

Disentangling Reasoning Capabilities from Language Models with Compositional Reasoning Transformers

Wanjun Zhong^{1*}, Tingting Ma^{2*}, Jiahai Wang¹, Jian Yin¹,
Tiejun Zhao², Chin-Yew Lin³ and Nan Duan³

¹ The School of Computer Science and Engineering, Sun Yat-sen University
Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, P.R.China

² Harbin Institute of Technology ³ Microsoft

zhongwj25@mail2.sysu.edu.cn, hittingtingma@gmail.com
{wangjiah, issjyin}@mail.sysu.edu.cn, tjzhao@hit.edu.cn
{cyl, nanduan}@microsoft.com;

Abstract

This paper presents ReasonFormer, a unified reasoning framework for mirroring the modular and compositional reasoning process of humans in complex decision-making. Inspired by dual-process theory in cognitive science, the representation module (automatic thinking) and reasoning modules (controlled thinking) are decoupled to capture different levels of cognition. Upon the top of the representation module, the pre-trained reasoning modules are modular and professional in specific and fundamental reasoning skills (e.g., logic, simple QA, etc). To mimic the controlled compositional thinking process, different reasoning modules are dynamically activated and composed in both parallel and cascaded manners to control what reasoning skills are activated and how deep the reasoning process will be reached to solve the current problems. The unified reasoning framework solves multiple tasks with a single model, and is trained and inferred in an end-to-end manner. Evaluated on 11 datasets requiring different reasoning skills and complexity, ReasonFormer demonstrates substantial performance boosts, revealing the compositional reasoning ability. Few-shot experiments exhibit better generalization ability by learning to compose pre-trained skills for new tasks with limited data, and decoupling the representation module and the reasoning modules. Further analysis shows the modularity of reasoning modules as different tasks activate distinct reasoning skills at different reasoning depths.¹

1 Introduction

Prevailing language models (LMs) (Devlin et al., 2018; Brown et al., 2020) demonstrate impressive performance in natural language processing tasks,

* Equal contributions during internship at Microsoft.

¹Code and model are available at <https://github.com/microsoft/KC/tree/main/papers/ReasonFormer>.

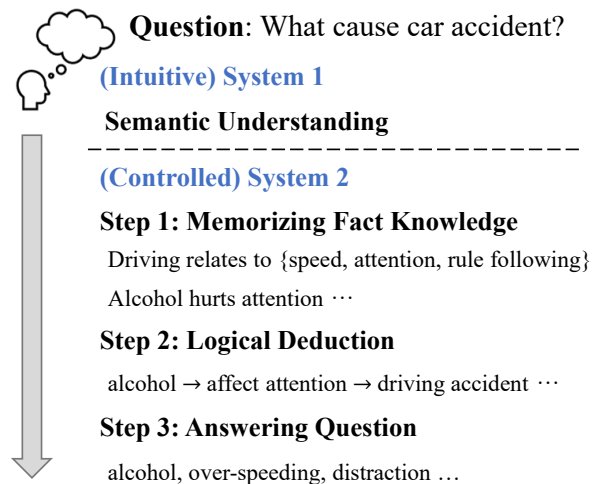


Figure 1: Compositional reasoning process of humans in complex decision-making. Humans solve the problems by cascaded executions of fundamental skills.

and have ushered in a new trend in AI research. Despite the emerging fervor, the homogeneous LMs relying on a single call of the model are less modular and are hard to explicitly model the complex reasoning process (Helwe et al., 2021) like humans.

In the dual-process theory (Daniel, 2017) in cognitive psychology, there are two cognitive systems interacted to form a whole reasoning process. System 1 (automatic thinking) generates intuitive patterns of ideas, and System 2 (controlled thinking) constructs reasoning in an orderly logical series of compositional reasoning processes. Besides, in the process of System 2, different functional brain areas could be modular and interact with each other. System 2 can decide how to compose different reasoning skills and when to stop thinking. As the example shown in Fig. 1, when finding the cause of a car accident, humans intuitively comprehend the question (System 1), and then conduct compositional reasoning (System 2: recalling fact → logical deduction → answering question).

We would like to incorporate this mechanism

into AI models in decision-making, and make the following assumptions: (1) the representation module (System 1) and reasoning module (System 2) can be decoupled and (2) the “complicated” reasoning process can be disentangled into multi-step executions of compositional “fundamental” reasoning modules, whose compositionality can be learnt with limited data. Also, the “fundamental” nature of basic reasoning skills allows them to have rich training instances for reliable skill pre-training.

Under these motivations, this paper proposes the modular and compositional reasoning framework - ReasonFormer, to mirror human’s compositional reasoning process, with the following characteristics: (1) the representation module and reasoning modules are decoupled; (2) reasoning modules are modular and professional in fundamental reasoning skills; (3) reasoning modules are compositional in parallel and cascaded manner, to dynamically decide the activated reasoning skills and the reasoning complexity; (4) the general-purpose reasoning framework is end-to-end and unified in solving multiple tasks with one model.

Specifically, the representation module learns contextual representations of problems. Upon the top of the it, there are cascaded reasoning modules to perform compositional multi-step reasoning. The reasoning modules are pre-trained to expert in specific reasoning skills (e.g., logic, QA, fact, etc.). These pre-trained reasoning skills are considered relatively fundamental and have rich resources. Two additional blocks complete the whole framework: the reasoning router and the reasoning adapter. The reasoning router decides which reasoning skills are activated in each reasoning step, and when to stop the reasoning process. The adapter adapts the reused reasoning modules to different steps of the reasoning process.

