Weak Reward Model Transforms Generative Models into Robust Causal Event Extraction Systems

Italo Luis da Silva[1] **Hanqi Yan**[1] **Lin Gui**[1] **Yulan He**[1,2]
[1]King's College London [2]The Alan Turing Institute
`{italo.da_silva,hanqi.yan,lin.1.gui,yulan.he}@kcl.ac.uk`

Abstract

The inherent ambiguity of cause and effect boundaries poses a challenge in evaluating causal event extraction tasks. Traditional metrics like Exact Match and BertScore poorly reflect model performance, so we trained evaluation models to approximate human evaluation, achieving high agreement. We used them to perform Reinforcement Learning with extraction models to align them with human preference, prioritising semantic understanding. We successfully explored our approach through multiple datasets, including transferring an evaluator trained on one dataset to another as a way to decrease the reliance on human-annotated data. In that vein, we also propose a weak-to-strong supervision method that uses a fraction of the annotated data to train an evaluation model while still achieving high performance in training an RL model.[1]

1 Introduction

Causal event extraction is a crucial task in natural language understanding. It involves identifying cause and effect clauses within an event and the relationship between them. An example text along with its causal event annotations from the Fine-grained Causal Reasoning (FCR) dataset (Yang et al., 2022) is shown in Figure 1. The emergence of powerful generative models leads to a shift from span-based *extraction* to the *generation* of structured information (Guo et al., 2023; Sainz et al., 2023). However, recent studies suggest that ChatGPT (OpenAI, 2023) struggles to surpass smaller supervised models (Han et al., 2023), even when augmented with Chain-of-Thought (CoT) (Wei et al., 2022b) and few-shot In-Context Learning (ICL) (Brown et al., 2020).

We focus on fine-tuning smaller language models using text annotated with causal and effect spans for causal event extraction. However, we observe that unlike traditional named entity recognition, where entities have clear and often unambiguous boundaries, cause or effect spans may include intermittent text and could have blurred word boundaries. This means that even with minor word omissions, the semantic meaning of the cause and effect spans remains the same. Consequently, the same text could have multiple valid annotations. Therefore, training supervised models based on strictly matching only one set of valid human annotations may result in less robust models.

Source Text

The firm's gross margin is set to stabilize as Harley refocuses its efforts on more profitable markets, and our base case assumes that it stabilizes around 32% in 2029, helped by a more measured approach to entering new markets.

Gold Extraction

Cause: *Harley refocuses its efforts on more profitable markets*

Effect: *The firm's gross margin is set to stabilize*

Relation: *cause*

Figure 1: Example instance from the Fine-grained Causal Reasoning (FCR) dataset.

Evaluating causal event extraction is not straightforward. Evaluation metrics based on direct token-level overlapping tend to neglect semantically valid variations. Recent studies show that they do not align well with human evaluations (Han et al., 2023). This issue could be exaggerated under the generative settings (Qi et al., 2023). While Large Language Models (LLMs) are considered an alternative in evaluating the generation tasks due to their flexibility and ability to capture high-level semantics, discrepancies still exist between GPT-3.5 evaluation outputs and human evaluations, so human evaluators remain crucial to provide reliable feedback (Min et al., 2021), despite the high cost.

To address the high expense of human evaluation, we explore training evaluators for causal event extraction to account for semantic variations. We sample event extraction results from GPT-3.5 and

---

[1] Our code is available at `https://github.com/oyarsa/event_extraction/tree/causal-event-extraction`.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 14707–14719
November 12-16, 2024 ©2024 Association for Computational Linguistics

a fine-tuned FLAN-T5 (Chung et al., 2022) model, inviting human annotators to label the correctness of these extractions as 'valid' or 'invalid'. These human evaluation results are then used to train an evaluator. Our experiments demonstrate that an evaluator trained on a subset of human evaluations from one dataset can be transferred to other datasets without losing alignment with the actual human evaluation results.

