Reward Engineering for Generating Semi-structured Explanation

Jiuzhou Han[♮] **Wray Buntine**[♭] **Ehsan Shareghi**[♮]
[♮] Department of Data Science & AI, Monash University
[♭] College of Engineering and Computer Science, VinUniversity
jiuzhou.han@monash.edu wray.b@vinuni.edu.vn
ehsan.shareghi@monash.edu

Abstract

Semi-structured explanation depicts the implicit process of a reasoner with an explicit representation. This explanation highlights how available information in a specific query is utilised and supplemented with information a reasoner produces from its internal weights towards generating an answer. Despite the recent improvements in generative capabilities of language models, producing structured explanations to verify a model's true reasoning capabilities remains a challenge. This issue is particularly pronounced for not-so-large LMs (e.g., FLAN-T5-XXL). In this work, we first underscore the limitations of supervised fine-tuning (SFT) in tackling this challenge, and then introduce a carefully crafted reward engineering method in reinforcement learning (RL) to better address this problem. We investigate multiple reward aggregation methods and provide a detailed discussion which sheds light on the promising potential of RL for future research. Our proposed method on two semi-structured explanation generation benchmarks (ExplaGraph and COPA-SSE) achieves new state-of-the-art results. [1]

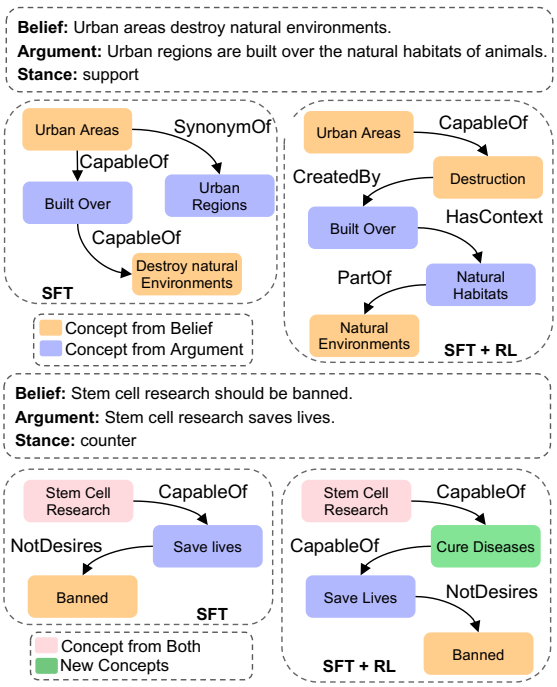


Figure 1: Given the belief and argument, the task is to predict the stance (support/counter) and generate an explanation graph representing the reasoning process. The explanation graph under SFT+RL is more expressive.

1 Introduction

Language models have shown great capability in complex reasoning tasks (Touvron et al., 2023b; Bubeck et al., 2023; Touvron et al., 2023a; Chung et al., 2022; Brown et al., 2020; Yang et al., 2018; Lin et al., 2019). Despite their proficiency in generating accurate results, a comprehensive assessment of the models' true capabilities in reaching the correct output necessitates an explainable mechanism. In this spirit, generating structured explanations (Saha et al., 2021; Brassard et al., 2022) is a viable approach as they explicitly representing the relationships between facts employed during reasoning, and are amenable to evaluation. Unstructured natural language explanations lack these aspects. Figure 1 illustrates two examples of stance detection task, where the structured outputs are intended to explain the stance.

For this purpose, Saha et al. (2021) propose to use multiple models for predicting answer, internal nodes, external nodes and relations. Cui et al. (2023) incorporate a generative pre-training mechanism over synthetic graphs by aligning input pairs of text-graph to improve the model's capability in generating semi-structured explanation. Both works train separate models for prediction of response, and generation of explanations. It is reasonable to expect that even a moderately-sized language model such as FLAN-T5-XXL (Chung et al., 2022) should be capable of producing both answers and the corresponding structured explanations. We

[1] Our code is available at https://github.com/Jiuzhouh/Reward-Engineering-for-Generating-SEG.

investigate this in our work. In parallel, Large LMs at the scale of GPT-4 (OpenAI, 2023) have shown a great capability in producing both an answer and an unstructured reasoning trace through methods like Chain-of-Thought (CoT) (Kojima et al., 2022; Wei et al., 2022). One might hope that an ideal structured representation of the reasoning trace could also be comfortably surfaced via in-context prompting of LLMs. But it has been demonstrated that LLMs struggle to generate structured format output (Han et al., 2023). We empirically verify this struggle in the context of generating structured explanations.

Our objective is to equip moderately-sized LMs with the ability to not only provide answers but also generate structured explanations. To facilitate this, we first utilise supervised fine-tuning (SFT) as the de-facto solution. We then turn our focus to RLHF[2] as a mechanism to further align the explanations with ground-truth on top of SFT. We design a reward engineering method in RL and explore multiple reward aggregations that leverage both reward modelling and reward metrics. Our proposed method, implemented on the backbone of a FLAN-T5 (Chung et al., 2022), achieves new state-of-the-art results on two benchmarks, ExplaGraph (Saha et al., 2021) and COPA-SSE (Brassard et al., 2022). As a byproduct, our empirical comparison also highlight the limitations of LLMs like GPT-4 and GPT-3.5-instruct to succeed at structured explanation generation (SEG). Furthermore, we delve into a discussion on RL for SEG and highlight what reward metrics work better, and spotlight the challenges (i.e., reward hacking) of balancing the dynamic of policy optimization.

We hope the findings of our work to shed light on both challenges and potentials of RL in SEG as well as the broader space of graph generation.

2 Semi-structured Explanation

Structured explanation refers to a specific form of explanation that highlights the underlying decision-making processes of the model via an explicit representation of relationships between different reasoning factors. In this section, we briefly review different forms of explanations and introduce the semi-structured explanation tasks of our interest.

[2]Throughout this paper, we use RLHF and RL interchangeably. Noting that our framework does not involve human feedback alignment, but leverages the same framework to create a better alignment between LM's predictive behaviour and ground-truth.