We comprehensively evaluate the framework on 11 datasets emphasizing different reasoning skills and complexity, and highlight the following findings: (1) Substantial performance boosts demonstrate models’ harvest of compositional reasoning ability, and both the reasoning-centric pre-training and reasoning adapter bring compounding performance gains. (2) Results of few-shot experiments show that specialized modules enables better generalization by learning to compose pre-trained skills for low-resource tasks, and decoupling of representation module and reasoning modules. (3) Further analysis reveals the distinct reasoning skills

required for different tasks at different reasoning depths, shoring up the modularity of reasoning modules.

2 Reasoning Skills Formulation

The compositional reasoning process of LMs’ relies on the pre-training of several fundamental reasoning skills and their compositionality. Hence, the selection of skills is critical.

Selection Principles. There are two major principles in selecting skills: (1) **Fundamental:** Complex problems can be decomposed and solved by simpler basic skills. So the basic skills should be more fundamental, well-defined, and can be covered in the required skill set of as many tasks as possible; (2) **Resourceful:** Reliable skill pre-training requires large-scale pre-training data. However, in the real-world scenario, the annotated data is expensive to obtain for most reasoning tasks. So it is expected that there are already rich resource or data can be collected via self(semi)-supervised manner.

Basic Skills Selection. Humans always solve complex problem with fundamental skills, like understanding key information (e.g., entity and its type) of events, recalling related facts, understanding causal relations between events, and extracting answers for the question. This motivates us to select the following basic skills: the **logic ability** to logically deduce the cause or consequence of events; **simple question answering (QA)** to understand the context and answer simple questions; **named entity recognition (NER)** to identify important entities in the context; **natural language inference (NLI)** to identify semantic relevance of two sentences and **factual knowledge** to memorize commonsense knowledge and understand daily events. There is an additional **general** skill to learn the commonly shared knowledge across selected skills. We keep this setting in our paper as they are relatively well defined and resourceful ².

We adopt self-supervised methods to construct pre-training corpus for {*logic ability, factual knowledge, NER*}, semi-supervised method to construct pre-training corpus for *simple QA*, and large-scale supervised data for *NLI*. Further details are given in § 4.2 and examples are given in Appendix A.

²It is worth noting that this selection is tentative. There are plausible ways for selecting basic skills or knowledge domains, which also inspire future directions.

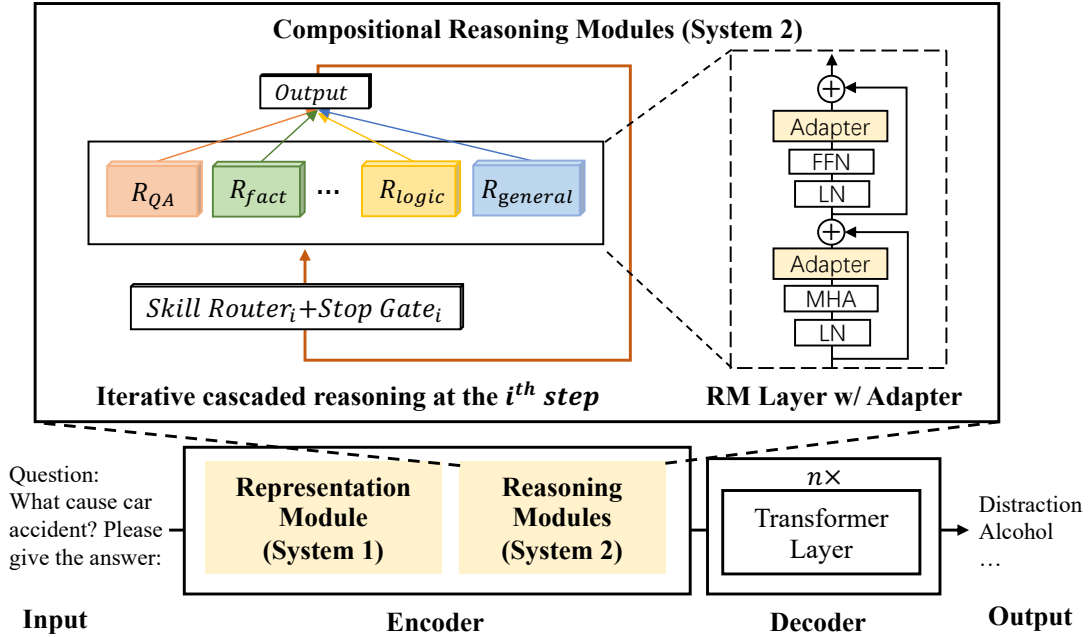


Figure 2: ReasonFormer framework. The representation module (§ 3.1) and reasoning modules (RMs) (§ 3.2) are decoupled to form the compositional reasoning process. The RMs are pre-trained with different reasoning skills R_{skill} (§ 2). The reasoning adapter (§ 3.2.1) adapts the shared RMs to different reasoning steps. Router decides activated skills. Stop gate decides when to stop reasoning (§ 3.2.2). Red lines indicate cascaded reasoning process.

3 ReasonFormer Framework

As shown in Fig. 2, the general-purpose reasoning framework is built based on encoder-decoder architecture to process multiple tasks (i.e., all pre-training tasks and downstream tasks) with a unified model, where all tasks are tackled as unified text-to-text generation tasks. We first reformat all the tasks into the same format using hard prompts (Sanh et al., 2021). For example, the question-answering task input can be prompted with the template: *The question is {Question}. Please give the answer:*, and the expected output is the answer text.

