Connecting-the-Facts: Integrating Associative Thinking into Large Language Models via Learning from Knowledge Graphs

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have exhibited impressive proficiency in various applications, which poses an increasing demand for their advanced reasoning capabilities. Associative thinking, *i.e.*, the mental process of connecting known knowledge with seemingly unrelated ideas, is crucial for humans yet remains underexplored for LLMs. In this paper, we systematically study associative thinking in LLMs, leveraging Knowledge Graphs (KGs). We propose **CtF** (*i.e.*, **C**onnecting **t**he **F**acts), which generates associative thinking data based on compositional rules mined from KGs. Then, we improve LLMs via fine-tuning on the generated dataset KGREASON, and introduce a trial-and-error mechanism to better simulate the human process of internal knowledge exploration. The results of extensive experiments demonstrate that our method effectively equips LLMs with the abstract reasoning capabilities, and further benefits LLMs in various reasoning tasks.

1 Introduction

Large Language Models (LLMs) have established new state-of-the-arts across a wide range of natural language processing (NLP) tasks (Brown et al., 2020; Bang et al., 2023). Increasingly, their impressiveness has expanded to many practical applications and complex problems, which demand advanced reasoning abilities of various kinds, such as arithmetic reasoning (Cobbe et al., 2021), commonsense reasoning (Talmor et al., 2018), and symbolic reasoning (Srivastava et al., 2022).

Associative thinking (Beaty and Kenett, 2023) allows humans to connect known knowledge from memory with new problems, which often seem unrelated literally. For instance, given that *Mary* works at *Company A* and that *Company A* is located in *City B*, humans can effortlessly infer that *Mary* might live in *City B*. However, as illustrated in Figure 1, current LLMs, despite having the necessary knowledge, still struggle with this type of reasoning and hallucinate. For humans, associative thinking depends on the ability to retrieve and link memories with general knowledge (Tulving, 1972) and operates learned patterns of behavior and knowledge (VanLehn, 1996; Anderson, 2013). Inspired by this, teaching LLMs to link separate facts stored in the internal knowledge base using patterns from commonsense is key to enhance their capability of associative thinking.

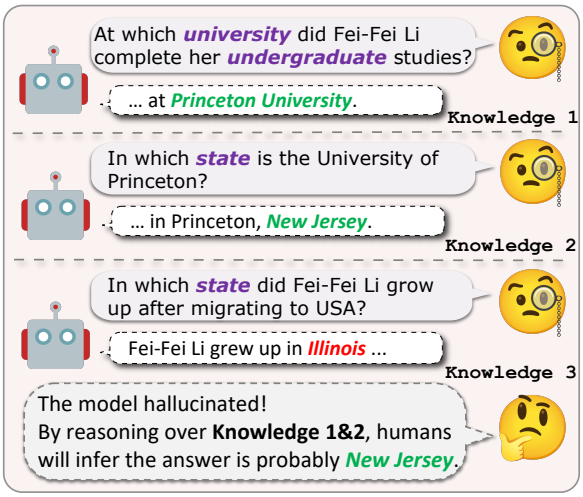

Figure 1: Current LLMs struggle with associative thinking, *i.e.*, combining acquired knowledge to infer new knowledge.

In this paper, we systematically explore LLM's associative thinking capabilities leveraging Knowledge Graphs (KGs). Specifically, we propose **CtF** (*i.e.*, **C**onnecting **t**he **F**acts), which improves LLMs' associative thinking via synthesized data derived from compositional rules in KGs. **CtF** includes methodologies for both dataset construction and model learning. The ***dataset construction*** is based on KGs. As illustrated in Figure 2, it includes three steps: *1) rule mining* , which mines compositional rules in KGs; *2) knowledge selection*,

which identifies interrelated triples (facts) matching those rules; and *3) sample generation*, which transforms the triples into natural language samples. For ***model learning***, we first fine-tune LLMs on our dataset via behavior cloning, which we observe often leads to rule overfitting and consequent hallucination. Hence, we further enhances **CtF** with a *trial-and-error* mechanism, which simulates humans' internal process of knowledge exploration for improved generalizability.

We conduct extensive experiments to validate the effectiveness of **CtF**, which covers both anonymized and regular settings. In the anonymized settings, we replace entity names to ensure analysis uninfluenced by data leakage. In the regular settings, we showcase the value of **CtF** for not only real-word associative reasoning, but also other reasoning benchmarks.

The contributions of this paper are mainly summarized as follows:

- We systematically study associative thinking in LLMs. We propose **CtF** to enhance LLMs' associative thinking based on compositional rules and facts from KGs, and fine-tune them with two learning strategies: behavior cloning and trial-and-error.
- We construct the KGREASON dataset to evaluate and enhance LLM's associative thinking, which covers various settings and reasoning steps. Our code and dataset will be released to facilitate future research in this direction.
- We conduct extensive experiments across various settings. Our results show that **CtF** effectively improves LLMs' associative thinking, with promising generalizability to novel facts, rules and upgraded challenges. Moreover, we showcase the broader value of **CtF** via its benefits on various reasoning tasks.

2 Related Works

LLM Reasoning LLMs have achieved significant success in many NLP tasks and their capabilities have been extended to complex reasoning tasks such as common-sense reasoning (Talmor et al., 2018), arithmetic reasoning (Cobbe et al., 2021), and symbolic reasoning (Srivastava et al., 2022). It has been observed that LLMs perform poorly on reasoning tasks when using standard prompts (Wei et al., 2022). To address this issue, Brown et al. proposed few-shot prompting (Brown et al., 2020), which provides the model with examples of question-answer pairs and has proven effective in reasoning tasks. To further enhance performance, (Wei et al., 2022) proposed Chain-of-Thought (CoT) prompting (Wei et al., 2022), which provides the model with input-output examples that include explicit reasoning steps. Different from CoT, Program of Thoughts (PoT) uses language models to express the reasoning process as a program, and then executes the generated programs to derive the answer (Chen et al., 2022). Tree-of-Thought (ToT) extends the concept of CoT by incorporating a hierarchical structure into the reasoning process (Yao et al., 2024). Although ToT is effective for decision-making and path selection, it requires access to external information such as context. In contrast, our work introduces the **CtF** framework to enhance the associative thinking abilities of LLMs by utilizing their internal knowledge base.

