiding the model training over hops leads to improvements in lower hop levels ($k = 0, 1$) compared to traditional model training with mixed hops. For instance, for the test $k = 1$, the guided model 2-*hop (g-inf)* improves by 0.56 and 0.36 the model trained with S-RT on hops $k = 2$ and $k = 3$, respectively.

4.3 Do QA Tasks Leverage Incrementally Trained Reasoning Models (RQ3)?

To answer **RQ3**, we used: 1) two QA tasks, namely Multiple Choice Question Answering (MCQA), and Reading Comprehension (RC). We applied the inoculation technique presented before to all the models; 2) the *Random* model, denoted *Rand*, the *RoBERTa* model, denoted *RoB*, and the *LoT* model as baselines. The *RoBERTa* model has been chosen, given its performance superiority as shown in the previous experiments (see Sections 4.1 and 4.2). For datasets examples and illustrations of the tasks, we refer the reader to Appendix B.

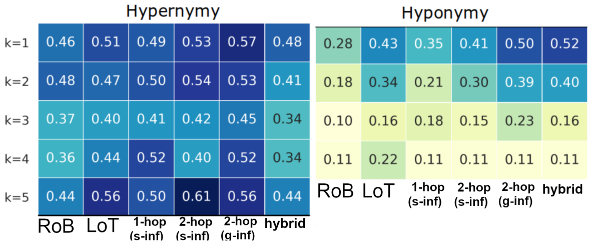


Figure 5: Accuracy values using the hypernymy and hyponymy subsets broken into number of hops k (rows) for the models (columns).

Multiple Choice Question Answering (MCQA).

For MCQA, we re-used a publicly available code[3] as Richardson and Sabharwal (2020) and then applied the inoculation technique (Liu et al., 2019a). We plot the learning curves of each probe for the average of five different runs with random subsets (see Appendix C).

In Table 6, we report the results of our inoculated models. We can see that our models 2-*hop (g-inf)* and *hybrid* achieve the best average performance scores over all the baselines.

To deepen our analysis of the reasoning over increasing numbers of hops, we experimented with our models with the hypernymy and hyponymy subsets, up to 4 and 3 hops levels, respectively. By filtering the numbers of hops, we report the performance variation of our models in Figure 5. We can see that for the hypernymy (resp. hyponymy), the *hybrid* (followed by 2-*hop (g-inf)*) model outperforms all models in all depths but for $k = 4, 5$. Furthermore, we interestingly observe a positive trend toward reducing the performance decrease rate between hop levels when using our proposed guided training approach. For instance, when comparing levels $k = 2$ and $k = 4$, we observe that performance decrease is reduced from 0.14 to 0.01 for 2-*hop (s-inf)* and 2-*hop (g-inf)* respectively. Similarly between the hyponymy levels $k = 2$ and $k = 3$ we can see a performance decrease reduced from 0.18 to 0.16 for *LoT* and 2-*hop (g-inf)* models respectively. This observation clearly indicates the positive impact of incremental reasoning on performance.

Reading Comprehension (RC). For RC, results under inoculation conditions are reported in Table 7. As can be seen, most of the models behave similarly for the Middle set, with 2-*hop (s-inf)* as the most performing model. On the contrary, we can observe a clear improvement for all models on the High set when compared to *RoBERTa*. In this case, the most performing model is 2-*hop (s-inf)* (69.76) closely followed by 2-*hop (g-inf)* (68.72) and *hybrid* (68.56). Therefore, chains of reasoning seem to be a key component of the solution, even if most of the studied multi-hop models correctly capture the needed knowledge.

Models	**Middle (%Imp)**	**High (%Imp)**
RoBERTa	77.18 (-)	59.22 (-)
LoT	77.04 (-0.2%)	68.68 (+16.0%)
1-**hop** ***(s-inf)***	76.56 (-0.8%)	68.94 (+16.0%)
2-**hop** ***(s-inf)***	**77.32** (+0.2%)	**69.76** (+17.8%)
2-**hop** ***(g-inf)***	75.65 (-2.0%)	68.72 (+16.0%)
hybrid	76.37 (-1.0%)	68.56 (+15.8%)

Table 7: Accuracy (%) comparing different reasoning models on the RACE dataset for middle school and high school. Improvement percentages (%Imp) are given w.r.t. RoBERTa.

Finally, we compare the results from Table 6 and Table 7. We observe that model scores are very close on the RC task, even when using different distractors and the number of hops. The uniformity between all models' performances suggests that multi-hop reasoning is not a key component in solving these questions.

5 Related Work

Our main study focused on multi-hop reasoning. Recent studies have proposed solutions using decomposition-aggregation approaches (Min et al., 2019) by combining or extending different model architectures to leverage reasoning performance (Feng et al., 2020; Yasunaga et al., 2021; Bauer et al., 2018), creating explained reasoning paths (Ding et al., 2019) or using chain of thought prompting (Wei et al., 2022). In contrast, we focus on leveraging the inner reasoning skills of PLMs, benefiting from their internal architecture and knowledge captured in pre-training. We argue that our results may be a solid alternative to a standard PLM in this kind of work.