2.1 Related Work

Explanation in Explainable NLP literature (Wiegreffe and Marasovic, 2021) can be categorised into three major types: (I) *Highlight Explanations* are subsets of the input elements which explain a prediction. For textual NLP tasks, the elements are usually words, phrases or sentences. The representative highlight explanations datasets are WikiQA (Yang et al., 2015), HotpotQA (Yang et al., 2018), CoQA (Reddy et al., 2019), BoolQ (DeYoung et al., 2020), which have different granularities from words to sentences. (II) *Free-text Explanations* are texts in natural language that are not constrained to the input elements, hence more expressive and readable. It is a popular explanation type for both textual and visual-textual tasks with benchmarks like VQA-E (Li et al., 2018), e-SNLI (Camburu et al., 2018), WinoWhy (Zhang et al., 2020), ECQA (Aggarwal et al., 2021). (III) *Semi-structured Explanations* are a specific format of explanations which are written in natural language but not entirely free-form. Semi-structured explanations have aroused the public attention in recent years because they combine the properties of both highlight and free-text explanations. Semi-structured explanations do not have one unified definition, but represent explanations in a (semi-)structured format. Benchmarks like WordTree (Jansen et al., 2018; Xie et al., 2020), eQASC (Jhamtani and Clark, 2020), ExplaGraph (Saha et al., 2021), COPA-SSE (Brassard et al., 2022) fall under this category.

2.2 Tasks

Since WordTree is based on lexically overlapping sentences and eQASC is based on natural language reasoning chain, neither of them have a unified form of semi-structured explanations. In this work, we focus on two semi-structured explanation tasks: ExplaGraph (Saha et al., 2021) and COPA-SSE (Brassard et al., 2022). Both of them are question-answering tasks and the explanations contain concepts and relations in triple format, which are clear to understand and easy to evaluate. We provide a brief overview of them in what follows and an example of each task in Table 1.

ExplaGraph Given a belief and an argument, the task requires a model to predict whether a certain *argument* supports or counters a *belief*. Each instance in the data is also accompanied by a commonsense explanation graph which reveals an in-

ternal reasoning process involved in inferring the predicted stance. The explanation graph is a connected directed acyclic graph (DAG), in which the nodes are concepts (short English phrase) and relations are chosen based on ConceptNet (Liu and Singh, 2004). Concepts are either internal (part of the belief or the argument) or external (part of neither but necessary for the explanation). Semantically, the explanation graphs are commonsense-augmented structured arguments that explicitly support or counter the belief.

COPA-SSE Given a premise and a question, the task of COPA-SSE is to select from two options the one that more plausibly has a causal relation with the premise, and generate a corresponding semi-structured commonsense explanation. The semi-structured explanation is created by crowd workers, which contains multiple triples in [head, relation, tail] format. The nodes are concepts and relation are from ConceptNet as well. Different from ExplaGraph, the semi-structured explanation in COPA-SSE is not necessary to be a DAG.

The difficulty of these two tasks is that first the model needs to correctly understand the question and answer it, then generate a reasonable and semantically correct semi-structured explanation. The answers are in a form of an unstructured natural language, while the explanations are of structured format. Tasking a model to generate both modalities, as we will show in the experiment section, imposes a major challenge.

3 Reward Engineering for SEG

Motivated by the success of reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Dubois et al., 2023; Touvron et al., 2023b) in LLMs, we propose to use RL for semi-structured explanation generation task. To achieve this, we design a reward engineering method by incorporating different sources of reward. The RLHF typically begins with a pre-trained LLM, which is fine-tuned with supervised learning on a downstream task, namely the SFT model. The process has two phases: the reward modelling phase and the RL fine-tuning phase. Our reward engineering is designed to improve the reward modelling phase. The objective of RL fine-tuning is to optimize the policy model against a reward model. In our work, we use proximal policy optimization (PPO) (Schulman et al., 2017).

ExplaGraph
Input: Predict the stance and generate an explanation graph given the belief and argument. Belief: People around the world are able to connect thanks to social media. Argument: Before social media existed there was no quick and easy way to connect with others globally.
Output: support (social media; causes; connection)(connection; used for; people)(people; at location; globally)(connection; made of; fast connection)
COPA-SSE
Input: Given the premise, choose from a or b and generate an explanation graph. Premise: The man woke up late. What happened as a RESULT? a: He missed an appointment with the dentist. b: He made an appointment with the dentist.
Output: a [[The man, HasProperty, sleepy], [Sleepiness, Causes, oversleeping], [oversleeping, Causes, missing events], [a dentist appointment, HasProperty, an event]]

Table 1: An example of the input-output for each task. The explanations are presented as a set of triples of [head, relation, tail]. These triples form: a connected graph in the case of ExplaGraph, or a semi-structured set in the case of COPA-SSE.

3.1 Reward Model

In the reward modelling phase, given the input and a generated output, the reward model, R_ϕ, generates a single scalar representing its overall quality. To train a reward model, first we need to collect the paired preference data. In this work, we generate the paired data using the SFT model, which is fine-tuned on the semi-structured explanation task. The SFT model generates the outputs from the training data, then we pair the generated output with its corresponding reference. To improve the quality of the paired preference data, we filter out the pairs where the generated output is the same as the reference. In each pair, the reference is regarded as the preferred data. The filtered paired preference data is then used to train the reward model.

3.2 Reward Metric

In addition to collecting the reward from the reward model, we propose to collect another reward from evaluation metrics. This metric reward can explicitly reflect the quality of the generated output which is naturally complementary to the reward from the reward model. Since the semi-structured explanation is represented in format of a set of triples

(i.e., [head, relation, tail]), following the previous work (Saha et al., 2021), we consider each triple as a sentence and use the existing text matching metrics to calculate the graph matching score. Specifically, BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2019) are extended to Graph-BLEU, Graph-ROUGE and Graph-BERTScore. Another metric is Graph Edit Distance (GED) (Abu-Aisheh et al., 2015), which takes into account the graph structure of the explanation.

3.3 Reward Aggregation

The reward model R_ϕ takes input prompt x and generated output y, and generates a single scalar $R_\phi(x, y)$. For the metric reward, given the generated output y and the reference y', the evaluation metric R_m is used to calculate a metric score as the reward $R_m(y, y')$. To aggregate two rewards, an important premise is that the order of magnitude of two rewards should not have too much difference (e.g., 0.01 vs 100), otherwise the effect of one reward could be washed away. To regulate this, we explore various aggregation configurations for the final reward $R(x, y, y')$,

$$R(x, y, y') = \alpha R_\phi(x, y) + (1 - \alpha) R_m(y, y') \quad (1)$$

where α is a coefficient to control the weights of different rewards. In RL phase, we use the total reward to provide feedback to the language model. In particular, we formulate the following optimization problem,

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)} \left[R(x, y, y')\right] - \beta \mathbb{D}_{\mathrm{KL}} \left[\pi_\theta(y \mid x) \| \pi_{\mathrm{ref}}(y \mid x)\right] \quad (2)$$

where β is the KL coefficient controlling deviation from the base reference policy π_{ref} (the initial SFT model). In practice, the language model policy π_θ is also initialised to the initial SFT model.