Given the prompted task inputs, the modular and compositional framework consists of two components in its encoder: the representation module (System 1) and the reasoning modules (System 2).

The **representation module** (§ 3.1) captures the intuitive understanding of problems by calculating initial contextual representations. Upon the top of the representation module, there are several pre-trained **reasoning modules** (§ 3.2) with different reasoning skills, waiting for interaction to form a compositional reasoning process. For reasoning process organization, there are **reasoning routers** (§ 3.2.2) to decide the (parallel) activated skills and when to stop the (cascaded) reasoning process.

3.1 Representation Module

Similar to the perceptive function of System 1, the representation module targets basic contextual understanding, and builds the foundation of the following-up reasoning process. As LMs exhibit impressive ability on contextual understanding, we build the representation module with cascaded Transformer layers. Given the tokenized input X with length m , the initial representations learnt from representation module are denoted as:

$$\mathbf{H}^0 = \{h_{[CLS]}^0, h_1^0, h_2^0, \dots, h_m^0\} \quad (1)$$

where $[CLS]$ is a special token.

3.2 Reasoning Modules

To simulate the cognitive process (System 2) formed by controlled interaction between various functional areas in human brains, the reasoning modules are modular and compositional. Reasoning modules (**RMs**) learn different reasoning skills specified during pre-training, and are automatically composed during downstream adaptation (§ 3.3) with reasoning router (§ 3.2.2). Compositionality is not only at the parallel level (different skills), but also at the cascaded level (multi-step reasoning). Since different reasoning steps intuitively model different levels of information, there are additional

reasoning adapters to adapt the reused modules to different reasoning steps.

3.2.1 Reasoning Modules Architecture

Each reasoning module is implemented by several Transformer layers. As shown in Fig.2(b), the shared reasoning modules with the same skill at different reasoning depths have shared parameters (excluding the reasoning adapter). For example, *Fact* modules at steps $\{0, 1, \dots, n\}$ share major parameters. The output from the last reasoning step will be recursively taken as the inputs of the reused reasoning modules with step-specific adapters.

Reasoning Adapter. To adapt the reused reasoning module to different depths of the reasoning process, we add step-specific reasoning adapters to the reasoning modules. Inspired by Houlsby et al. (2019) on domain adaptation, as shown in Fig. 2, we add two reasoning adapters following the multi-head attention layer and FFN layer in the Transformer layer of reasoning modules. Besides, the reasoning adapters for different skills and different reasoning depths are non-shared.

3.2.2 Reasoning Router

To compose the reasoning process, the reasoning router is critical in deciding which skills are activated per step, and how many reasoning steps are required for problem-solving. As the example in Fig. 1, problem-solving needs to recall facts, and make logical deductions, then answer questions. Therefore, the activated skills and reasoning depths may varied for every instance.

At the parallel level of each step, the **skill router** calculates activating scores for reasoning modules. After each reasoning step, the **stop gate** decides whether executed reasoning steps are sufficient in problem-solving through a stop gating mechanism.

Unlike Mixture-of-Experts (MoE) (Shazeer et al., 2017) that uses token-wise routing, we adopt an instance-level routing strategy, which can capture more comprehensive semantics of problems.

Skill Router. Since the i^{th} reasoning step has n reasoning modules: $\{R_1, \dots, R_n\}$ and a skill router S^i , the output \mathbf{H}^i of the i^{th} reasoning step can be calculated by router-weighted averaged outputs from the k activated reasoning modules:

$$\mathbf{H}^i = \sum_{j=1}^k S^i(\tilde{\mathbf{H}}^{i-1})_j R_j(\tilde{\mathbf{H}}^{i-1}) \quad (2)$$

where $S^i(\tilde{\mathbf{H}}^{i-1})_j$ (scalar weight) and $R_j(\tilde{\mathbf{H}}^{i-1})$ (updated hidden vectors) are the outputs from the router and the j^{th} reasoning module, respectively.

Since deciding the skills is a non-trivial task, we adopt a relatively complex router for deeper understanding. We use one Transformer layer T to project the original output for routing weight calculation. Then, we use an FFN layer followed by a Softmax function for weighted score calculation:

$$S^i(\tilde{\mathbf{H}}^{i-1}) = \text{Softmax}(\text{FFN}(T(\tilde{\mathbf{H}}^{i-1}))) \quad (3)$$

Afterwards, we **sparsely activate** (Shazeer et al., 2017) k reasoning modules with top- k skill routing scores at each reasoning step. The router training objectives are detailed in § 3.3.

Stop Gate. After each reasoning step, the stop gate decides whether the current reasoning depth is sufficient to solve the problem. Taking \mathbf{H}^i as the input, the stop gate uses a residual gating mechanism G_{stop}^i to control the information flow from executed reasoning steps and calculate the final output $\tilde{\mathbf{H}}^i$ for the i^{th} reasoning step by:

$$\tilde{\mathbf{H}}^i = \mathbf{H}^{i-1} + G_{stop}^i(\mathbf{H}^i) \quad (4)$$

An FFN layer is used as the stop gate G_{stop}^i . When the reasoning process is sufficient, the following-up process will be softly stopped by G_{stop}^i .

3.3 Pre-training and Adaptation

The unified model enables multi-task learning for both pre-training and downstream tasks. The major difference between pre-training and adaptation is that only in the pre-training stage we have the supervision for the activated skills.