There are recent demonstrations that trained PLMs can perform simple reasoning tasks (Talmor et al., 2020a). Even if these models are not naturally good solvers of complex tasks such as multi-hop reasoning (Kassner et al., 2020), they are capable of learning when trained on such tasks (Clark et al., 2020; Richardson and Sabharwal, 2020). However, these training setups propose mixing different depth levels of reasoning, leading to unpredictable results, and, thus, a lack of

[3] `https://github.com/yakazimir/semantic_fragments`

model interpretability. They do not let recognize the knowledge captured at each hop level and whether acquired knowledge, if any, is actually reused to solve higher-hop level reasoning. We proposed a single-hop design that lets us analyze the actual contribution of each reasoning level.

Although our work is inspired by the previous literature, it is different from Talmor et al. (2020a) and Talmor et al. (2020b), as they evaluate the inner reasoning capabilities of PLMs in simple reasoning tasks, but we evaluate incrementally trained PLMs for multi-hop reasoning performed on NLP downstream tasks. Similarly, Kassner et al. (2020) evaluate the model reasoning skills controlling the data given in pre-training. In contrast, we analyze how the training data and elements in the context affect the task in a multi-hop scenario.

6 Conclusions

Transformers have been recently gaining increasing attention for reasoning tasks over language. In this paper, we have specifically studied whether we can endow PLMs used in multi-hop reasoning tasks with the ability to incrementally acquire knowledge by following the inference path over the sequence of hops. Our underlying objective is to control the training of PLMs better, leading to more understandable and predictable multi-hop reasoning models. In particular, we have complemented previous findings in the literature by showing that PLMs trained on 1-hop reasoning tasks can extrapolate the reasoning to 2-hops but that 2-hop reasoning models struggle to generalize over slightly simpler 1-hop tasks. Keeping in mind the human-style reasoning from simpler to complex tasks, we advocate incremental reasoning over the structure of the inference path as a step forward. We provide a training data generation strategy that relies critically on inference distractors connecting intermediate relevant facts in the reasoning path. By applying our approach, our models achieve higher or similar performance trends than fine-tuning multi-hop models but consume fewer resources. Furthermore, we show that the incrementally trained multi-hop PLMs are transferable to other QA-based tasks.

Although our experimental settings are limited to low depths of inference ($k = 1, 2$), our findings show both the feasibility and the benefit of incremental reasoning and open new research opportunities. We may potentially extend this work toward the benchmarking of multi-hop reasoning interpretability with the design of baseline models, dataset generation strategies with upper bounds, and evaluation metrics including, but not limited to, inference path recall.

7 Acknowledgments

This work was supported by the ANR-19-CE23-0028 MEERQAT project. This work was granted access to the HPC resources of IDRIS under the allocation 20XX-AD011012638R1 made by GENCI.

References

John R. Anderson. 1980. *Cognitive Psychology and Its Implications*. W.H.Freeman & Co Ltd.

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for Generative Multi-Hop Question Answering Tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230, Brussels, Belgium. Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Mach. Learn.*, 28(1):41–75.

Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. 2020. Compositionality and generalization in emergent languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4427–4442, Online. Association for Computational Linguistics.

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

Rajarshi Das, Ameya Godbole, Manzil Zaheer, Shehzaad Dhuliawala, and Andrew McCallum. 2019. Chains-of-reasoning at TextGraphs 2019 shared task: Reasoning over chains of facts for explainable multihop inference. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 101–117, Hong Kong. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2694–2703, Florence, Italy. Association for Computational Linguistics.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep dominance-how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785.

Yanai Elazar, Hongming Zhang, Yoav Goldberg, and Dan Roth. 2021. Back to square one: Artifact detection, training and commonsense disentanglement in the Winograd schema. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10486–10500, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309. Association for Computational Linguistics.

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

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

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*, Online. Association for Computational Linguistics.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019a. Inoculation by fine-tuning: A method for analyzing challenge datasets. 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 2171–2179, Minneapolis, Minnesota. Association for Computational Linguistics.

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:1907.11692 [cs]*. ArXiv: 1907.11692.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2021. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In *35th AAAI Conference on Artificial Intelligence*.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *ACL*.

Kyle Richardson and Ashish Sabharwal. 2020. What Does My QA Model Know? Devising Controlled Probes using Expert. *Transactions of the Association for Computational Linguistics*, 8(0):572–588. Number: 0.

Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst Patterns Revisited: Automatic Hypernym Detection from Large Text Corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–363, Melbourne, Australia. Association for Computational Linguistics.

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, Online. Association for Computational Linguistics.

Satoshi Sekine, Akio Kobayashi, and Kouta Nakayama. 2018. SHINRA: Structuring Wikipedia by Collaborative Contribution. In *1st Conference on Automated Knowledge Base Construction (AKBC 2019)*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020a. oLMpics-On What Language Model Pre-training Captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020b. Leap-Of-Thought: Teaching Pre-Trained Models to Systematically Reason Over Implicit Knowledge. *Advances in Neural Information Processing Systems*, 33.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Peifeng Wang, Filip Ilievski, Muhao Chen, and Xiang Ren. 2021. Do Language Models Perform Generalizable Commonsense Inference? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3681–3688, Online. Association for Computational Linguistics.