Furthermore, we propose using the evaluator as a reward model to fine-tune the causal event extraction model, FLAN-T5, through reinforcement learning instead of traditional cross-entropy loss to prioritise semantic similarity over exact matching. The Policy Proximal Optimisation (PPO) (Schulman et al., 2017) algorithm is used to align generative models' behaviours with human preferences. In this method, a reward model is first trained on human preference data and is used to produce feedback scores, guiding the policy model to reinforce high scoring and penalise low-scoring generations.

In this paper, we incorporate the trained evaluator as the reward model into PPO for causal event extraction. Our contributions are threefold:

- We built a causal relation extraction platform to collect human evaluation data, which is then used to train an evaluator (i.e. a reward model). It shows a 0.94 correlation with human evaluations.
- The reward model is integrated into the PPO algorithm for fine-tuning a FLAN-T5 model for causal event extraction. It achieves an average improvement of 4% across three datasets.
- To decrease the reliance on human evaluations and ground-truth references, we propose a weak-to-strong framework to fully exploit data efficiency of our proposed approach. We succeeded in using 50% of the supervised data augmented by weak supervision with dynamic filtering as a reward model for RL training, obtaining comparable performance with the full reward model.

2 Related Work

We will introduce the recent work in causal extraction tasks, reward models for reinforcement learning, weakly-supervised reward models and data augmentation for generative models.

2.1 Causal event extraction

The goal of causal event extraction is to identify and extract cause and effect events from an input text. Prior works focus on identifying relations between entities, often trigger words (Huguet Cabot and Navigli, 2021; Chen et al., 2020; Ma et al., 2022). The works that focused on relations between events focus exclusively on simple causal (Mirza and Tonelli, 2016; Mariko et al., 2020) relations, with no fine-grained relations considered.

Existing works employed span-based extraction (Becquin, 2020) and sequence tagging (Saha et al., 2022), but they are limited to single cause and effect scenarios, with simple relations. However, the recent increase in generative models, such as T5 (Raffel et al., 2020), GPT-3.5 and GPT-4 (OpenAI, 2023) highlight another possibility. They have shown the outstanding generalisation to not only learn from IE training data through fine-tuning (Paolini et al., 2021), but also extract information in few-shot and even zero-shot scenarios relying solely on in-context examples or instructions (Wei et al., 2022a; Wang et al., 2022a). However, other works (Nasar et al., 2021; Zhou et al., 2022) have shown deficiencies in scenarios where there is a shortage of training data, an area that has not been fully explored.

Traditional metrics such as exact match (EM) and token F1 rely on the idea that a correct extraction is one that completely matches the annotation. There are other automated metrics such as ROUGE (Lin, 2004), BLEU (Papineni et al., 2001), BLEURT (Sellam et al., 2020) and BERTScore (Zhang et al., 2020) that attempt to solve this problem, but we found them to not correlate well with human annotations. Our solution was to train our own evaluation models so that they correspond well with human evaluation. (Section 3).

2.2 Reward model in generative model

Reinforcement Learning through Human Feedback (PPO) (Ouyang et al., 2022) has seen applications for instruction tuning (Shu et al., 2023; Lai et al., 2023), controlled text generation (Castricato et al., 2022; Shulev and Sima'an, 2024), summarisation (Roit et al., 2023) and other generative tasks (Cetina et al., 2021; Pang et al., 2023). However, to the best of our knowledge, it has not been applied to causal event extraction as a mechanism to combat the limitations of automated metrics. Feedback acquisition is one of the significant components, where humans or reward models assess the quality of the base model's responses to serve as a supervision signal for generative models.

A critical aspect of this paradigm is to accurately model human preferences, which involves the costly and time-consuming process of collecting feedback data. Therefore, many recent works focus on how to fully steer the capabilities of generative models with minimum supervision (Yu et al., 2020; Otani et al., 2022).

Several methods have improved LLMs by (self-) creating training data to augment fine-tuning. Self-Instruct (Wang et al., 2022b) is a method for self-instruction creation of prompts and responses, which can be used to improve a base LLM. Several approaches have also created training data by distilling from powerful LLMs, and shown a weaker LLM can then perform well. For example, Alpaca (Taori et al., 2023) fine-tuned a Llama 7B model with text-davinci-003 instructions created in the style of self-instruct. Alpagasus (Chen et al., 2024) employed a strong LLM-as-a-Judge (ChatGPT) to curate the Alpaca dataset and filter to a smaller set, obtaining improved results.