Pre-training. Before reasoning pre-training, the model weights of ReasonFormer are initialized with pre-trained weights from T5 (Raffel et al., 2020). The details of model initialization and pre-training corpus collection are introduced in § 4.3.1 and § 4.2, respectively. Since model acknowledges which skill it is learning, we add **skill routing loss** L_r in addition to the teacher-forcing loss, to guide the routers in activating skills. For example, if the current instance focuses on logic ability, it should activate $\{logic\ ability, general\}$ skills. L_r can be set as the cross-entropy loss for the multi-skill classification, where the activated skill has label 1 and 0 otherwise. During pre-training, all the reasoning steps activate the same skill for one instance.

Adaptation. During downstream adaptation, we have no prior knowledge about the required skills for different tasks, so we expect the model can automatically learn which skills are essential for each specific task. Therefore, we adopt standard teacher-forcing loss for generative training.

4 Experiment Setup

4.1 Datasets

To verify the effectiveness of ReasonFormer, we extensively conduct experiments on **11** datasets emphasizing different reasoning types and complexity. Specifically, ReClor (Yu et al., 2020) emphasizes on logical reasoning. Commonsense QA (CSQA) (Talmor et al., 2018), ARC (Clark et al., 2018), PIQA (Bisk et al., 2020) and HellaSwag (Zellers et al., 2019) stress commonsense knowledge. Abductive NLI (aNLI) (Bhagavatula et al., 2019) is a natural language inference dataset. HotpotQA (Yang et al., 2018a) and WikiHop (Welbl et al., 2018) focus on multi-hop question answering. Mutual (Cui et al., 2020), DREAM (Sun et al., 2019) focus on reasoning over dialogue. RACE (Lai et al., 2017) is a general QA dataset. These datasets are related to the fundamental reasoning skills (§ 2) and fit nicely for analyzing the compositional reasoning process modeled by ReasonFormer.

During **Evaluation**, the Hotpot QA adopts *Exact Match* (*EM*) as the metric, while the rest tasks use *accuracy* as the metric. The answer for multi-choice QA and classification tasks are selected by the highest log-likelihood scores of options.

4.2 Pre-training Corpus

To reduce the manual efforts in data collection, we mainly select self(semi)-supervised pre-training corpus construction methods.

To improve LMs’ **logic** ability, we adopt the self-supervised logical pre-training corpus built by LogiGAN (Pi et al., 2022), which uses logical indicators to identify logical phenomena in a general text corpus. For **QA-centric** pre-training, we adopt the semi-supervised pre-training corpus construction method from ProQA (Lewis et al., 2021; Zhong et al., 2022a), which adopts a generation-filtering pipeline to build QA-centric corpus. To help the model in **identifying entities** from text, we use the self-supervised NER corpus (Chen et al., 2022) built from Wikidata and Wikipedia anchor link. To learn **factual knowledge**, we use Wikidata as a commonsense knowledge base to construct

self-supervised pre-training corpus. Specifically, we sample 1 million fact triples from Wikidata and construct the KG completion task (Moiseev et al., 2022) by recovering the masked tailed entities with the head entities and relations given as inputs.

Furthermore, since **natural language inference** task already have rich supervised data, we directly use MNLI (Williams et al., 2018) and SNLI (Bowman et al., 2015) datasets as the pre-training corpus.

Finally, 1 million instances are collected for each reasoning skill, and there are 5 millions pre-training instances in total for 5 reasoning skills. The examples and prompts for constructing inputs/outputs of the pre-training corpus are given in Appendix A.

4.3 Models

4.3.1 Model Initialization

We adopt encoder-decoder framework. In the encoder, the representation module has 9 Transformer layers, each shared reasoning module has 3 Transformer layers and the maximum reasoning depths is 3. We initialize the major model parameters from pre-trained T5_{base} (Raffel et al., 2020). Thus, the representation module is initialized by the 1th → 9th layers of T5 encoder, and the reasoning module is initialized by 9th → 12th layers of T5 encoder. The decoder is the same with T5.

4.3.2 Compared Methods

The major focuses of the experiment are to explore the effectiveness of ReasonFormer, and verify our hypotheses that complex problems can be disentangled and solved by compositional reasoning modules, and the decoupling of representation module and reasoning modules. We compare ReasonFormer with two series of methods.

T5 series. (1) **Vanilla T5** is the released T5 model (Raffel et al., 2020) (*google/t5-v1_1-base*) pre-trained with C4 corpus excluding other supervised data; (2) **Reasoning Pre-Trained T5 (RPT-T5)** is the T5 model continually pre-trained with our reasoning-centric pre-training corpus (§ 4.2).

MoRM series. Inspired by Mixture-of-Experts (MoE) methods (Shazeer et al., 2017; Lepikhin et al., 2020a), we develop Mixture-of-Reasoning Modules (MoRM) methods for comparison. Unlike MoE that builds parallel experts in the FFN layer of Transformer Layers, MoRM builds parallel reasoning modules (RMs) on the top of the representation module, and sparsely activate these RMs. Specifically, after initialized with T5, the last