4 Experiment

4.1 Datasets and Evaluation Metrics

ExplaGraph (Saha et al., 2021) contains 2,368/398/400 samples as training/dev/test set. Since the labels of the test set are not public, we provide the evaluation results on dev set.[3] As shown in Table 1, for ExplaGraph, the instruction we use is "*Predict the stance and generate an explanation graph given the belief and argument.*" We concatenate the instruction with the belief and argument as input, and the output is a stance concatenated with a linearised explanation graph. We use the same evaluation metrics provided in the ExplaGraph (Saha et al., 2021): Stance Accuracy (SA), Structural Correctness Accuracy of Graphs (StCA), Semantic Correctness Accuracy of Graphs (SeCA), Graph-BertScore (G-BS), Graph Edit Distance (GED), Edge Accuracy (EA).

COPA-SSE (Brassard et al., 2022) contains 1,000/500 samples as training/test set. Since each instance in COPA-SSE contains multiple human-rating semi-structured explanations, we only use the one with the highest rating score as the reference. For COPA-SSE, the instruction we use is "*Given the premise, choose from a or b and generate an explanation graph.*" This instruction is concatenated with the premise and two options as input. The output is the answer along with a semi-structured explanation. For evaluation, we use Answer Accuracy (AA), Triple Match F1 Score (T-F1), Graph Match F1 Score (G-F1), Graph-BertScore (G-BS), Graph Edit Distance (GED).

The detailed descriptions of all evaluation metrics are provided in Appendix A.

4.2 Models

LLM Baselines. To probe the capability of LLMs on generating semi-structured explanations, we conducted experiments on two advanced LLMs, ChatGPT (*gpt-3.5-turbo-instruct*) and GPT-4 (*gpt-4*). We performed 2-shot and 6-shot in-context learning. In addition to the standard prompting we also prompted the models by providing the list of relation types (giving LLM a higher chance of extracting the right relations) in ExplaGraph dataset. The full prompts used for these two tasks are shown in Appendix D.

SFT. For supervised fine-tuning (SFT), we conduct experiments on decoder-only architecture models, LLAMA2 (Touvron et al., 2023b), and encoder-decoder architecture models, FLAN-T5 (Chung et al., 2022). For LLAMA2, we use LLaMA2-7B and LLaMA2-13B, and for FLAN-T5, we use FLAN-T5-Large, FLAN-T5-XL and FLAN-T5-XXL. We perform instruction-tuning on the models using LoRA (Hu et al., 2022), which is a parameter-efficient fine-tuning method.

[3] We have submitted our prediction test set to evaluate and we will update the test evaluation result once we receive it.

		Answer	Explanation				
		SA↑	**StCA↑**	**SeCA↑**	**G-BS↑**	**GED↓**	**EA↑**
	RE-SP (Saha et al., 2021)	72.30	**62.30**	18.50	47.00	0.62	27.10
	T5-Large (Saha et al., 2022)	86.20	46.50	31.60	36.80	0.66	26.80
	T5-Large + CL (Saha et al., 2022)	86.20	52.70	37.90	41.70	0.62	29.80
	BART-Large (Cui et al., 2023)	88.19	36.43	26.13	28.42	0.74	20.77
	BART-Large + EG3P (Cui et al., 2023)	88.19	48.99	37.43	38.73	0.65	25.03
FS=2	ChatGPT (gpt-3.5-turbo-instruct)	76.63	7.79	2.76	6.23	0.95	3.90
FS=2	+ relation	73.62	20.85	4.27	16.17	0.86	10.89
FS=2	GPT-4 (gpt-4)	**95.73**	6.53	2.01	5.16	0.95	4.63
FS=2	+ relation	94.47	19.60	6.53	15.31	0.86	12.62
FS=6	ChatGPT (gpt-3.5-turbo-instruct)	78.89	11.56	3.76	9.09	0.92	5.77
FS=6	+ relation	79.65	21.11	4.32	16.66	0.86	11.13
FS=6	GPT-4 (gpt-4)	95.48	22.11	13.07	17.55	0.84	13.83
FS=6	+ relation	94.97	27.89	13.81	21.45	0.81	18.48
SFT	LLaMA2-7B	88.69	40.95	23.87	31.05	0.71	26.68
SFT	LLaMA2-13B	89.45	43.72	26.38	33.86	0.69	27.62
SFT	FLAN-T5-Large-780M ○	77.64	22.11	13.07	16.41	0.85	14.03
SFT	FLAN-T5-XL-3B ◇	90.45	38.19	27.63	29.39	0.73	26.42
SFT	FLAN-T5-XXL-11B ⋆	91.71	46.98	35.18	36.14	0.66	31.23
SFT + RL	○ + RL with only R_ϕ	77.39	22.11	13.07	18.09	0.84	15.40
SFT + RL	○ + RL with only R_m	78.39	21.36	13.57	16.33	0.84	14.40
SFT + RL	○ + RL with R_ϕ, R_m w/o weights	78.89	25.63	16.33	20.36	0.81	16.98
SFT + RL	○ + RL with R_ϕ, R_m with weights	79.40	24.87	15.08	20.12	0.82	17.00
SFT + RL	◇ + RL with only R_ϕ	90.45	49.25	36.18	38.92	0.64	34.67
SFT + RL	◇ + RL with only R_m	90.45	40.70	28.73	31.36	0.71	28.14
SFT + RL	◇ + RL with R_ϕ, R_m w/o weights	90.95	50.50	36.38	39.60	0.63	36.39
SFT + RL	◇ + RL with R_ϕ, R_m with weights	89.45	46.98	34.67	37.55	0.66	32.64
SFT + RL	⋆ + RL with only R_ϕ	91.46	57.54	44.47	44.83	0.59	39.38
SFT + RL	⋆ + RL with only R_m	91.96	59.55	46.73	47.28	0.57	38.61
SFT + RL	⋆ + RL with R_ϕ, R_m w/o weights	91.96	61.81	**48.49**	**47.50**	**0.56**	**44.16**
SFT + RL	⋆ + RL with R_ϕ, R_m with weights	91.46	56.03	42.46	44.25	0.60	38.67

Table 2: The evaluation results on ExplaGraph dev set. The α used in "with weights" is 0.9. **Bold** shows the best result for a column, and arrows indicate the direction of improvement, i.e., ↑: higher is better. Colors denote the best within each group of methods.