3 Approximating Human Evaluation

Automated metrics for the evaluation of generated text have limitations in aligning with human evaluation. Metrics such as F1 score can measure the overlap between the gold standard extraction and model outputs, but fail to recognise the semantic aspects of such comparisons. In causal event extraction, we often have situations where the output is different and has incomplete overlap with the gold standard but is nonetheless correct. Automated metrics are unable to deal with these situations since they cannot account for semantic differences, such as when adding or removing words does not change the meaning of an extraction.

One way to circumvent this issue is to employ human annotators to evaluate model outputs. While effective, it is expensive and time-consuming, severely limiting experimentation and the development of new approaches.

To address these limitations, we propose to collect human feedback to train an evaluation model for high-quality feedback generation. The goal is to have an automated way to evaluate model outputs that approximates the judgement a human would have made without the time-consuming and expensive aspects of human evaluation.

3.1 Human Feedback Collection

Platform setup. We built a platform to collect human annotations for causal-effect extraction tasks. For each sample, annotators are given the *Source Text*, *Cause* and *Effect*. For both *Cause* and *Effect*, we provide the *Reference* and *Model Output*. Annotators are asked to make a binary decision on whether the *Model Output* is a valid extraction for the given source text, with a sample only being valid if both *Cause* and *Effect* are correct. See Section E (Appendix) for more details.

To enhance the generalisability of the annotation data, we first apply two different generative models, FLAN-T5 and GPT-3.5, to generate the cause and effect results for evaluation. We remove instances where the generated outputs are exact matches with the reference, as those cases are trivial to evaluate. The remaining generated outputs are organised using our tagged template. Figure 2 shows an example instance from the FinCausal (Mariko et al., 2020) dataset, including the *Source Text*, the *Cause* and *Effect* spans, and the equivalent version in our tagged format.

Source Text

It found that total U.S. health care spending would be about $3.9 trillion under Medicare for All in 2019, compared with about $3.8 trillion under the status quo. Part of the reason is that Medicare for All would offer generous benefits with no copays and deductibles, except limited cost-sharing for certain medications.

Gold Extraction

Cause: *Part of the reason is that Medicare for All would offer generous benefits with no copays and deductibles, except limited cost-sharing for certain medications.*

Effect: *It found that total U.S. health care spending would be about $3.9 trillion under Medicare for All in 2019, compared with about $3.8 trillion under the status quo.*

Structured output (tagged format)

[Cause] *Part of the reason is that Medicare for All would offer generous benefits with no copays and deductibles, except limited cost-sharing for certain medications.* **[Relation]** ***cause*** [Effect] *It found that total U.S. health care spending would be about $3.9 trillion under Medicare for All in 2019, compared with about $3.8 trillion under the status quo.*

Figure 2: An example instance from the FinCausal dataset. Top to Bottom: Source text in original dataset, Gold Standard Extraction, Structured output.

The gold standard extractions for cause and effect are formatted into the same structured output. Finally, both the formatted model output and the reference, along with the *Source Text*, are presented to the annotators (shown in the following **Examples**). Our instructions for annotators primarily address the shortcomings of the current evaluation methods. We identify the two most common issues: *Wording Variation* and *Hallucination*.

Pitfalls of Existing Evaluation Schema. Three representative cases are shown below. GPT-3.5 was used as an evaluator. In both cases, GPT-3.5's evaluation results differ from those of human evaluators. The evaluator errors for *Wording Variation* always occur in the span border, either adding some tokens or removing some tokens. The *Hallucination* issue happens when the generative model copies the text correctly but generates incorrect numbers and symbols. *Swapped Events* happen when the generative model identifies the correct events in the text, but swaps the cause and effect. These examples illustrate how even a competent model struggles to reproduce human responses, motivating the need for a specialised evaluation method.

Table 1 shows the incidence of these types of errors in a random samples of 100 instances for both GPT-3.5 and our trained evaluator model, demonstrating how our evaluator is more robust.