Associative Thinking Associative thinking is a human cognitive ability defined as "the forming of associative elements into new combinations which either meet specific requirements or are in some way useful"(Mednick, 1962). This type of thinking is rapid and often subconscious(Kahneman, 2011), and it is considered a crucial component of creativity (Beaty and Kenett, 2023). In AI systems, the thought of associative thinking is essential for tasks such as language processing and decision-making. In natural language processing, Collins and Quillian introduced semantic networks that model relationships between concepts, paralleling human associative memory (Collins and Quillian, 1969). Additionally, cognitive computing frameworks have been developed to replicate human-like associative thinking, thereby enhancing AI's ability to tackle complex tasks (Raghavan et al., 2016). In large language models (LLMs), associative thinking is leveraged to enhance creativity (Mehrotra et al., 2024). Overall, these studies underscore the importance of associative thinking across various domains.

3 Methodology

3.1 Preliminaries and Task Formulation

Knowledge Graphs KGs store collections of facts as triples, denoted as $G = \{(e, r, e') \mid e, e' \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ and $\mathcal{R}$ represent the sets of entities and relations, respectively.

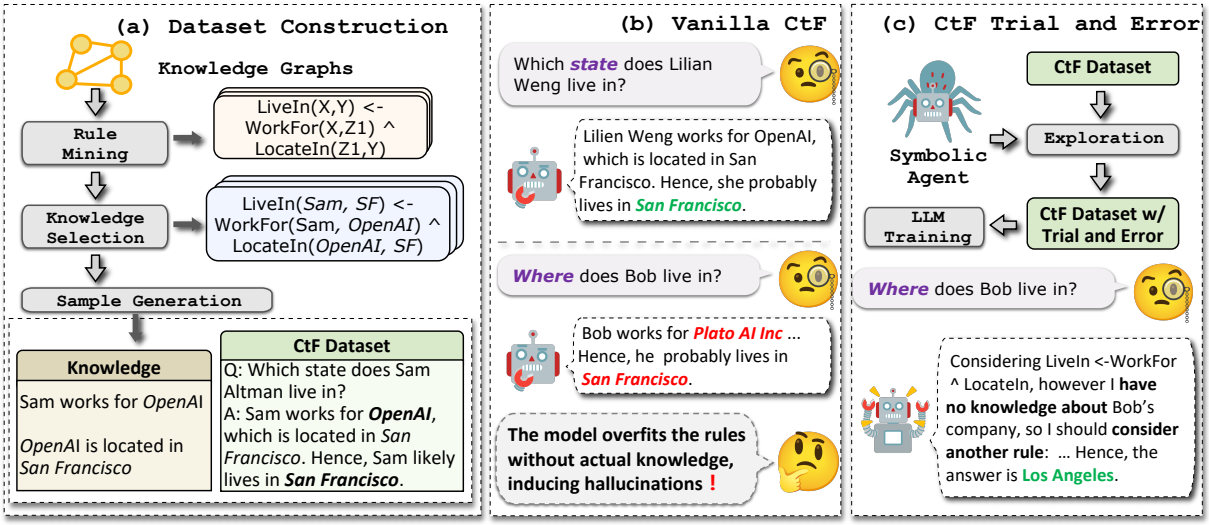

Figure 2: The framework of Connecting-the-Facts(**CtF**). **(a) Dataset Construction** includes three steps, *i.e.*, rule mining, knowledge selection and sample generation. This yields a knowledge dataset and **CtF** dataset. **(b) Vanilla CtF** trains LLMs in a behavior cloning manner, which may induce rule overfitting and hallucination. **(c) CtF (Trial and Error)** is henced proposed, which enables LLMs to simulate humans' internal process of knowledge exploration.

Atoms and Rules KGs contain compositional rules that can be extracted or modeled to infer new knowledge. These rules consist of multiple relational atoms, and an atom can be expressed as $r(X, Y)$, where r is a relation and X, Y are variables for entities. A *rule* in the KGs can be expressed by the following formula:

$$r_h(X, Y) \leftarrow r_1(X, Z_1) \wedge \ldots \wedge r_n(Z_{n-1}, Y) \quad (1)$$

Here, $r_h(X, Y)$ denotes the *head* atom and denotes the *rule head*, while $r_1(X, Z_1) \ldots \wedge r_n(Z_{n-1}, Y)$ denotes the *rule body*. For example, in the rule $\textit{LiveIn}(X, Y) \leftarrow \textit{WorkFor}(X, Z_1) \wedge \textit{LocateIn}(Z_1, Y) \wedge \textit{LiveIn}(X, Y)$ is the rule head and $\textit{WorkFor}(X, Z_1) \wedge \textit{LocateIn}(Z_1, Y)$ is the rule body.

Task Formulation based on a set of existing facts, by applying implicit rules. Associative reasoning aims to enable models to infer new, previously unknown facts from a set of known facts. In the question, given an atom $r_h(X, Y)$, where X is known and Y is unknown, we seek to determine Y. The task of associative reasoning is to infer the new fact Y based on a set of existing facts, by applying implicit rules.

3.2 Data Construction

Rule Mining In this step, we begin by mining two-hop rules and then combine them to create three-hop and four-hop rules.

Traditional rule mining approaches primarily rely on algorithms, which often result in low accuracy of the generated rules. To address this issue, we propose a new rule mining method that utilizes large models for rule selection, leading to higher-quality rules.

To derive rules for data construction from triples in the knowledge graph, we utilize a breadth-first approach to sample two-hop atoms combinations that connect the head entity to the tail entity. The algorithm we use is shown in Appendix A.1 These combinations serve as instances for 2-hop rules. For example, given an atom $r_3(e1, e3)$, we can sample the instance $r_1(e1, e3) \leftarrow r_2(e1, e2) \wedge r_3(e2, e3)$. The head and tail of this path correspond to the head entity e_1 and the tail entity e_3 of the atom respectively. The corresponding rule is $r_1(X, Y) \leftarrow r_2(X, Z) \wedge r_3(Z, Y)$. After sampling across the entire knowledge graph, we obtain a series of rule instances. First, we count the number of instances corresponding to each rule and set a threshold N_t. Rules with fewer than N_t instances are considered atypical and are removed from the list. For the rules obtained in the first round, we employ advanced models to evaluate their quality.