Datasets	Reasoning	T5 Series		MoRM			ReasonFormer	
		T5	RPT-T5	w/o RPT (S)	RPT (S)	RPT (F)	S	F
Activated Parameters (M)		248	248	272	272	357	294	407
ReClor	Logic	35.2	36.8	35.4	36.8	35.4	39	39.4
ARC	Commonsense	31.4	32.7	25.4	34.1	31.1	35.1	34.1
CSQA	Commonsense	56.5	65.1	57.2	63	64.7	66.9	68.2
RACE	General	63.8	67.4	66.4	68.8	70.9	72.5	73.5
DREAM	General	59.3	64.5	56.6	61.8	67.7	70.5	70.5
aNLI	NLI	66.9	66.3	68.2	68.8	69.8	69.6	69.5
MuTual	Dialog	67.3	70.2	66.8	69.5	70.5	72.2	72.5
WikiHop	MultiHop	63.6	66.1	63.5	66.1	66.9	67.1	67.4
HotpotQA	MultiHop	61.1	63.3	63.1	63.3	63.8	65.2	65.5
Hellaswag	Commonsense	31.5	33.7	34.2	37.9	43	53.9	54.9
PIQA	Commonsense	61.4	63.3	64.6	65.4	67.6	67.5	67.9
Avg.		54.3	57.2	54.6	57.7	59.2	61.8	62.1

Table 1: Main results on 11 reasoning tasks. **RPT indicates reasoning-centric pre-training** introduced in § 2 & § 4.2. S indicates sparse activation (top 2) of RMs, while F denotes full activation of all RMs.

3 Transformer layers in the encoder are duplicated parallelly for N_s (numbers of skills) times, and the outputs of them are weighted average by the routing scores of the activated RMs. It increases the model size in the similar way with ReasonFormer, so it can verify whether the improvements are brought by the increased parameters. Besides, the major differences between ReasonFormer and MoRM are (1) MoRM involves no cascaded reasoning steps (depth=1); (2) Like MoE, RMs in MoRM are jointly trained for all instances without skill routing loss (§ 3.3), emphasizing no expertise of RMs. We also report the results of MoRM after **reasoning-centric pre-training (RPT-MoRM)**.

5 Experiment Analysis

5.1 Main Results

As presented in Table 1, ReasonFormer outperform T5 series and MoRM series across all tasks emphasizing the wide scope of different reasoning skills. Thus, we have the following findings:

ReasonFormer > MoRM & T5: ReasonFormer surpasses other methods (even with more activated parameters) by a large margin, giving evidence to our primary hypothesis that the expertise of reasoning modules and the cascaded compositional reasoning process essentially help the model in solving complex reasoning problems.

RPT-T5 > T5: The substantial performance boosts brought by RPT demonstrate that reasoning-targeted pre-training is essential in injecting various reasoning abilities into LMs.

Sparse v.s. Full: Sparse activation of RMs leads to slightly reduced but comparable performance

compared with full activation. It suggests that although activating more skills is beneficial, the most essential RM still plays the key role in problem-solving. The modularity of RMs can reduce the computation burden while keeping performance.

These positive findings manifest that ReasonFormer can model compositional reasoning and verify our primary hypothesis that the complex problem can be decomposed and well solved with pre-trained basic skills, and the representation module can be decoupled with the reasoning modules.

5.2 Ablation Study

We explore the truly functional components of ReasonFormer through ablation studies on 7 datasets. We evaluate the effectiveness of the following components: (1) reasoning pre-training; (2) cascaded reasoning mechanism; (3) expertise of reasoning skills (skill gating loss) and (4) reasoning adapter.

Reasoning Pre-training. We assume that the first factor contributing to the improvements is the multi-task reasoning-centric pre-training. Since vanilla LMs mainly focus on learning contextual semantics, and don't emphasize higher-level reasoning ability (Pi et al., 2022; Helwe et al., 2021), it is intuitive that reasoning-driven pre-training can enhance the model in solving complex problems. Results in Table 2 suggest that the ablation of pre-training from all models leads to a substantial performance drop, showing the importance of reasoning-centric pre-training in helping the reasoning modules to learn fundamental skills.

Cascaded Reasoning Mechanism. The second hypothesis is that the cascaded reasoning mechanism facilitates problem-solving with different

Modules	Models/Dataset	ARC	CSQA	DREAM	WikiHop	HotpotQA	Hellaswag	PIQA	Avg.
Cascaded (S)	ReasonFormer	35.1	66.9	70.5	67.1	65.2	53.9	67.5	60.9
	w/o RPT	24.1	56.9	59.5	64.2	64.4	34.7	65.6	52.8
	w/o adapter	33.1	64.2	68.4	66.8	64.8	47.1	67.4	58.8
Single (S)	ReasonFormer	34.8	63.7	65.9	66.6	63.9	39.7	66.3	57.3
	w/o RPT	25.4	57.3	56.7	63.6	63.1	34.3	64.6	52.1
	w/o modularity	34.1	63.1	61.8	66.1	63.4	37.9	65.4	55.9

Table 2: Ablation study on 7 datasets. RPT denotes reasoning pre-training (§ 2). Modularity denotes skill gating loss (§ 3.2.2), and adapter is reasoning adapter (§ 3.2.1). Cascaded and Single are different in the cascaded steps.

Models	Freezed Modules	#Tuned Para. (M)	ReClor Acc	CSQA Acc	RACE Acc	ARC Acc	MuTual Acc	WikiHop Acc	Avg.
T5	no	248	28.2	23.2	26.2	25.1	32.6	18.3	25.6
ReasonFormer	no	294	29.0	39.2	29.2	30.1	40.3	26.4	32.4
	RM	251	29.4	39.1	29.2	31.1	38.7	26.9	32.4
	rep.	230	29.2	38.9	29.2	30.1	38.0	26.5	32.0
	RM+rep.	188	29.2	37.8	28.4	28.8	31.6	25.4	30.2

Table 3: Few-shot experiments after freezing different modules. The representation module is abbreviated as rep.

complexity and composition orders. **Single** is an ablated version of ReasonFormer in which the reasoning modules are not cascaded horizontally (depth=1) and the adapter is also eliminated. Comparison between performances of Cascaded and Single version of ReasonFormer (Line 1 v.s. Line 4) demonstrates that the cascaded reasoning mechanism brings notable improvements and reveals the effectiveness of multi-step reasoning process.