Example 1

Source Text: Our near-term earnings forecast is depressed due to the incorporation of crack spread futures curves despite a recent uptick.

Reference: [Cause] the incorporation of crack spread futures curves **despite a recent uptick** [Relation] cause [Effect] Our near-term earnings forecast is depressed.

Output: [Cause] the incorporation of poor crack spread futures curves [Relation] cause [Effect] Our near-term earnings forecast is depressed.

Evaluator: Invalid **Human:** Valid

Example 2

Source Text: Analyst Ratings This is a breakdown of recent ratings and recommmendations for Auris Medical and Elite Pharmaceuticals, as provided by MarketBeat.com. Auris Medical currently has a consensus price target of $75.00, indicating a potential upside of 2,383.44

Reference: [Cause] Auris Medical currently has a consensus price target of $75.00 [Relation] cause [Effect] a potential upside of 2,383.44%.

Output:[Cause] Auris Medical currently has a consensus price target of $9.50 [Relation] cause [Effect] a potential upside of 655.21%
Evaluator: Valid **Human:** Invalid

Example 3

Source Text: CBRE's preeminent competitive position derives in large part from its lengthy track record of effective and ethical service, which has bolstered its reputation..

Reference: [Cause] its lengthy track record of effective and ethical service [Relation] cause [Effect] CBRE's preeminent competitive position.

Output: [Cause] CBRE's preeminent competitive position [Relation] cause [Effect] its lengthy track record of effective and ethical service **Evaluator**: Valid **Human:** Invalid

Error type	GPT-3.5	Our evaluator
Incorrect evaluation	54%	37%
- Word mismatch	28%	33%
- Hallucination	7%	0%
- Swapped Events	19%	4%
Correct evaluation	46%	63%

Table 1: Comparison of error types between GPT-3.5 and our trained evaluation model.

Instructions For Human Annotators. To alleviate the issues observed in the existing evaluation methods, we establish criteria for annotators. Only entries where both *Cause* and *Effect* satisfy all conditions should be considered as valid.

- Wording may differ between *Reference* and *Model Output*. This is fine, as long as the Model tokens come from the source text.
- There are no significant discrepancy between *Model Output* and *Reference*, such as numbers, subjects, time.
- If *Cause* and *Effect* happened to be in the same sentence but not overlapping, make sure the tokens in *Cause* are not included in the *Effect* and vice versa.
- In the rare cases where the *Reference* is obviously incorrect, ignore it and analyse the *Model Output* with relation to the source text only.

3.2 Alignment with Human Feedback

We conducted human evaluation on the extraction results from GPT-3.5 (10-shot) and FLAN-T5 on the training sets of the three datasets: FCR (Yang et al., 2022), FinCausal (Mariko et al., 2020) and SCITE (Li et al., 2021).[2] The Cohen's Kappa is 0.75, 0.51 and 0.84 for FCR, FinCausal and SCITE, respectively, showing a good level of agreement between annotators on all datasets.

We use the extraction results from GPT-3.5 (10-shot) and FLAN-T5 to train evaluation models by obtaining human evaluations for the outputs of the training sets for FCR and FinCausal. These human-evaluated outputs were then used to train the evaluation models, while the development set outputs were used to evaluate their performance[3], with the guiding metric being agreement between evaluator outputs and the human annotation (Zheng et al., 2023). Our goal is for these trained evaluators to approximate human judgement so we can use them as proxies for human evaluation in our experiments.

Our evaluation model is the DeBERTa-v3-based (He et al., 2022) classifier, specifically the `xsmall` variant, which we call DeBERTa-Valid. It

---

[2] Dataset statistics are shown in Table 5.

[3] We did not train an evaluator on SCITE because the number of training samples is too small.

takes both the source text and the gold standard extraction as inputs, along with the model output, to produce a classification. It is a binary classifier, with the positive class referring to 'valid' examples and the negative class to 'invalid'. We also explore variations of the DeBERTa classifier:

- DeBERTa-Entailment: an instance is considered correct if there is an entailment between the extracted output and the original source text. Its inferior performance shows its inefficiency in evaluating the generated cause/effects.
- DeBERTa-Valid variants: one variant excludes the reference extraction, and another excludes the source text. The poor performance of the variant without the reference shows its importance to our evaluator. Notably, the version without the source text also shows decreased agreement, indicating that the evaluator still needs it, as the references are not always reliable.