Then we combine two-hop rules to create longer rules. Given two rules, if the rule head of one rule is part of the rule body of another rule, we replace that part with the rule body of the first rule. For example, consider the following two rules:

1) $\textit{BornIn}(X, Y) \leftarrow \textit{HighSchool}(X, Z_1) \wedge \textit{LocateIn}(Z_1, Y)$;
2) $\textit{CitizenOf}(X, Y) \leftarrow \textit{BornIn}(X, Z_1) \wedge \textit{CityOf}(Z_1, Y)$.

The head of Rule1, $\textit{BornIn}(X, Y)$, is part of the body of Rule2, so we replace it with the body of Rule1 to form a 3-hop rule: $\textit{CitizenOf}(X, Y) \leftarrow \textit{HighSchool}(X, Z_1) \wedge \textit{LocateIn}(Z_1, Y) \wedge \textit{CityOf}(Z_1, Y)$.

Knowledge Selection In the process of associative reasoning, if LLMs do not have access to the facts that support the rule, the reasoning cannot be completed due to the lack of key information. On the other hand, if the fact we query is already embedded in the LLMs' internal parameters, the validity of the reasoning cannot be assured due to potential data leakage. Therefore, it is essential to select knowledge that is appropriate for **CtF** data construction.

We construct datasets in both anonymized settings and regular settings, and different knowledge selection processes are applied to the two settings. For each rule obtained, we first identify all its instances from the knowledge graph. To prevent the model from overfitting to a particular rule during training, which could lead to path dependency, we ensure a balanced quantity of each rule in the training data. We achieve this by sampling an equal number of instances for each rule. Next, for each instance, we gather the involved facts and use them for knowledge selection.

In the **anonymized setting**, we replace all entity names in the facts with random, non-existent strings, making all entities new knowledge to the LLMs. We then collect all the anonymized facts and use them to generate knowledge data for knowledge injection. In the **regular setting**, the entities in the facts represent real-world knowledge. For each instance, we gather the relevant facts and use them to perform knowledge probing on the LLM. If the LLM knows all the facts supporting the body of the instance but does not know the fact represented by the instance head, we retain this instance for generating samples in the subsequent step.

Sample Generation Finally, we apply advanced LLMs [1] to transform the knowledge into natural language sentences.

For knowledge dataset, we generate a paragraph of natural language description for each entity. For **CtF** dataset, we generate a sample for each instance obtained in the previous step . For the rule head $r_h(X, Y)$, we prompt advanced LLMs to generate a natural language question. Given the relationships between entities, if multiple Ys correspond to a single X, questioning Y will result in multiple answers, complicating evaluation. Therefore, we choose the unique entity between X and Y to generate questions. For the rule body, we combine all the facts to form a reasoning chain that describes the reasoning process from X to Y, which will be used as the answer to the question. The details of sample generation are shown in Appendix A.3

Our experiments include both anonymized and regular settings. In each setting, the **CtF** dataset is used to fine-tune LLMs in a supervised manner. Conversely, the knowledge dataset is employed exclusively during the continuous pretraining stage in anonymized settings to inject knowledge into LLMs.

3.3 Learning of Associative Thinking

Vanilla Training First, we directly train LLMs on KGREASON in a behavior cloning manner. (In the anynomized setting, the model should first be continues pretrained on the knowledge dataset to get the needed knowledge.) However, we observe a phenomenon called *rule overfitting*. In this situation, the trained models tend to rely on rules encountered during training, even in the absence of supporting facts.

Trial and Error Hence, we introduce a trial-and-error (T&E) mechanism to **CtF** learning, which simulates the human process of exploring over our internal knowledge.

When doing associated reasoning, humans initially select a plausible rule and start reasoning based on it when presented with a question. During this process, if we realize that we lack a crucial fact required by the rule, we switch to an alternative rule. Hence, we integrate the concept of trial and error with our method, incorporating exploration of the LLM's internal knowledge base into the reasoning process. This approach enables LLMs to discern when to apply a rule and when to backtrace it due to a lack of supporting facts, subsequently switching to a more appropriate rule.

We design a symbolic agent, which is an algorithm module to work in conjunction with LLMs to generate exploration path, employing a trial-and-error approach. For each sample, the symbolic agent first selects a possible rule as a candidate

[1] In this paper, we use GPT-3.5-turbo-1106 for sample generation.

path and then searches for supporting facts for the rule in the internal knowledge base of LLMs. If any part of the rule body lacks supporting facts, the process is recorded as an error, and the symbolic agent switches to another rule as the candidate path. This process repeats until a reasoning path with sufficient supporting facts is found, leading to the desired result. The entire exploration process is captured as a data sample, comprising at least one error and the correct reasoning path. The algorithm used in trial and error process is shown in Appendix A.4.

4 Experiments

4.1 Settings

Datasets We select Wikidata5m (Wang et al., 2021) as our data source, which is a million-scale knowledge graph dataset that is aligned with Wikidata, facilitating data processing and usage. We construct a dataset KGREASON, which includes a knowledge dataset and a **CtF** dataset. The construction method is detailed in Section 3.

Anonymized Settings and Regular Settings We conduct experiments in both the anonymized settings and the regular settings.

In the **anonymized settings**, we conduct experiments to study LLM's associative thinking, avoiding the influence of LLMs' inherent knowledge for this task. In these settings, all entity names are replaced with random, non-existent character names, ensuring that the model parameters contain no prior knowledge of these entities. Consequently, our training data comprises knowledge dataset used during the continuous pre-training stage, as well as **CtF** dataset used during the instruction fine-tuning stage. The knowledge data includes corpus information related to each entity, which injects the necessary prerequisite knowledge for reasoning into the model. Meanwhile, the **CtF** and **CtF** (T&E) data serve as the **CtF** dataset during supervised fine-tuning stage. The statistics of our training data is showed in Table 1.