		Answer	Explanation			
		AA↑	**T-F1↑**	**G-F1↑**	**G-BS↑**	**GED↓**
FS=2	ChatGPT (gpt-3.5-turbo-instruct)	94.8	0.55	0.00	43.99	45.79
FS=2	GPT-4 (gpt-4)	**100.0**	1.29	0.00	59.97	34.89
FS=6	ChatGPT (gpt-3.5-turbo-instruct)	93.4	0.85	0.00	47.86	45.55
FS=6	GPT-4 (gpt-4)	99.8	2.19	0.00	62.41	31.36
SFT	LLaMA2-7B	60.8	1.21	8.20	63.97	19.93
SFT	LLaMA2-13B	83.8	1.39	8.40	65.40	19.85
SFT	FLAN-T5-Large-780M	88.0	0.93	5.91	65.67	20.05
SFT	FLAN-T5-XL-3B	95.4	1.73	8.39	**69.25**	20.00
SFT	FLAN-T5-XXL-11B ⋆	97.4	1.87	8.42	67.20	19.77
SFT + RL	⋆ + RL with only R_ϕ	98.0	2.01	11.71	67.93	18.65
SFT + RL	⋆ + RL with only R_m	97.2	1.93	10.85	67.50	19.02
SFT + RL	⋆ + RL with R_ϕ, R_m w/o weights	97.8	**2.33**	**12.47**	68.80	**17.49**
SFT + RL	⋆ + RL with R_ϕ, R_m with weights	97.2	2.05	10.87	67.68	18.75

Table 3: The evaluation results on COPA-SSE test set. The weight factor α used in last setting is 0.5. **Bold** shows the best result for a column, and arrows indicate the direction of improvement, i.e., ↑: higher is better. Colors denote the best within each group of methods.

RL. Previous work has shown that the encoder-decoder architecture models generally perform better than decoder-only architectures in transduction tasks that need a deep understanding of the input (Fu et al., 2023). This finding is in line with our results 4.3. Therefore, we only use FLAN-T5 models as our backbone models for RL. For reward modelling, since it does not need to perform the down-stream tasks directly, we use LLaMA-7B for simplicity. Inspired by the previous work (Touvron et al., 2023b), we first fine-tune the pre-trained LLaMA-7B on the task data, then the reward model is initialised from the fine-tuned LLaMA-7B model checkpoint. This can help the reward model to better understand the input and improve the performance. The training details are provided in the Appendix C.

Other Baselines. For ExplaGraph, all of these baselines fine-tune a RoBERTa model to predict the stance label by conditioning on the belief and argument. For explanation graph generation, RE-SP (Saha et al., 2021) combines different models to predict the internal nodes, external nodes and relations, respectively. T5-Large (Saha et al., 2022) and BART-Large (Cui et al., 2023) generate explanation graphs as post-hoc explanations by conditioning on the belief, argument and the predicted stance using T5-Large model and BART-Large model. T5-Large+CL (Saha et al., 2022) further implements contrastive learning methods on T5-Large. BART-Large+EG3P (Cui et al., 2023) incorporates a generative pre-training mechanism over synthetic graphs on BART-Large to improve the model's capability for SEG task. For COPA-SSE, since it is a relatively new dataset, there are no public baselines we can compare.

4.3 Results

ExplaGraph. We demonstrate the evaluation results on ExplaGraph in Table 2, comparing with other baseline methods. For SFT results, FLAN-T5-XXL performs better than LLaMA2-13B. As the model size increases, the performance also improves accordingly. Even only doing SFT on FLAN-T5-XXL can achieve higher SA and EA than all five baseline methods. For the RL results, when we only use single reward R_m or R_ϕ in RL, the performance is improved. The improvements are much more remarkable in FLAN-T5-XL and FLAN-T5-XXL. The metric reward we use is G-BERTScore (see §4.4 for ablation on the metrics) and the KL coefficient β is 0.3 (see §4.5 for ablation on coefficients) for RL, which are the best setting based on our experiments.

Using single metric reward R_m is more effective than using the reward model R_ϕ. The aggregation of R_ϕ and R_m without using weights on FLAN-T5-XXL has the best performance among all settings, which outperforms the baselines on five metrics by a large margin. Since we did not add any constraints on the structure of predicted graph comparing with the RE-SP (Saha et al., 2021) baseline method which explicitly enforces graph structure

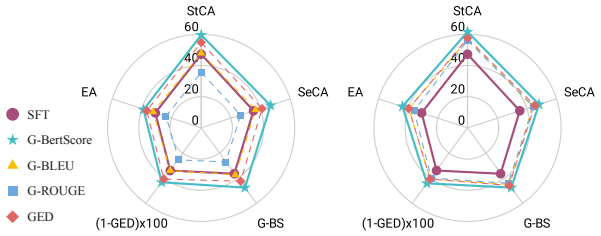
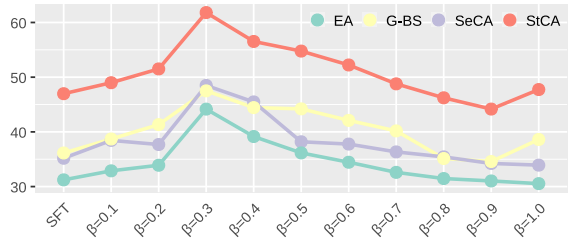


Figure 2: Comparison, on ExplaGraph, of SFT and various RL configurations to calculate R_m. The KL Coefficient β is 0.3 for all experiments. (left) RL using only reward metric, (right) RL using both reward model and metric without any weights.

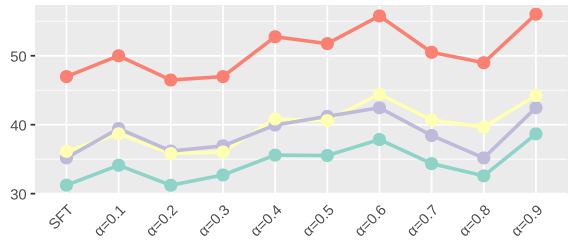
constraints (i.e., connectivity and acyclicity), this could explain why StCA is not the highest for our method. The aggregation of two rewards using weight performs even worse than using single reward. We speculate that using weight decreases the effect of two rewards, thus leading to an undesired influence to the RL.

COPA-SSE. The evaluation results on COPA-SSE is shown in Table 3. Using RL can steadily improve the performance of the SFT model, especially when conducting rewards aggregation without using weights. This is consistent with the result shown on ExplaGraph dataset.