In addition, we use GPT-3.5 with or without self-consistency as additional automated evaluators for the causal event extraction task. To verify the effectiveness of our trained evaluator models, we calculate the agreement between our evaluator outputs and human evaluations on the development set, along with categorical metrics such as Exact Match in Table 2. We also examine the correlation between continuous metrics commonly used to evaluate extraction results, such as F1 and BertScore, and human evaluations. Pearson correlation results are shown in Tables 3. In both tables, we observe the low scores of existing automated metrics, highlighting their inability to replicate human evaluations. In contrast, our trained DeBERTa-based model achives higher agreement and correlation scores.

The results lead to the following observations: (a) automatic metrics do not align well with human evaluation. (b) LLMs demonstrate similar results to SentenceTransformer (Reimers and Gurevych, 2019) (SentTF), even with advanced prompting techniques, such as CoT and Self-Consistency (Wang et al., 2022a). (c) Supervised classification models (DeBERTa-*) perform the best. The inclusion of the reference is particularly crucial, which allows the reward model to achieve near-complete agreement with human evaluation.

We use DeBERTa-Valid, the best-performing model, as our proxy for human evaluation and the primary reward model in the following sections.

Metric	**T5**	**GPT-3.5 (10-shot)**
Exact Match	55.60	72.04
GPT-3.5	64.85	35.88
GPT-3.5-self-consistency	85.58	77.92
DeBERTa-entailment	68.61	43.19
DeBERTa-Valid-w/o-Reference	65.03	35.98
DeBERTa-Valid-w/o-SourceText	92.51	82.47
DeBERTa-Valid	**94.08**	**86.26**

Table 2: Agreement between human annotations and different metrics/evaluators on FCR (continuous metrics omitted). Various metrics are used to evaluate causal event extraction results from T5 and GPT-3.5 (10-shot)

Metric	**T5**	**GPT-3.5 (10-shot)**
ROUGE-L	80.94	67.15
BLEU	76.73	66.46
BLEURT	77.93	68.63
BertScore	75.94	65.83
F1	80.61	65.64
SentTF	63.70	47.53
DeBERTa-Valid	**87.04**	**72.98**

Table 3: Pearson correlation between human evaluations and different metrics/evaluators on FCR.

Transfer to other datasets. While using human evaluation to train an evaluation model leads to high-performing evaluators, this approach can be costly, especially for large datasets. We propose an alternative: train an evaluation model in one dataset and transfer it to others with similar structure. This is supported by the agreement between different combinations of evaluators and datasets, as shown in Table 4. We observe high agreement between the FCR evaluator and the transferred datasets' human evaluations, demonstrating the evaluator's transferability. As a result, we use the FCR evaluator as the default reward model and the evaluator for causal event extraction in our experiments.

Source $i \rightarrow$	**Target j**		
	FCR	FinCausal	SCITE
FCR	**94.08**	**92.04**	**96.86**
FinCausal	73.57	91.58	88.48

Table 4: Agreement between the feedback generated by the reward model trained on dataset i and human evaluation, when applying this model to generate feedback for dataset j^4.

4 Causal Event Extraction with Weak Reward Model

In this section, we introduce our Reinforcement Learning (RL) framework designed to align our generative extractor with human preferences. We also describe our process for training a weakly supervised reward model, which aims to minimise the data needed for train the reward model.

4.1 Reinforcement Learning for Cause Event Extraction

Our goal is to leverage the feedback from the trained evaluator described in Section 3 to improve the generative extractor to be better aligned with human preferences. See Figure 3 for an overview of our method.