In the **regular settings**, we aim at validating the effectiveness of **CtF** in real-world scenarios. The entities and relations in the regular settings reflect real-world knowledge. Consequently, we fine-tune LLMs with the **CtF** dataset without the knowledge injection step. To avoid significantly diminishing the model's general ability, we mix ten times the amount of general instruction data during the fine-tuning stage. Besides associative reasoning, we further evaluate the benefits of **CtF** learning for LLMs' general reasoning abilities in downstream tasks.

Evaluation Splits To evaluate the associative thinking ability of the model, we designed two test datasets: In-Domain (ID) and Out-of-Domain (OOD) tests.

1) In-Domain Tests The reasoning paths of the samples in ID setting also appear in the training samples. A higher score on the ID setting indicates that LLMs can learn associated reasoning by applying the learned rules.

2) Out-of-Domain Tests Unlike the ID setting, the reasoning paths of the samples in the OOD setting do not appear in the training data. A higher score on the OOD setting signifies that the associative thinking capability of LLMs effectively generalizes to previously unseen rules.

Additionally, both ID and OOD settings are divided into three subsets, each corresponding to a different rule length.

Models We conduct our experiments on Llama3-8B-instruct (AI@Meta, 2024) and Mistral-7b-instruct-v0.2 (Jiang et al., 2023).

During the fine-tuning stage, we employ full fine-tuning for model training. For each training dataset, the model is trained for 4 epochs. After each epoch, it is tested on the evaluation datasets, and the best performance is reported as the result of that training setting.

Methods We compare the following four methods in the anonymized settings.

1) Vanilla CoT: In this approach, we prompt the model to answer the question with step-by-step reasoning without any fine-tuning.

2) In-Context Learning CtF (ICL-CtF): In this approach, we provide the model with six examples of question and answer pairs from our **CtF** learning data.

3) CoT Learning: In this approach, we fine-tune the model using a combination of general CoT QA pairs.

4) CtF: In this approach, we finetune the model with our **CtF** data.

5) CtF (T&E): In this approach, we finetune the model with our **CtF** (T&E) data.

Metrics For the associative reasoning task, given the rule head $r_h(X, Y)$, we pose a question in natural language to identify Y, where Y is the *golden entity* for the question.

Dataset	#Entity	#Relation	#Rule	#Samples	Avg. Sample Hops	
					CtF	CtF(T&E)
Overall	6611	520	644	2793	2.43	5.34
2-hop	4748	203	326	1993	2	4.4
3-hop	978	159	172	400	3	6.6
4-hop	885	158	146	400	4	8.8

Table 1: Statistics of the training data in anonymized setting of KGREASON dataset

We use exact match accuracy as our metric. For each subset of our test data, the evaluation formula is as follows:

$$score(T) = \frac{E}{L(T)} \tag{2}$$

where T represents the test dataset, E is the number of samples for which the predicted entity matches the golden answer exactly, and $L(T)$ is the total number of samples in the test dataset T.

4.2 Evaluation of Dataset Quality

Since each of our data sample is derived from instances governed by specific rules, we assess the quality of the dataset by evaluating the quality of these rules. We established three metrics to evaluate the quality of the rules, which are:

Validity, which is used to evaluate the sufficiency and necessity of the rule. A rule that is both sufficient and necessary receives 2 points, one that is sufficient but not necessary receives 1 point, and one that is neither sufficient nor necessary receives 0 points.

Plausibility, which is used to evaluate whether the rule is reasonable, i.e., whether the instances applied by the rule can correctly infer the target knowledge. A reasonable rule receives 1 point, while an unreasonable rule receives 0 points.

Practicality, which is used to evaluate the applicability of the rule, i.e., whether humans would likely choose to use this rule when making inferences. A rule that is highly likely to be used receives 2 points, one that is applicable under certain conditions receives 1 point, and one that is unlikely to be used receives 0 points.

Based on the above three metrics, we manually scored each rule and calculated the average value of each metric for rules of different lengths. The result is shown in Table 2

From the evaluation results, we can observe that: The validity of the rules is slightly greater than 1, indicating that most of the rules are sufficient but not necessary. All three sets of rules with different lengths achieved high scores in Plausibility, demonstrating the validity of our rules. In terms of practicality, all three sets of rules scored above 1.5, indicating that although exceptions may exist, they are generally natural and applicable in common-sense scenarios, and thus align with human's experience.

	Validity	Plausibility	Practicality
2-hop	1.31	0.94	1.83
3-hop	1.23	0.90	1.79
4-hop	1.17	0.88	1.68

Table 2: Evaluation results of different rule lengths across three metrics.

4.3 Results in the Anonymized Settings

We conduct experiments using each method mentioned in Section 4.1. For our **CtF** and **CtF** (T&E) methods, we construct three versions of data for each method using rules of different lengths. This approach allows us to explore the relationship between rule length and the model's knowledge reasoning ability. The results of the experiments are shown in Table 3.

CtF Effectively Improves LLMs' Knowledge Reasoning Ability. Our results show that both **CtF** and **CtF** (T&E) consistently outperform the baselines on all test datasets. Notable, given some **CtF** examples, ICL-**CtF** generally outperforms vanilla CoT. However, it still yield relatively low scores, suggesting that LLMs without fine-tuning struggle with associated reasoning based on their internal knowledge, despite having key information within their parameters. Regarding fine-tuning, while training with CoT proves useful for many reasoning tasks, such as logical reasoning, it does not lead to noticeable improvement in associated reasoning.