Performance of LLMs. The GPT-4 performs far better than ChatGPT both in answer prediction and explanation generation, which reveals GPT-4 has a stronger reasoning ability than ChatGPT. Including the relation information (denoted as +relation) can greatly improve the performance in both models. Surprisingly, the stance accuracy on GPT-4 using few-shot learning has surpassed the SFT models. However, even using 6-shot learning on LLMs, the performance on SEG is still far behind the SFT models. For COPA-SSE task, GPT-4 even achieves 100% accuracy on answer prediction using 2-shot learning. However, when using 6-shot learning, the answer accuracy drops a little bit on both GPT-4 and ChatGPT models, although the quality of explanation increases. We speculate that adding more demonstrations introduces some extra information which may affect the model's judgement on answer prediction. G-F1 score is 0 on all settings, which means none of the generated semi-structured explanation matches exactly to the gold reference. This indicates the challenge of generating semi-structured explanation on LLMs and provides a direction for future research.



(a) Different β values.



(b) Different α values.

Figure 3: FLAN-T5-XXL - SFT in comparison (on ExplaGraph dev set) with SFT+RL under (a) different values of KL Coefficient β (we use the aggregation method without weights), and (b) different values of weight factor α (fixing $\beta = 0.3$).

4.4 Effect of Different Metrics in R_m

In Section 3.2, we introduced four metrics Graph-BLEU, Graph-ROUGE and Graph-BERTScore, and Graph Edit Distance which could be used to calculate R_m. To probe the effect of these metrics, we conduct probing experiments on ExplaGraph. The results are shown in Figure 2 (Full results provided in Table 9 of Appendix). Graph-BERTScore performs best among all metrics. We speculate this is because the BLEU and ROUGE are calculated using overlapping n-grams. Essentially for the graph-structured data containing multiple triples, the calculation of n-grams becomes less meaningful. However, Graph-BERTScore is a semantic evaluation metric which is still useful in graph-structured data, thus leading to better performance in R_m. Interestingly, GED - which considers the structure of the explanation - as a reward metric is not as effective as Graph-BERTScore. This echos the challenge of identifying sources of feedback for RLHF that align well with the underlying task specification (Casper et al., 2023).

4.5 Effect of β and α Coefficients

KL Coefficient β is a significant parameter controlling the deviation from the SFT model. To investigate the effect of β, we conduct experiments on ExplaGraph dataset using different values of β (from 0.1 to 1.0). The results are demonstrated in Figure 3a (See Table 7 in Appendix B for full

	Rank 1st	Rank 2nd	Rank 3rd	Avg. Rank
Gold	87	38	75	1.94
SFT	46	93	61	2.08
SFT+RL	67	69	64	1.98

Table 4: Human evaluation results on 100 ExplaGraph samples by 2 assessors (200 evaluations in total).

results). As the β increases from 0.1, the performance becomes better until β is over 0.3. From 0.3 to 1.0, the performance goes down gradually, although they achieve the highest SA. In general, setting β as 0.3 leads to the best performance in both ExplaGraph and COPA-SSE tasks. When β is small (e.g., 0.1) the new model deviates far from the old model. In this case, although there is a slight improvement, the model may also learn some undesired pattern to achieve higher rewards (i.e., reward hacking). As the β increases, it forces the new model to remain close to the old model, leading a steady improvement. When β is close to 1.0, the performance is almost identical to SFT.

Weight factor α in our reward aggregation method is used to control the importance of different rewards. Although using the reward aggregation method without weights (i.e., removing α and $1-\alpha$) performs better, here we investigate the effect of α (from 0.1 to 1.0). The results are shown in Figure 3b (See Table 8 in Appendix B for full results). From the results, there is no explicit pattern, but in general, larger values of α result in better performance. This means in reward aggregation, the reward from reward model R_ϕ is more significant than metric reward R_m. A dynamic adaptation of α depending on instances is an interesting direction to investigate in future.

5 Analysis

5.1 Human Evaluation

To further evaluate the quality of the generated output from SFT and SFT+RL models, we conduct a human evaluation on 100 randomly sampled instances from ExplaGraph which have correct stance predictions. For each instance, given a belief, an argument and its corresponding stance, we provide assessors with three explanation graphs: Gold reference, SFT, and SFT+RL output. For the evaluation process we recruited two annotators (with at least Master's degree in NLP). Assessors were instructed to rank the three explanation graphs without disclosing their sources, based on the quality of each graph. The human evaluation (total of 200 evaluations) results are demonstrated in Table 4. As expected, Gold reference ranks first most of the time, followed by SFT+RL output, then SFT output. Based on the average ranking, the SFT+RL output has a higher ranking than the SFT output and a small gap with the gold reference. This indicates that using RL can improve the quality of the generated semi-structured explanation graphs. To our surprise, gold reference has the highest third ranking. Since the ground-truth is created by human annotators, it is inevitably influenced by subjectivity[4]. This necessitates the human evaluation in addition to the automatic evaluation.

Triple Level Redundancy
Belief: Marriage offers numerous benefits. **Argument:** Marriage is just a piece of paper.
Output: counter (marriage; is a; piece of paper)(piece of paper; not capable of; numerous benefits)(piece of paper; not capable of; numerous benefits)
Concept Level Redundancy
Belief: Entrapment helps solve crimes. **Argument:** Entrapment violates liberties.
Output: counter (entrapment; capable of; violates liberties)(violates liberties; not capable of; helps solve crimes)(entrapment; synonym of; entrapment)

Table 5: Two types of redundancy errors in SFT+RL outputs. Errors are shown in red color text.

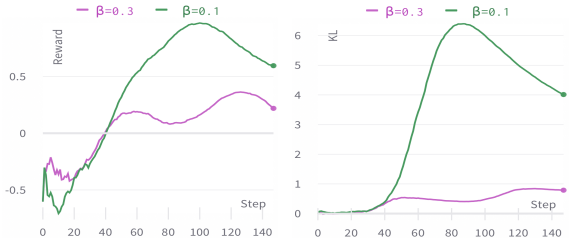
5.2 Qualitative Examples

In Figure 1 we demonstrate two examples from ExplaGraph. In the first example, SFT output fails to generate the relation between "*natural habitats*" and "*natural environments*", while SFT+RL output generate the relation "*PartOf*". This is important for connecting the belief with the argument in the explanation graph. In the second example, SFT+RL output generates a new concept "*cure disease*" which helps to better understand the function of "*stem cell research*". Additionally, it also increases the chances of generating external concepts even we do not explicitly force the model to do so (i.e., predict the internal and external concepts separately). See more examples in Appendix E.