Expertise of RMs. We assume that modularity and expertise of reasoning modules enables them to be flexibly composed. We ablate it from the Single version of ReasonFormer (Line 6) by pre-training all the RMs jointly without skill routing loss (§ 3.3) using the whole pre-training corpus. The apparent performance drop suggests that the expertise of RMs enables the model to discriminate the functionality of various skills and selectively compose them to form a whole reasoning process.

Reasoning Adapter. The reasoning adapters adapt the shared RMs to different reasoning steps. It is intuitively important as different levels of cognition focus on the information at different granularity. From Table 2, eliminating the reasoning adapter (Line 3) from ReasonFormer (Cascaded) harms the overall performance, testifying the distinct mechanisms at different levels of reasoning and the importance of reasoning-centric adaptation.

5.3 Low-resource Experiments

It is interesting to know whether the fundamental skills of RMs can be easily composed to solve new tasks with limited training data, and whether the

representation module and RMs can be decoupled during adaptation. If the answer is true, then the model’s generalization ability will be greatly enhanced with easy composition of pre-trained skills.

Under these motivations, we conduct few-shot experiments. We first examine the generalization of ReasonFormer and examine the decoupling of modules in ReasonFormer by freezing different modules during learning. We **freeze the RMs** (Line 3) to testify whether the skills can be directly reused without further fine-tuning. Then we **freeze the representation module** (Line 4) to verify the decoupling of representation and RMs. From Table 3, we highlight the following findings.

(1) ReasonFormer outperforms T5, showing that the generalization ability of ReasonFormer is enhanced by reasoning pre-training and explicit modeling of the compositional reasoning process.

(2) Freezing RMs (Line 3) achieves comparable and even slightly better performance than its fully tuned version, demonstrating that the learned skills can be easily composed with limited training data without further tuning RMs.

(3) Freezing the representation module (Line 4) also leads to comparable performance, proving that the representation module and RMs can be decoupled in adaptation. It suggests that it is feasible to reduce computation burden during few-shot adaptation by freezing the well-trained representation module and only tuning the RMs for tasks, which is more efficient when the representation module (e.g., gigantic LM) is extremely large for tuning.

(4) Freezing both modules (Line 5) hurts performance, showing that model adaptation to data

Commonsense QA	aNLI	Hotpot QA
<p>Question: Where would you find magazines along side other works?</p> <p>Options: A. Doctor B. Bookstore C. Market D. Train station E. Mortuary</p> <p>Activated Skills: General (0.99) Fact (0.97) QA (0.35) General (0.65)</p> <p style="text-align: center;">Reasoning Steps →</p>	<p>Premise: Julie had a coworker named Barry who loved to make trouble for others. Julie was embarrassed.</p> <p>Hypothesis Choices: A. Barry didn't tell anyone that Julie farted B. Barry laughed at Julie's unzipped pants</p> <p>Activated Skills: General (0.99) NER (0.3) NLI (0.3) General (0.7) General (0.7)</p> <p style="text-align: center;">Reasoning Steps →</p>	<p>Question: What is the name of the fight song of the university whose main campus is in Lawrence</p> <p>Passages: Kansas Song is a fight song of the University of Kansa, which is a public research university in the U.S.. The main campus in Lawrence ...</p> <p>Activated Skills: General (0.97) QA (0.7) QA (0.4) General (0.3) General (0.6)</p> <p style="text-align: center;">Reasoning Steps →</p>

Figure 3: Case study for activated skills analysis on 3 datasets. The answer is marked with orange.

distribution of specific tasks is still essential.

5.4 Reasoning Skills Analysis

Qualitative analyses are conducted to explore how the pre-trained skills are composed to solve different reasoning tasks, and how the skills changed at different reasoning depths. Therefore, we calculate the skill routing weights at every reasoning step (up to 3) for three tasks (i.e., {*Commonsense QA*, *aNLI*, *Hotpot QA*}). The case study provides examples and corresponding (top 2) activated skills at each step. As shown in Fig. 3, the activated skills are varied for different tasks, and are dynamically composed to form a series of reasoning steps. For commonsense reasoning, it emphasizes {*fact*, *QA*}. For NLI task, it emphasize {*NER*, *NLI*}. For multi-hop QA task, it executes the QA module for multiple steps. The statistical analysis of averaged routing scores on the whole evaluation set also demonstrate the same trend. These observations show improved interpretability of decision-making and give evidence to the hypothesis that the compositional cognitive process of humans can be transferred to AI model.

6 Related Works

Multi-step Reasoning. Multi-step reasoning is a characteristic of human thinking. Multi-hop reasoning (Yang et al., 2018b; Yu et al., 2021) asks the system to logically switch attention to different contexts (Zhong et al., 2022b) or make a multi-step deduction for a new conclusion (Dalvi et al., 2019; Zhong et al., 2021). Recently, *chain-of-thought* prompting (Wei et al., 2022) provides the model with manual prompts about the intermediate reasoning steps. Creswell and Shanahan (2022) use LMs to iteratively select evidence and generate inferences. However, they always require discrete manual-written reasoning traces. Dohan et al. (2022) is a position paper raising interest in mod-

eling these cascaded inference processes of LMs with a probabilistic program.