With Trial & Error, CtF (T&E) Further Improves Performance in OOD Settings. In **CtF**,

Model	Method	Rule Length	ID				OOD			
			2-hop	3-hop	4-hop	all	2-hop	3-hop	4-hop	all
Mistral-7b	Vanilla CoT	-	5.48	6.97	4.48	5.64	4.98	5.47	4.98	5.15
	ICL-CtF	2	7.46	7.96	7.46	7.63	7.96	6.97	6.97	7.30
		2,3	7.96	7.96	6.97	7.63	8.46	5.97	7.46	7.30
		2,3,4	6.47	6.97	7.46	6.97	6.97	4.98	7.46	6.47
	CoT Learning	-	6.47	5.97	5.47	5.97	6.97	6.97	6.97	6.64
	CtF	2	11.94	14.93	12.94	13.27	16.92	7.96	5.97	10.28
		2,3	15.92	19.90	16.9	17.58	16.42	8.96	14.93	13.43
		2,3,4	13.43	15.92	21.89	17.08	12.44	16.92	22.89	17.41
	CtF(T&E)	2	11.94	8.90	9.95	10.28	14.93	11.94	11.94	12.94
		2,3	**18.41**	**24.80**	16.92	**20.07**	**20.90**	**23.88**	19.90	**21.56**
		2,3,4	14.93	17.91	**22.89**	18.57	18.41	19.90	**25.87**	21.39
Llama3-8b	Vanilla CoT	-	4.98	6.97	8.96	6.97	5.97	8.96	5.97	6.97
	ICL-CtF	2	8.46	6.97	6.97	7.46	7.96	6.97	6.97	7.30
		2,3	7.46	7.96	6.97	7.46	7.96	7.46	5.97	7.13
		2,3,4	7.46	8.96	8.46	8.29	6.97	8.96	8.96	8.29
	CoT Learning	-	6.97	7.47	6.47	6.97	7.97	8.30	6.97	7.75
	CtF	2	**17.41**	12.94	13.93	14.76	13.43	8.96	7.96	10.12
		2,3	15.42	19.90	14.93	16.75	19.40	15.92	13.93	16.42
		2,3,4	16.42	18.91	**21.89**	19.07	10.45	17.91	19.90	16.09
	CtF(T&E)	2	11.94	15.92	15.92	14.59	18.91	16.92	11.94	15.92
		2,3	11.94	**20.90**	16.92	16.58	**21.39**	**23.88**	18.91	**21.39**
		2,3,4	16.92	19.90	20.90	**19.24**	12.94	21.89	**22.89**	19.24

Table 3: Results(%) of the experiments on anonymized setting with different methods and different rule length. The best results are **bolded**, and the second best ones are underlined.

the scores for ID dataset are generally higher than those for OOD dataset, demonstrating the phenomenon of *rule overfitting*, where LLMs rely on reasoning paths encountered during training. This rule overfitting can lead to hallucinations and a loss of generalization on OOD dataset. **CtF** (T&E) surpasses **CtF** on most datasets, with a particularly significant improvement on OOD dataset. This suggests that **CtF** (T&E) enables LLMs to consider more appropriate rules for questions, rather than blindly applying previously encountered rules.

Learning With Long Rules. To study the impact of rule length, we conduct experiments using datasets with varying rule lengths. We observe that on the 2-hop and 3-hop test datasets, **CtF** (T&E)-3-hop achieves the highest scores, while **CtF** (T&E)-4-hop achieves the highest score on the 4-hop test dataset. As the rules in the data lengthen, the model's performance does not continuously improve; instead, training with rules having a maximum length of 4-hop actually decreases performance.

To find the reason of this, we calculate the length of rules the model uses in the outputs, the result is shown in Figure 3 From the results, the model tend to use longer rules when training with longer rules.

When a shorter path is available, using a longer reasoning path increases the difficulty of reasoning, as each step in the reasoning process requires the model to explore and make decisions.

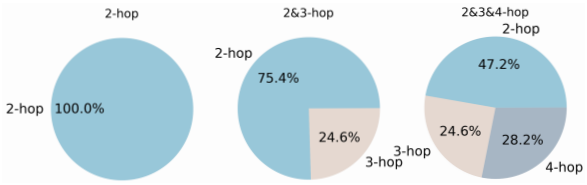

Figure 3: Proportion(%) of various rule lengths in the model's output when trained with data of different rule lengths.

Training with longer rules helps LLMs learn to use more complex rules in reasoning. However, longer rules are not always better. Training with them can lead to a tendency to use longer rules during reasoning, even when shorter or simpler paths are available, potentially decreasing performance on simpler tasks.

Error Analysis We conduct additional experiments to analyze the reasons for the model's in-

correct predictions in ID and OOD settings. We categorize the types of errors based on the first incorrect step in the model's reasoning process. The **Rule** error indicates that the model selects an inappropriate reasoning path, while the **Fact1** error indicates that the model uses an incorrect Y in the fact $r_1(X, Y)$. The results are presented in Table 4.

	Rule	Fact1	Fact2
ID	34.78	38.89	26.32
OOD	63.15	23.27	13.57

Table 4: Proportion(%) of different types of errors in ID and OOD setting across rule, fact1, and fact2.

The results indicate that on ID dataset, most errors are caused by the model using incorrect facts. This is likely because the reasoning paths in the ID dataset also appear in the training data. In contrast, on OOD dataset, more errors are attributable to the reasoning paths selected by the model. Regarding fact errors, we note that the model frequently hallucinates during the initial stages of entity selection. We believe this occurs because, in the later stages of reasoning, the model's knowledge selection scope becomes more restricted as it incorporates additional supporting facts

Specifically, we find that in some error samples, the model selects an appropriate reasoning path and uses correct facts, yet still arrives at an incorrect entity. This occurs because for a fact $r_h(X, Y)$, the combination of r_h and X can correspond to multiple Ys. Although the model deduces a result different from the ground truth due to using different facts, the reasoning process remain valid.

4.4 Results in the Regular Settings

Downstream Tasks with Regular Settings In regular settings, we train the model with regular data that utilizes real-world entities for data construction, and further test the model on downstream tasks.

The results of regular settings are shown in Table 5.

Model	Method	ID	OOD
Mistral-7b	Vanilla CoT	0.00	0.00
	ICL-CtF	5.50	6.20
	CtF	27.00	21.89
	CtF (T&E)	21.33	22.22

Table 5: Performance(%) of different methods on regular setting.