5.3 Error Analysis

During the human evaluation process, we collected the errors in SFT+RL outputs. Specifically, there are two types of redundancy errors: Triple Level Redundancy and Concept Level Redundancy. We demonstrate an example of each type in Table 5. Triple Level Redundancy means the outputs con-

[4]Cohen's κ of our human evaluation result is 0.18±0.15 with confidence 95% indicating a slight agreement, which also underscores the subjectivity of the explanation task.



(a) Mean reward plot. (b) KL plot.

Figure 4: An illustration of the mean reward and the kl during RL training on ExplaGraph: (a) as the training continues, the rewards of both settings increase. While in (b) when β is 0.1, the large KL indicates significant deviation from the original SFT model, thus leading to a reward hacking phenomenon.

tain repetitive triples. Based on our observation, the repetitive triple is usually the last triple in the generated explanation graph. In the Triple Level Redundancy example in Table 5, the triple "*(piece of paper; not capable of; numerous benefits)*" is generated twice. Concept Level Redundancy means the outputs contain repetitive concepts. This type of error is usually associated with a specific relation "*synonym of*". In the Triple Level Redundancy example in Table 5, the triple "*(entrapment; synonym of; entrapment)*" contains the repetitive concept "*entrapment*". We speculate these undesired behaviours emerge during the policy optimization stage in RL. One general solution for these errors is to enhance robustness and generalization of the reward model (e.g., improve the quality of the preference paired data). In addition, one can also explicitly target redundancy in the RL phase (i.e., via metric design or direct penalty on the reward). It is worth noting that this might not be effective in practice due to the rarity of such patterns during the optimization phase. We leave further exploration of these to future.

5.4 Reward Hacking

Reward hacking (Skalse et al., 2022) is a phenomenon where a model achieves high rewards by optimizing a reward function but leading to a low evaluation score on the outputs. Previous work have shown that reward hacking could happen in RLHF training on LLMs (Peng et al., 2023; Touvron et al., 2023b). The second term in Eq. 1 is a constraint useful for training stability and mitigating the risk of reward hacking. We demonstrate a mean reward plot and a KL plot in Figure 4 to showcase that the RL training with small KL Coefficient β (i.e., 0.1) leads to reward hacking. We

Belief: Cosmetic surgery should be banned. **Argument:** Cosmetic surgery is not worth the risk.
Gold: support (cosmetic surgery; is a; risky)(risky; used for; human body)(human body; has property; precious)(precious; desires; banned)(banned; used for; risk)
SFT+RL ($\beta = 0.3$)**:** support (cosmetic surgery; has property; dangerous)(dangerous; desires; banned)(cosmetic surgery; has property; not worth the risk)
SFT+RL ($\beta = 0.1$) support (cosmetic surgery; is a; dangerous)(dangerous; desires; banned)(cosmetic surgery; is a; not worth the risk)(not worth the risk; desires; banned)(cosmetic surgery; synonym of; plastic surgery)(plastic surgery; synonym of; cosmetic surgery)

Table 6: An example from ExplaGraph dev set to show the output from the model which encounters reward hacking problem (SFT+RL $\beta = 0.1$).

demonstrates an example showing different outputs from these two settings in Table 6. Under $\beta = 0.1$, the model tends to generate longer texts with unnecessary information. It is worth mentioning that the choice of KL Coefficient depends on different tasks. As discussed earlier (§5.3), this stands out as one of the inherent challenges of RLHF application to this task, and choosing a proper KL Coefficient has a potential in addressing this to some degree.

Additionally, we observe the average number of triples for SFT and SFT+RLHF on ExplaGraph to be roughly the same (SFT: 3.0±0.56, SFT+RLHF: 3.1±0.33). This finding seems to differ from observations in a recent study on text generation (Singhal et al., 2023) which highlights that RLHF tends to generate much longer outputs compared to SFT. We speculate this observation could be an artefact of mild reward hacking, in which a longer sequence could collect further reward via redundancy.

6 Conclusion

In this work, we focused on the semi-structured explanation generation task and proposed to train a single model with SFT+RL to generate both answers and structured explanations. We highlighted the inadequacy of SFT in performing this complex task, and proposed a carefully designed reward engineering method in RL to better address this problem. We investigated different reward aggregation methods and conduct extensive experiments under different settings to better highlight the dynamic of the RL objective function and reward choices. Our method achieves the new SoTA results on two SEG benchmarks, ExplaGraph and COPA-SSE. We provide detailed analysis from different perspectives and hope these empirical findings will be beneficial for the future research on investigating RL in SEG.

Limitations

In this work, we only focused on the online alignment method (i.e., using PPO in RL), while there are other offline alignment approaches to align language models with preference data, like DPO (Rafailov et al., 2023), PRO (Song et al., 2023), RRHF (Yuan et al., 2023). It is also worth investigating the performance of these methods on SEG tasks.

Ethics Statement

Our work uses the existing open-source pre-trained models, as such it could inherit the same ethical concerns which has been widely discussed in the community. We uses the public available datasets which is broadly accepted by the community. The created training data from COPA-SSE did not generate any new data, which also do not have the ethical issues.

References

Zeina Abu-Aisheh, Romain Raveaux, Jean-Yves Ramel, and Patrick Martineau. 2015. An exact graph edit distance algorithm for solving pattern recognition problems. In *ICPRAM 2015 - Proceedings of the International Conference on Pattern Recognition Applications and Methods, Volume 1, Lisbon, Portugal, 10-12 January, 2015*, pages 271–278. SciTePress.

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for commonsenseqa: New dataset and models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3050–3065. Association for Computational Linguistics.

Ana Brassard, Benjamin Heinzerling, Pride Kavumba, and Kentaro Inui. 2022. COPA-SSE: semi-structured explanations for commonsense reasoning. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 3994–4000. European Language Resources Association.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, John A. Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuan-Fang Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *ArXiv*, abs/2303.12712.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9560–9572.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Ségerie, Micah Carroll, Andi Peng, Phillip J. K. Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Biyik, Anca D. Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *CoRR*, abs/2307.15217.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Han Cui, Shangzhan Li, Yu Zhang, and Qi Shi. 2023. Explanation graph generation via generative pre-training over synthetic graphs. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9916–9934. Association for Computational Linguistics.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022. 8-bit optimizers via block-wise quantization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4443–4458. Association for Computational Linguistics.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy

Liang, and Tatsunori B. Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *CoRR*, abs/2305.14387.

Zihao Fu, Wai Lam, Qian Yu, Anthony Man-Cho So, Shengding Hu, Zhiyuan Liu, and Nigel Collier. 2023. Decoder-only or encoder-decoder? interpreting language model as a regularized encoder-decoder. *CoRR*, abs/2304.04052.