LM Modularity. Since human brains have various functional areas, it is inspiring to explore the modularity of LMs. Mixture-of-Experts (MoE) (Shazeer et al., 2017; Lepikhin et al., 2020b) use experts in FFN layers for sparse learning. However, their major motivation is to increase the model capacity while keeping efficiency, without emphasis on the speciality of expert. Recent works begin to explore domain-specific experts (Gururangan et al., 2021) and modality-specific experts (Wang et al., 2021). SkillNet proposes skill-specific experts (Zhang et al., 2022). However, the activated skills need to be manually specified, and do not explicitly model the cascaded reasoning process and disentangling of perception and cognition.

Considering these directions in the whole picture, this paper targets to explore the modeling of modular and the compositional multi-step reasoning process of AI models in an end-to-end manner.

7 Conclusion

This paper stimulates the compositional reasoning process of humans in decision-making, and makes the following hypotheses: (1) the intuitive perception system (System 1) and cognitive reasoning system (System 2) can be decoupled and (2) the complex decision-making can be disentangled into multi-step execution of fundamental reasoning skills. Correspondingly, we propose ReasonFormer, a compositional general-purpose reasoning framework. ReasonFormer decouples the representation module and reasoning modules, which are pre-trained to expert in fundamental reasoning skills. The reasoning modules are dynamically composed in parallel and cascaded manner to form a whole reasoning process. ReasonFormer is end-

to-end and unified in solving multiple tasks with one model. Extensive experiments on 11 tasks reveal the compositional reasoning ability of ReasonFormer and disentangling of representation and reasoning modules.

Limitations

As mentioned in Sec. 2, the current selection of fundamental reasoning skills for language models is limited by the availability of well-defined tasks and clear definitions of those tasks, as well as the availability of sufficient training data. As a result, some skills may overlap or may not be fundamental enough. For example, simple QA skill may overlap with NER skill to some extent. In the future, it would be worthwhile to explore self-supervised training tasks that can inject more fundamental abilities into language models. Additionally, the selection and combination of fundamental reasoning skills can be further explored. For example, the inclusion of numerical reasoning ability to solve mathematical problems. Additionally, methods for skill-centric pre-training corpus construction can also be explored to improve the effectiveness of these skills.

Ethics Statement

The present study was conducted in accordance with ethical principles. This study involved the analysis using publicly available data and knowledge sources (e.g., Wikipedia). Thus, this work did not involve any human participants and potential risks regarding credentials or privacy. Therefore, no ethical clearance was required and there were no potential risks associated with the conduct of this research.

Acknowledgments

Jian Yin is the corresponding author. Wanjun Zhong and Jian Yin are supported by the National Natural Science Foundation of China (U1911203, U2001211, U22B2060), Guangdong Basic and Applied Basic Research Foundation (2019B1515130001), Key-Area Research and Development Program of Guangdong Province (2020B0101100001)

References

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Han-

nah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. *arXiv preprint arXiv:1908.05739*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jiawei Chen, Qing Liu, Hongyu Lin, Xianpei Han, and Le Sun. 2022. Few-shot named entity recognition with self-describing networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5711–5722, Dublin, Ireland. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Antonia Creswell and Murray Shanahan. 2022. Faithful reasoning using large language models. *arXiv preprint arXiv:2208.14271*.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. Mutual: A dataset for multi-turn dialogue reasoning. *CoRR*, abs/2004.04494.

Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, and Peter Clark. 2019. Everything happens for a reason: Discovering the purpose of actions in procedural text. *ArXiv*, abs/1909.04745.

Kahneman Daniel. 2017. Thinking, fast and slow.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

David Dohan, Winnie Xu, Aitor Lewkowycz, Jacob Austin, David Bieber, Raphael Gontijo Lopes, Yuhuai Wu, Henryk Michalewski, Rif A Saurous, Jascha Sohl-dickstein, et al. 2022. Language model cascades. *arXiv preprint arXiv:2207.10342*.

- Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A Smith, and Luke Zettlemoyer. 2021. Demix layers: Disentangling domains for modular language modeling. *arXiv preprint arXiv:2108.05036*.
- Chadi Helwe, Chloé Clavel, and Fabian M Suchanek. 2021. Reasoning with transformer-based models: Deep learning, but shallow reasoning. In *3rd Conference on Automated Knowledge Base Construction*.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.
- Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020a. Gshard: Scaling giant models with conditional computation and automatic sharding. *Learning*.
- Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020b. Gshard: Scaling giant models with conditional computation and automatic sharding. *Learning*.
- Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.
- Fedor Moiseev, Zhe Dong, Enrique Alfonseca, and Martin Jaggi. 2022. SKILL: Structured knowledge infusion for large language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1581–1588, Seattle, United States. Association for Computational Linguistics.
- Xinyu Pi, Wanjun Zhong, Yan Gao, Nan Duan, and Jian-Guang Lou. 2022. Logigan: Learning logical reasoning via adversarial pre-training. *arXiv preprint arXiv:2205.08794*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.
- Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge dataset and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.
- Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. 2021. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *arXiv preprint arXiv:2111.02358*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.
- Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018a. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *CoRR*, abs/1809.09600.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018b. HotpotQA: A

dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jianxing Yu, Qinliang Su, Xiaojun Quan, and Jian Yin. 2021. Multi-hop reasoning question generation and its application. *IEEE Transactions on Knowledge and Data Engineering*.

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)*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Fan Zhang, Duyu Tang, Yong Dai, Cong Zhou, Shuangzhi Wu, and Shuming Shi. 2022. Skillnet-nlu: A sparsely activated model for general-purpose natural language understanding. *arXiv e-prints*, pages arXiv–2203.