The results from the regular setting validate the conclusions drawn from the anonymized experiments: When prompted with **CtF** examples, ICL-**CtF** outperforms vanilla CoT on both ID and OOD dataset. Both **CtF** and **CtF** (T&E) enhance the LLM's associative thinking capabilities. Furthermore, **CtF** (T&E) further reduces the LLM's rule dependency, leading to improved performance on OOD dataset.

To further investigate the generalization of **CtF**, we tested our **CtF** exploration method on other popular benchmarks. The results of the downstream tasks are shown in Table 4. The results indicate that **CtF** (T&E) outperforms the baseline on three commonsense reasoning benchmarks, suggesting that **CtF generalizes well to other reasoning tasks that require various types of knowledge**, such as world knowledge.

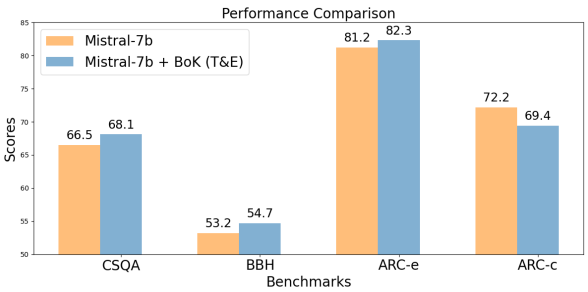

Figure 4: Performance comparison of Mistral-7b and Mistral-7b + **CtF** (T&E) on four reasoning benchmarks.

5 Conclusion

In this paper, we study associative thinking in LLMs. Leveraging KGs, we propose **CtF** (*i.e.*, **C**onnecting **t**he **F**acts) to enhance LLMs' associative thinking. Specifically, we construct the KGREASON dataset based on compositional rules and facts in KGs, and fine-tune LLMs on KGREASON through both behavior cloning and trial-and-error methods. Results of extensive experiments show that fine-tuning on KGREASON effectively improves LLMs' associative thinking capabilities, and the trial-and-error strategy enables LLMs to better simulates the human way of internal knowledge exploration. Moreover, **CtF** also benefits LLMs' performance in various reasoning tasks, showcasing the broader value of associative thinking for LLMs.

Limitations

Evaluation of the Associative Reasoning Ability Because knowledge reasoning in LLMs remains underexplored in previous studies and no public datasets or benchmarks are suitable for our task, the training and testing of the model's associative thinking ability are conducted using the dataset KGREASON that we constructed. To address this concern, we strive to ensure the diversity of our dataset.

Data of the Regular Setting is Model-Specific In the regular setting, the entities in the data represent real-world knowledge. To prevent data leakage, we perform knowledge probing on each model, making the regular setting data model-specific. In contrast, for the anonymized setting, we construct data that requires knowledge injection during the continuous pretraining stage but is applicable to all models.

Ethics Statements

In this paper, we propose Connecting-the-Facts, which includes both data construction and two model learning method to enhance the associative thinking ability of LLMs. Since we utilize advanced LLMs for sample generation, it is inevitable that biases present in the LLMs may influence the knowledge. To address these ethical concerns, we will optimize the rule mining method and better align the model's output with human cognition.

References

AI@Meta. 2024. Llama 3 model card.

John R Anderson. 2013. *The architecture of cognition*. Psychology Press.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Roger E Beaty and Yoed N Kenett. 2023. Associative thinking at the core of creativity. *Trends in cognitive sciences*, 27(7):671–683.

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.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

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:1803.05457v1*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Allan M Collins and M Ross Quillian. 1969. Retrieval time from semantic memory. *Journal of verbal learning and verbal behavior*, 8(2):240–247.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Daniel Kahneman. 2011. Thinking, fast and slow. *Farrar, Straus and Giroux*.

Sarnoff Mednick. 1962. The associative basis of the creative process. *Psychological review*, 69(3):220.

Pronita Mehrotra, Aishni Parab, and Sumit Gulwani. 2024. Enhancing creativity in large language models through associative thinking strategies. *arXiv preprint arXiv:2405.06715*.

Vijay V Raghavan, Venkat N Gudivada, Venu Govindaraju, and Calyampudi Radhakrishna Rao. 2016. *Cognitive computing: Theory and applications*. Elsevier.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

E Tulving. 1972. Episodic and semantic memory. *Organization of memory/Academic Press*.

Kurt VanLehn. 1996. Cognitive skill acquisition. *Annual review of psychology*, 47(1):513–539.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

A Details of CtF data construction

A.1 Rule Mining

In the rule mining step, we first use a breadth-first search algorithm to find composite rule instances from raw KGs, the algorithm we use is as Algorithm 1.

Algorithm 1: BFS for rule instances

```
for each triple (A, r1, B) do
    if B is in triplets then
        for each (r2, C) for the value of B do
            if A is in triplets then
                for each (r3, C') for the value of A do
                    if C' == C then
                        return combination
```

A.2 Knowledge Selection

In the anonymized setting, after identifying the supporting facts for the rules, additional processing of the data is required. If the head of one instance is part of another instance's body, using this instance for sample generation and model training can lead to data leakage, resulting in an unfair evaluation. To address this issue, we separate the head and body parts within the instances. By traversing all instances, we create a set for the body facts. If an instance's head fact appears in this set, the instance is at risk of data leakage and is therefore discarded.

A.3 Sample Generation

Knowledge Dataset The knowledge dataset is used to inject knowledge into LLMs exclusively in anonymized settings. It is employed during the continuous pretraining stage and, as such, is presented in the form of a corpus.

For each entity in our knowledge base, we establish a mapping to all related facts. Using these facts, we prompt advanced LLMs to generate a descriptive paragraph, encapsulating the knowledge about the entity. The prompt we use is Prompt A.3

Prompt 1: prompt for knowledge corpus generation

Rewrite the following sentence as a paragraph describing {{entity}}, taking care not to change the original meaning of the sentence or lose information: {{fact1}} {{fact2}} ... {{factn}}

To enhance the model's ability to memorize knowledge, we group related facts of each entity into sets of 10 and input them into the LLMs to generate a corpus. For each entity, we prompt the LLM to generate four different versions of the knowledge corpus. Furthermore, to improve the model's extraction of knowledge from its internal knowledge base, we integrate the corresponding **CtF** data into the pretraining corpus. Thus, each subset of the **CtF** dataset with different rule lengths has a corresponding knowledge dataset for pretraining.