Jiuzhou Han, Nigel Collier, Wray L. Buntine, and Ehsan Shareghi. 2023. Pive: Prompting with iterative verification improving graph-based generative capability of llms. *CoRR*, abs/2305.12392.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Peter A. Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton T. Morrison. 2018. Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Harsh Jhamtani and Peter Clark. 2020. Learning to explain: Datasets and models for identifying valid reasoning chains in multihop question-answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 137–150. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *NeurIPS*.

Qing Li, Qingyi Tao, Shafiq R. Joty, Jianfei Cai, and Jiebo Luo. 2018. VQA-E: explaining, elaborating, and enhancing your answers for visual questions. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, volume 11211 of *Lecture Notes in Computer Science*, pages 570–586. Springer.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2829–2839. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics*.

Hugo Liu and Push Singh. 2004. Conceptnet — a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22:211–226.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Baolin Peng, Linfeng Song, Ye Tian, Lifeng Jin, Haitao Mi, and Dong Yu. 2023. Stabilizing RLHF through advantage model and selective rehearsal. *CoRR*, abs/2309.10202.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *CoRR*, abs/2305.18290.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7:249–266.

Swarnadeep Saha, Prateek Yadav, and Mohit Bansal. 2022. Explanation graph generation via pre-trained language models: An empirical study with contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin,*

Ireland, May 22-27, 2022, pages 1190–1208. Association for Computational Linguistics.

Swarnadeep Saha, Prateek Yadav, Lisa Bauer, and Mohit Bansal. 2021. Explagraphs: An explanation graph generation task for structured commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7716–7740. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4603–4611. PMLR.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2023. A long way to go: Investigating length correlations in rlhf.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward hacking. *CoRR*, abs/2209.13085.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2023. Preference ranking optimization for human alignment. *CoRR*, abs/2306.17492.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Sarah Wiegreffe and Ana Marasovic. 2021. Teach me to explain: A review of datasets for explainable natural language processing. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Zhengnan Xie, Sebastian Thiem, Jaycie Martin, Elizabeth Wainwright, Steven Marmorstein, and Peter A. Jansen. 2020. Worldtree V2: A corpus of science-domain structured explanations and inference patterns supporting multi-hop inference. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 5456–5473. European Language Resources Association.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 2013–2018. The Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. RRHF: rank responses to align language models with human feedback without tears. *CoRR*, abs/2304.05302.

Hongming Zhang, Xinran Zhao, and Yangqiu Song. 2020. Winowhy: A deep diagnosis of essential

commonsense knowledge for answering winograd schema challenge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5736–5745. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675.

Appendix

A Evaluation Metrics

Stance Accuracy (SA) measures the stance prediction accuracy which ensures that the explanation graph is consistent with the predicted stance.

Structural Correctness Accuracy of Graphs (StCA) requires satisfying all the constraints defined for the task, which include the graph be connected DAG with at least three edges and having at least two exactly matching concepts from the belief and two from the argument.

Semantic Correctness Accuracy of Graphs (SeCA) requires all edges to be semantically coherent and given the belief, the unambiguously inferred stance from the graph matches the original stance.

Graph-BertScore (G-BS) considers graphs as a set of edges and solve a matching problem that finds the best assignment between the edges in the gold graph and those in the predicted graph. Each edge is treated as a sentence and the scoring function between a pair of gold and predicted edges is given by BERTScore. Given the best assignment and the overall matching score, compute precision, recall and report F1 as the G-BERTScore metric.

Graph Edit Distance (GED) measures the number of edit operations (addition, deletion, and replacement of nodes and edges) for transforming the predicted graph to a graph isomorphic to the gold graph. The cost of each edit operation is chosen to be 1. The GED for each sample is normalized between 0 and 1 by an appropriate normalizing constant (upper bound of GED). Lower GED indicates that the predicted graphs match more closely with the gold graphs.

Edge Accuracy (EA) computes the macro-average of important edges in the predicted graphs. An edge is defined as important if not having it as part of the graph causes a decrease in the model's confidence for the target stance.

Answer Accuracy (AA) calculates the answer prediction accuracy.

Triple Match F1 Score (T-F1) calculates F1 score based on the precision-recall between the triples in the generated graph and the ground-truth.

Graph Match F1 Score (G-F1) focuses on the entirety of the graph and evaluates how many graphs are exactly produced the same.

	Answer	Explanation				
	SA↑	**StCA↑**	**SeCA↑**	**G-BS↑**	**GED↓**	**EA↑**
FLAN-T5-XXL - SFT	91.71	46.98	35.18	36.14	0.66	31.23
+ RL, $\beta = 0.1$	91.46	48.99	38.44	38.70	0.65	32.88
+ RL, $\beta = 0.2$	91.71	51.51	37.69	41.33	0.64	33.90
+ RL, $\beta = 0.3$	91.96	**61.81**	**48.49**	**47.50**	**0.56**	**44.16**
+ RL, $\beta = 0.4$	**92.21**	56.53	45.48	44.44	0.59	39.15
+ RL, $\beta = 0.5$	**92.21**	54.77	38.19	44.21	0.61	36.16
+ RL, $\beta = 0.6$	**92.21**	52.23	37.77	42.10	0.63	34.45
+ RL, $\beta = 0.7$	**92.21**	48.78	36.34	40.18	0.65	32.60
+ RL, $\beta = 0.8$	**92.21**	46.23	35.43	35.13	0.67	31.47
+ RL, $\beta = 0.9$	**92.21**	44.17	34.23	34.58	0.67	31.03
+ RL, $\beta = 1.0$	**92.21**	47.74	33.92	38.61	0.66	30.54

Table 7: The full evaluation results on ExplaGraph dev set using different values of KL Coefficient β. For the reward aggregation in RL, we use the aggregation method without weights.

	Answer	Explanation				
	SA↑	**StCA↑**	**SeCA↑**	**G-BS↑**	**GED↓**	**EA↑**
FLAN-T5-XXL - SFT	91.71	46.98	35.18	36.14	0.66	31.23
+ RL, $\alpha = 0.1$	91.96	50.00	39.45	38.68	0.64	34.12
+ RL, $\alpha = 0.2$	92.46	46.48	36.18	35.82	0.67	31.22
+ RL, $\alpha = 0.3$	**92.21**	46.98	36.93	36.04	0.66	32.71
+ RL, $\alpha = 0.4$	91.71	52.76	39.95	40.83	0.62	35.59
+ RL, $\alpha = 0.5$	91.46	51.76	41.21	40.59	0.63	35.53
+ RL, $\alpha = 0.6$	91.71	55.78	**42.46**	**44.43**	**0.60**	37.85
+ RL, $\alpha = 0.7$	91.46	50.50	38.44	40.69	0.64	34.36
+ RL, $\alpha = 0.8$	91.71	48.99	35.18	39.65	0.65	32.58
+ RL, $\alpha = 0.9$	91.46	**56.03**	**42.46**	44.25	**0.60**	**38.67**

Table 8: The full evaluation results on ExplaGraph dev set using different values of weight factor α. The KL Coefficient β used is 0.3 for all experiments.