Wanjun Zhong, Yifan Gao, Ning Ding, Yujia Qin, Zhiyuan Liu, Ming Zhou, Jiahai Wang, Jian Yin, and Nan Duan. 2022a. Proqa: Structural prompt-based pre-training for unified question answering. *arXiv preprint arXiv:2205.04040*.

Wanjun Zhong, Junjie Huang, Qian Liu, Ming Zhou, Jiahai Wang, Jian Yin, and Nan Duan. 2022b. Reasoning over hybrid chain for table-and-text open domain qa. *arXiv preprint arXiv:2201.05880*.

Wanjun Zhong, Siyuan Wang, Duyu Tang, Zenan Xu, Daya Guo, Jiahai Wang, Jian Yin, Ming Zhou, and Nan Duan. 2021. Ar-lsat: Investigating analytical reasoning of text. *arXiv preprint arXiv:2104.06598*.

A Example of Pre-training Tasks

For the basic question answering skill, QA-centric pre-training uses a generation-filtering pipeline to build semi-supervised large-scale corpus (Lewis et al., 2021; Zhong et al., 2022a): (1) use annotated QA data to train a passage-to-question-answer generator (2) taking the wikipedia passages as inputs, and generates corresponding pseudo questions and answers (3) filtering passage, question, answer pairs with a QA model.

For logic skill, we use the automatically constructed data from LogicGAN (Pi et al., 2022). It uses logical indicators (e.g., Therefore, as a result) to automatically identify logical inference phenomenon presented via natural language, and mask corresponding causes/results of events, and ask the pre-trained model to recover them to learn logical reasoning ability.

For the natural language inference, we use the public annotated corpus SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018). Given a sentence as premise, the model is expected to predict whether the premise sentence entails the hypothesis sentence.

For the named entity recognition skill, we use weakly-annotated data (Chen et al., 2022) obtained from Wikipedia and Wikidata. The mentions are the text with anchorlink and the types are obtained from Wikidata “instance of” or “subclass of” properties. We design three pretrain tasks similar to Chen et al. (2022): 1) given the sentence, identify all mentions in the sentence 2) given the sentence and interested types, output all mentions with these types in the sentence 3) given the sentence and mentions, predict all types of the mentions.

For the fact skill, we use fact triples from Wikidata, and design a task that predict the tail entity given the head entity and relation as Moiseev et al. (2022).

A summary of the examples for each tasks is presented in Fig 4.

B Implementation Details

Pretraining Details We use “google/t5-v1_1-base” from HuggingFace (Wolf et al., 2020) implementation as base model for all our experiments. We use a learning rate of 5e-5 and train all models with 5 epochs. The warmup ratio is set to 0.1. The total batch size is set to 72 for shared model and 64 for private model. The down projection hidden size of adapter is set to 256. We use 8 V100 GPUs for model training.

Downstream Adaptation Details For all the full data experiments, we use a learning rate of 1e-4 and training epoch of 10 with a batch size of 48 for our models. The model is validated at the end of each epoch. For all the few-shot experiments, we use a learning rate of 1e-5 and training epochs of 200 with a batch size of 8. The model is validated per 200 steps.

Skill	Corpus	Example	Prompt
QA	Wikipedia	Context: ...appeared to Saint Bernadette Soubirous in 1858. At the end... Question: To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France? Answer: Saint Bernadette Soubirous	Input: {context} [SEP] give me the answer of {question} Output: {answer}
Logic	BookCorpus	Context: All men are mortal, and Socrates is a man. Therefore, [MASK], MASK: Socrates is mortal	Input: {context} [SEP] give me the missing statement Output: {mask}
NLI	SNLI, MNLI	Premise: Conceptually cream skimming has two basic dimensions - product and geography. Hypothesis: Product and geography are what make cream skimming work. Label: Neutral	Input: {premise} [SEP] {hypothesis} give me the relation between the first and second sentences Output: {label}
NER	Wikipedia, Wikidata	Context: ...whose family had originally migrated from the state of Mysore. Type_str: city Mention_str: Mysore is city.	Input: {context} [SEP] give me the mentions with types {type_str} Output: {mention_str}
		Context: ... nationalist, communist and anarchist who was among the founding members of the Communist Party of India (Tashkent group). Mention_str: India, nationalist, Tashkent	Input: {context} [SEP] give me all mentions Output: {mention_str}
		Context: ...whose family had originally migrated from the state of Mysore. Mention_str: Mysore Type_str: Mysore is city.	Input: {context} [SEP] give me the types of mentions {mention_str} Output: {type_str}
Fact	Wikidata	Head: Knut Wijkmark Relation: child Tail: Nils Wijkmark	Input: {head} {relation} [SEP] give me the missing entity: Output: {tail}

Figure 4: Examples of pre-training tasks.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
Last section after the conclusion section.
- A2. Did you discuss any potential risks of your work?
I don't think of any risk. It is shown in the ethical statement section.
- A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

The applied datasets are mentioned in Section 4.

- B1. Did you cite the creators of artifacts you used?
Section 4.
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. These datasets are suitable for academic research purpose.
- 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. These datasets are suitable for academic research purpose.
- 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?
Not applicable. Left blank.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4.1
- 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.
We cover a wide scope of public evaluation datasets. The complete details can't be introduced in the main pages due to the page limits.

C Did you run computational experiments?

Section 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 and Section 4

- 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?

Section 4

- 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.)?

The evaluation metrics are introduced in Section 4.1

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.