CtF dataset The **CtF** dataset is used for model learning in the supervise finetuning stage, and has both anonymized and regular setting. We have 3 steps to generate samples for **CtF** dataset:

1) *Relation Template Generation* For each relation, we generate a template sentence which describes the relation of the two entites in the atom. e.g., having an atom *CitizenOf*(X, Y), the template is $\{\{X$ is a citizen of $Y\}\}$. The prompt we use to generate the relation templates is as follows:

Prompt 2: prompt for relation template generation

You will be given a triple and you should output a sentence which describes the relation between the two entities in the triple, here are some examples:
Triple: (<ENT1>, citizen of, <ENT2>)
Output: <ENT1> is a citizen of <ENT2>.
...
Triple: {{triple}}
Output:

2)*Question Template Generation* For each relation in the rule head, we generate a question template for it. Considering the sufficiency and non-necessity of the rules, and to ensure that each posed question has only one correct answer, we prompt the LLMs to generate a question with possible tone and only question for the unique entity in the atom. The prompt we use to generate question template is as follows:

Prompt 3: prompt for question template generation

You will be provided with a triple, and you should formulate a question that queries the relationship between the two entities. Ensure that the question you generate is in possible tone and has only one correct answer. Here are some examples:
Triple: (<ENT1>, citizen of, <ENT2>)
Output: Which country may <ENT1> be a citizen of?
...
Triple: {{triple}}
Output:

3)*Sample Generation* The sample of **CtF** dataset

is in the form of question-answer pairs, so for a sample, we generate question and answer respectively.

For question generation, we substitute the rule head into the corresponding template. For answer generation, we replace each fact in the rule body and rule head, connecting all resulting sentences into a template answer. We then prompt the LLM to polish these sentences into natural language, which serves as the answer for this sample. The prompt we use to generate the natural language answer is as follows:

Prompt 3: prompt for answer generation

Rewrite the following sentence into natural language, take care not to change the original meaning or lose information: {{sentence}}

A.4 Trial&Error Data Construction

Algorithm 2: CtF (T&E)

Data: knowledge graph G, rule head $r_h(X, Y)$, large language model M
Result: exploration process P

```
t ← 1;
while True do
    // Select candidate rule
    R_t ← CandidateRule(r_h);
    // Search for supporting facts
    for fact ∈ R_t(rule_body) do
        if not IsFactExist(fact, M) then
            RecordError();
            t ← t + 1;
            continue;
    return P
```

B Experiment Settings

B.1 Details of Datasets

Test Dataset of KGREASON The statistic of the test dataset of KGREASON is shown in Table 6

Benchmarks in Downstream Tasks

- **CommonsenseQA(CSQA)** (Talmor et al., 2019) CommonsenseQA is a new multiple-choice question answering dataset that requires different types of commonsense knowledge to predict the correct answers . It contains 12,102 questions with one correct answer and four distractor answers.
- **AI2 Reasoning Challenge (ARC)** (Clark et al., 2018) ARC is a dataset of 7,787 genuine grade-school level, multiple-choice science questions, assembled to encourage research in advanced question-answering. The dataset is partitioned into a Challenge Set(ARC-c) and an Easy Set(ARC-e).
- **BIG-Bench Hard(BBH)** (Suzgun et al., 2022) BBH is a diverse evaluation suite that focuses on tasks believed to be beyond the capabilities of current language models. It focus on a suite of 23 challenging BIG-Bench tasks for which prior language model evaluations did not outperform the average human-rater.

B.2 Details of Methods

Vanilla CoT In vanilla CoT, we prompt the model simply with {{let's think step by step}}. The prompt we use in as follows:

Prompt 4: prompt for Vanilla CoT

Instruction: associaitve reasoning is the process of using known knowledge to infer new knowledge. You will be given a question of associaitve reasoning task, use you internal knowledge to reason out the answer. Let's think step by step.
Question: Which state may Lily live in?

ICL-CtF In ICL-**CtF**, we prompt the model with 6 examples from the **CtF** dataset. In different rule length setting, we use examples with different rule length, the detail is shown in Table 7

The prompt we use is as follows:

Prompt 5: prompt for ICL-CtF

Instruction: associaitve reasoning is the process of using known knowledge to infer new knowledge. You will be given a question of associaitve reasoning task, use you internal knowledge to reason out the answer. Here are some examples:
{{Example1}}
...
{{Example6}}
Question: {{question}}

CtF The prompt we use for **CtF** is as follows:

Prompt 6: prompt for CtF

Instruction: associaitve reasoning is the process of using known knowledge to infer new knowledge. You will be given a question of associaitve reasoning task, use you internal knowledge to reason out the answer.
Question: {{question}}

Setting	Rule Length	#Entity	#Relation	#Rule	#Sample
ID	2-hop	460	135	147	501
	3-hop	307	96	70	501
	4-hop	360	118	73	501
OOD	2-hop	446	43	28	501
	3-hop	267	66	36	501
	4-hop	353	94	41	501

Table 6: Statistics of the test dataset of KGREASON

Rule Length	# Example		
	2-hop	2&3-hop	2&3&4-hop
2	6	0	0
3	3	3	0
4	2	2	2

Table 7: #Examples in ICL-CtF's prompting

CtF (T&E) The prompt we use for **CtF** (T&E) is as follows:

Prompt 7: prompt for **CtF** (T&E)

Instruction: associaitve reasoning is the process of using known knowledge to infer new knowledge. You will be given a question of associaitve reasoning task, use you internal knowledge to reason out the answer. If you don't have the knowledge of the supporting fact during reasoning, you should backtrace and change to another path until you can get the answer.
Question: {{question}}

C Case Study

C.1 Examples of Rules

Table 8 shows examples of the rules we mine from KGs.

C.2 Examples of Knowledge Dataset

Table 9 shows examples of the knowledge dataset we use in continuous pretraining stage.