Leon Weber, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, and Tim Rocktäschel. 2019. Nlprolog: Reasoning with weak unification for question answering in natural language. *CoRR*, abs/1906.06187.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546. Association for Computational Linguistics.

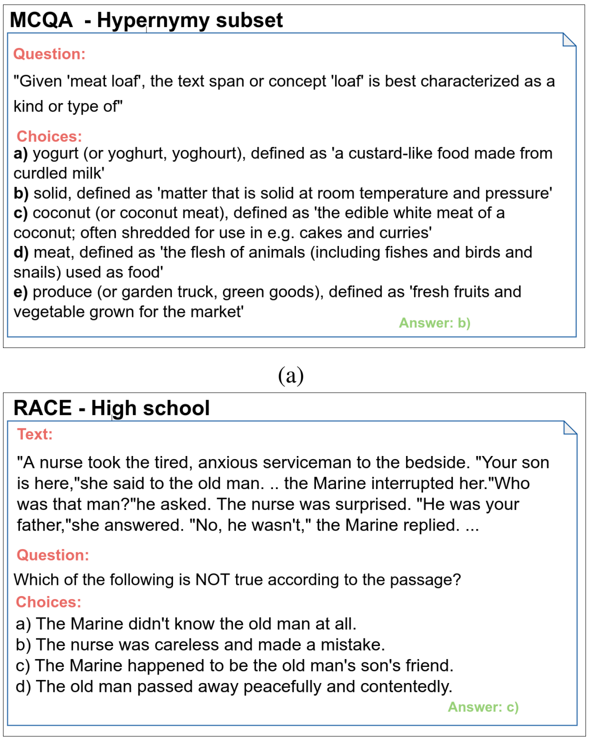
A Computing Infrastructure and Budget

All experiments were performed in a server Dell R740 bi pro Intel Xeon 2630 using Nvidia RTX6000 graphic card. A single training and test took around 100 and 120 minutes under this infrastructure. In summary, to compute the results of RQ1 and RQ2 we used approximately 120 GPU hours.

To compute the results of RQ3, including the inoculation technique, we used approximately 1,730 GPU hours.

B Dataset samples for Downstream Tasks

In Figure 6, we show entry samples for the MCQA (6a) and RACE (6b) tasks.



MCQA - Hypernymy subset

Question:

"Given 'meat loaf', the text span or concept 'loaf' is best characterized as a kind or type of"

Choices:

a) yogurt (or yoghurt, yoghourt), defined as 'a custard-like food made from curdled milk'
b) solid, defined as 'matter that is solid at room temperature and pressure'
c) coconut (or coconut meat), defined as 'the edible white meat of a coconut; often shredded for use in e.g. cakes and curries'
d) meat, defined as 'the flesh of animals (including fishes and birds and snails) used as food'
e) produce (or garden truck, green goods), defined as 'fresh fruits and vegetable grown for the market'

Answer: b)

(a)

RACE - High school

Text:

"A nurse took the tired, anxious serviceman to the bedside. "Your son is here,"she said to the old man. .. the Marine interrupted her."Who was that man?"he asked. The nurse was surprised. "He was your father,"she answered. "No, he wasn't," the Marine replied. ...

Question:

Which of the following is NOT true according to the passage?

Choices:

a) The Marine didn't know the old man at all.
b) The nurse was careless and made a mistake.
c) The Marine happened to be the old man's son's friend.
d) The old man passed away peacefully and contentedly.

Answer: c)

(b)

Figure 6: MCQA (a) and RACE (b) examples.

C Learning Curves from Inoculation Technique

Figure 7 shows the learning curves when applying the inoculation technique for the MCQA and RC tasks. We selected 5,000 as the number of samples with the best performance and smaller training size. Similar analysis was done for the RACE and S-RT datasets with equal conclusion w.r.t. the number of samples.

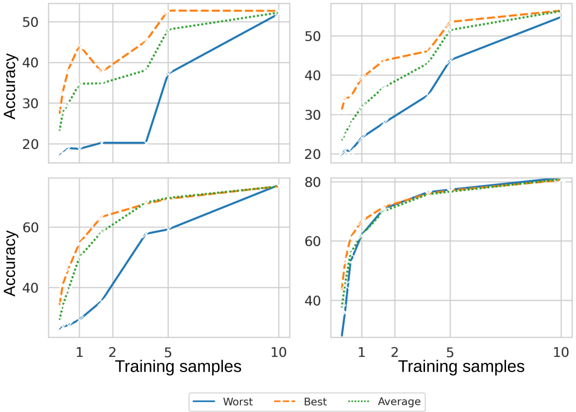


Figure 7: Learning curves for MCQA (upper) and RC (lower) tasks. For MCQA we show the Hypernymy (left) and Synonymy (right) dataset. For RC we show the Middle School (left) and High School (right) datasets. For all curves, the X axis represents the number of training samples (in thousands), and the Y axis, the accuracy score. Average values are reported with 5 runs for MCQA and 3 for RC.