B Full Results

Table 7 and Table 8 demonstrate the full results of experiments on ExplaGraph using different values of KL Coefficient β and weight factor α.

C Training Details

All models are implemented using Pytorch (Paszke et al., 2019) and Transformers (Wolf et al., 2020). We use Adam (Kingma and Ba, 2015) and Adafactor optimizer (Shazeer and Stern, 2018). For the implementation of parameter efficient training method used in FLAN-T5-XXL and LLaMA-7B, we use

	Answer	Explanation				
	SA↑	**StCA↑**	**SeCA↑**	**G-BS↑**	**GED↓**	**EA↑**
FLAN-T5-XXL - SFT	91.71	46.98	35.18	36.14	0.66	31.23
+ RL with only R_m (G-BS)	91.96	59.55	46.73	47.28	0.57	38.61
+ RL with only R_m (G-BL)	91.71	47.99	36.93	36.91	0.66	32.65
+ RL with only R_m (G-RO)	**92.46**	35.43	26.38	26.70	0.75	23.87
+ RL with only R_m (GED)	91.96	54.77	40.95	42.52	0.59	36.52
+ RL with R_ϕ, R_m (G-BS)	91.96	**61.81**	**48.49**	**47.50**	**0.56**	**44.16**
+ RL with R_ϕ, R_m (G-BL)	91.96	57.04	44.22	45.20	0.59	39.54
+ RL with R_ϕ, R_m (G-RO)	91.96	56.03	44.47	44.30	0.60	35.99
+ RL with R_ϕ, R_m (GED)	92.21	57.54	45.47	45.63	0.59	39.32

Table 9: The evaluation results on ExplaGraph dev set under various metrics to calculate R_m. We use the aggregation method without weights. The KL Coefficient β is 0.3 for all experiments.

Hyperparameter	Assignment
Model	FLAN-T5-XXL
Epoch	5
Batch Size	16
Optimizer	adamw_torch
Learning Rate	3×10^{-4}
Warm-up Step	50
Beam Size	4
Lora-r	4
Lora-alpha	16
Lora-dropout	0.05
Lora-modules	[q, v]

Table 10: Hyperparameters of SFT Model

PEFT (Mangrulkar et al., 2022) and 8-bit quantization technique (Dettmers et al., 2022). All training was done using a single A40 GPU with 48GB of RAM. Table 10, Table 11 and Table 12 show the hyperparameters for SFT Model, Reward Model and RL model, respectively.

D Prompts used for ChatGPT and GPT-4

For ExplaGraph task, we use the prompt "*Given a belief and an argument, infer the stance (support/counter) and generate the corresponding commonsense explanation graph that explains the inferred stance.*" followed by a few demonstrations. For including relation setting, we use the the prompt "*Given a belief and an argument, infer the stance (support/counter) and generate the corresponding commonsense explanation graph that explains the inferred stance. The available relations in explanation graph are antonym of, synonym of, at location, not at location, capable of, not capable of, causes, not causes, created by, not created by, is a, is not a, desires, not desires, has subevent, not has subevent, part of, not part of, has context, not has context, has property, not has property, made of, not made of, receives action, not receives action, used for, not used for.*" followed by a few demonstrations.

For COPA-SSE task, we only use the prompt "*Given the premise, choose from a or b and generate an commonsense explanation graph that explains the answer.*" followed by a few demonstrations.

Hyperparameter	Assignment
Model	LLAMA-7B
Epoch	5
Batch Size	16
Optimizer	adamw_torch
Learning Rate	3×10^{-4}
Warm-up Step	50
Beam Size	4
Lora-r	8
Lora-alpha	16
Lora-dropout	0.05
Lora-modules	[q, v]

Table 11: Hyperparameters of Reward Model

Hyperparameter	Assignment
Model	FLAN-T5-XXL
PPO Epoch	3
Batch Size	16
Optimizer	adafactor
Learning Rate	1.4×10^{-5}
Warm-up Step	50
Beam Size	4
Lora-r	8
Lora-alpha	16
Lora-dropout	0.05
Lora-modules	[q, v]
Target-KL	2
KL-coef	0.3

Table 12: Hyperparameters of RL Model

E More Qualitative Examples

In Table 13, we demonstrate two examples from ExplaGraph. In the first example, SFT output fails to generate the concept "*create people*", while the SFT+RL output is much more complete with regard to an explanation graph given the belief and argument. In the second example, even both of the SFT and SFT+RL outputs can correctly generate the first triple "*(austerity programs; capable of; cut funding)*", SFT+RL output contains the concept "*negative effects*", which is similar to the concept

Belief: Human cloning should be allowed, as it would be a great boon for medical advancements. **Argument:** It is immoral to create people for the sole purpose of curing others.
Gold: counter (human cloning; used for; create people)(create people; used for; body parts only)(body parts only; has context; immoral)(immoral; not desires; allowed) **SFT:** counter (human cloning; capable of; immoral)(immoral; not desires; allowed)(immoral; used for; curing others) **SFT+RL:** counter (human cloning; capable of; immoral)(immoral; not capable of; allowed)(human cloning; capable of; create people)(create people; capable of; curing others)
Belief: Austerity programs are terrible for the economy. **Argument:** Austerity programs cut funding.
Gold: support (austerity programs; capable of; cut funding)(cut funding; capable of; hurts business)(hurts business; causes; terrible)(terrible; has context; for economy) **SFT:** support (austerity programs; capable of; cut funding)(cut funding; capable of; bad for economy)(bad for economy; synonym of; terrible) **SFT+RL:** support (austerity programs; capable of; cut funding)(cut funding; capable of; negative effects)(negative effects; capable of; terrible for the economy)

Table 13: Two examples from ExplaGraph dev set to compare the gold explanation graph with the SFT output and SFT+RL output.

"*hurts business*" in the gold. In general, using RL can make the generated explanation graph more detailed and complete than only using SFT. Additionally, it also increases the chances of generating external concepts even we do not explicitly force the model to do so (i.e., predict the internal and external concepts separately).