C.3 Examples of CtF Dataset

Table 10 shows examples of data in **CtF** setting. Table 11 shows examples of data in **CtF** (T&E) setting.

C.4 Cases of Different Types of Errors

Table 12 shows different types of errors in error analysis, including rule error, fact1 error and fact2 error.

Rule Length	Rules
2-hop	Country(X,Y)←PlaceOfBirth(Z,X)∧CountryOfCitizenship(Z,Y) Contry(X,Y)←MemeberOfSportsTeam(Z,X)∧CountryOfCitizenship(Z,Y) CastMember(X,Y)←OriginalLanguageOfFilm(X,Z)∧LanguagesSpoken(Y,Z)
3-hop	Country(X,Y) ← PlaceOfBirth(Z,X) ∧ ResidentOf(Z,W) ∧ Country(W,Y) Country(X,Y) ← PlaceOfBirth(Z,X) ∧ EducatedIn(Z,W) ∧ Country(W,Y)
4-hop	Country(X,Y)← PlaceOfBirth(Z,X) ∧ Spouse(Z,W) ∧ PlaceOfBirth(W,V) ∧ CountryOfCitizenship(W,Y) DistantRelative(A,B) ← PlaceOfBirth(A,X) ∧ ParentOf(Y,A) ∧ CountryOfCitizenship(Y,Z) ∧ Spouse(A,W) ∧ ChildOf(B,W)

Table 8: Examples of different length of rules we mine

Knowledge
Excn, a company known for its innovative approach and cutting-edge solutions, has its headquarters situated in Dyxeti. This strategic location serves as the central hub for Excn's operations, allowing the company to efficiently coordinate its various departments and teams. Dyxeti's vibrant business environment and access to top talent make it an ideal setting for Excn to thrive and continue its mission of driving progress and success in the industry.
Ccmr is the location where the headquarters of Bplx is situated. The presence of Bplx's headquarters in Ccmr is significant as it serves as the central hub for the organization's operations and decision-making. Ccmr, with its strategic position, provides Bplx with easy accessibility to its stakeholders and allows for efficient management of the business. This centralized location in Ccmr ensures effective coordination among various departments and facilitates seamless communication between Bplx and its global network.
Nryxg is the official language of Gexdzjp, serving as the primary means of communication within the region. It is the designated language used for official documents, government proceedings, and educational instruction, reflecting the cultural and linguistic identity of the people of Gexdzjp. Through the use of Nryxg, individuals in Gexdzjp are able to effectively communicate and connect with one another, fostering a sense of unity and shared understanding among its inhabitants.

Table 9: Examples of the knowledge dataset

Rule Length	**Question**	**Answer**
2-hop	Which country might Anykid be a citizen of?	Cckqlvy has Anykid as a part of their team. Cckqlvy is from the country Vevedgta. Therefore, Anykid may be a citizen of Vevedgta. Consequently, Vevedgta is the correct answer.
3-hop	Which country may Pzdazwi belong to?	Mjzk is a citizen of Pzdazwi and is a cast member of Kqts, who is from the country Vevedgta. Therefore, it is possible that Pzdazwi is also from the country Vevedgta.
4-hop	What may be the original language of Yvag?	Brq is fluent in speaking, writing, or signing Owiyxpqf languages. Brq is a actor in Vffdgv, which is produced by Lmljpgzx. Yvag is directed by Lmljpgzx. Therefore, it is possible that the original language of Yvag is Owiyxpqf.

Table 10: Examples of CtF method in CtF dataset

Rule Length	Question	Answer
2-hop	To which country does Qzdhsdq belong?	To find the answer, we can follow the reasoning path: official_language(Y, X) ← language_of_work_or_name(A, X), country_of_origin(A, Y). Lzdxfzxw is from Vevedgta, but since we are unsure of Lzdxfzxw's military branch, this path is not applicable. Let's consider a different path: country(X, Y) ← military_branch(A, X), country_of_citizenship(A, Y). Bmezedu serves in the military branch Qzdhsdq, which is linked to Vevedgta. Therefore, we can conclude that Qzdhsdq is associated with Vevedgta. Hence, the answer is Vevedgta.
3-hop	What might be the language of Yjuphmb?	If a person has a work X and is an author of a book Y, then the language of book Y could be the same as the language of work X. In this case, Yjuphmb is a work of Kvyps, Kvyps is the author of Tkbz, and the language of Tkbz is Xsbfi. Therefore, it is possible that the language of Yjuphmb is Xsbfi.
4-hop	Which country might Liose from?	To find the answer, we can use the reasoning path: Country(X,Y) ← PlaceOfBirth(Z,X) ∧ EducatedIn(Z,W) ∧ Country(W,Y). However, since we are unsure of where Liose was educated, this path is not viable. Let's consider an alternative path: Country(X,Y)← PlaceOfBirth(Z,X) ∧ Spouse(Z,W) ∧ PlaceOfBirth(W,V) ∧ LocateIn(W,Y). Since Liose is born is Bhsz and Ari is her spouse, Ari was born in Zhww which locates in Vevegde, it is plausible that Liose is a citizen of Vevegde

Table 11: Examples of CtF (T&E) method in CtF dataset

Error Type	Question	Model Output	Explain
Rule	Can you tell me the country where Nzwscwm is from?	Nzwscwm is the country that the author of Wcsa is a citizen of. Wcsa is from the country Vevedgta, so it's possible that Nzwscwm is also from Vevedgta. So the answer is Vevedgta.	The reasoning path is not reasonable.
Fact1	To which country does Ztgl possibly belong?	Ztgl is the head coach of Mfqep. Mfqep is from the country Zxdxcgh. Therefore, it is possible that Ztgl also belongs to the country Zxdxcgh. Thus, Zxdxcgh is the answer.	Ztgl is not a coach, the first fact is wrong.
Fact2	What is the language Arstkb use?	Arstkb has cast member Qoztebgc, who speaks the language Crbzovw. This means that Arstkb use Crbzovw language. Thus, Crbzovw is the answer.	Qoztebgc does not speak Crbzovw, the second fact is wrong.

Table 12: Examples of three types of errors