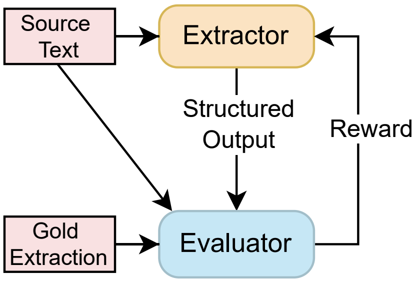


Figure 3: Architecture of our RL framework. PPO is used to optimise the extractor given the reward from the evaluator.

We initialise an RL policy from the FLAN-T5 supervised fine-tuned extractor (our reference model). It takes as input the source text and generates a structured output representing the cause and effect using our tagged format (Figure 2). Both input and output are sequences of tokens from the model vocabulary, which represents the action space. The policy itself is a probability distribution over the action space conditioned on the input tokens from the source text.

The RL objective is to find the optimal policy that maximises the reward. Our reward is generated by the evaluation model described in Section 3. It takes as input the source text, the gold standard extraction and the output from the RL policy, generating a scalar score. This is done at the sequence level, as a complete extraction is needed to determine the validity of the policy's output. Therefore, the score indicates whether the RL-generated extraction is valid, relative to the source text and the gold standard.

In addition to the reward model, we calculate the Kullback-Leibler (KL) divergence to measure the disparity between our policy and reference models. This helps us regulate the policy's ability to maintain the structured output format and prevent it from forgetting how to extract causes and effects. The final loss is a combination of the reward score and the KL divergence. We use the Proximal Policy Optimisation (PPO) algorithm to update the policy parameters by optimising this loss. During training, only the policy parameters are updated, while the reference and reward models are frozen.

4.2 Training a Weak Reward Model using Semi-Supervised Learning

Our approach works well but relies on the performance of the reward model. While we have trained a robust reward model, we explored scenarios with more limited data. To investigate this, we designed a weak-to-strong supervision process where we used a small portion of our dataset to train the evaluator, treating the remaining data as unlabelled for further improvements.

We randomly sampled $x\%$ of our labelled training data, where x is a hyperparameter. We first trained a DeBERTa classifier reward model on the $x\%$ data. We then used this classifier to generate labels for the remaining data. To gauge the model's confidence in each example, we applied softmax to its outputs and retained only those examples where the predicted class probability ranked in the top 75% separately for each class ('valid' and 'invalid'). This ensured an equal proportion of 'valid' and 'invalid' weak labels. Next, we combined these filtered examples with the original partial dataset to create the final weakly-supervised dataset, and trained a new DeBERTa model using this dataset.

Once we obtained a weakly-supervised reward model, we integrated it into our RL process to develop an RL-trained model. We then compared the performance of this new model with the original RL model trained with the full reward model. We find that the weakly-supervised RL model has competitive performance with the original RL model, demonstrating the effectiveness of our method (Section 5.3).

---

[4] Because SCITE is a small dataset, we could not train an effective evaluator with it. See Table 5 for dataset statistics.

5 Experiments

Datasets. We employ three causal extraction datasets: FCR, FinCausal and SCITE. Table 5 shows statistics about them regarding the number of examples in each split. Figure 2 shows an example. Each entry contains an input context, cause and effect spans. These are converted to our tagged format, which represents the relations textually. Table A1 (Appendix) shows more information.

Dataset	**Number of examples**		
	Train	Dev	Test
FCR	19892	2482	2433
FinCausal	3397	641	817
SCITE	1078	191	-

Table 5: Dataset statistics.

Inplementation and Metrics. We use FLAN-T5-Large as our policy model and DeBERTa-v3-xsmall trained on human annotation data as our reward model (Section 3). For evaluation, we obtained the formatted outputs from FLAN-T5-Large and gave them to our Human Proximal evaluator, denoted as Human Prox.[5], along with the references and source text. We also include automatic metrics such as Exact Match, Precision, Recall and F1 for comparison.

Baselines. We compare with another extractive IE model, *Seq-tagging*, which is a sequence labelling model to predict cause/effect BIO labels for each token. For the generative IE models, we compare with our backbone model *FLAN-T5-Large*. We also compare with the commercial large language models *GPT-3.5* and *GPT-4*, both prompted with a structure generative format, using in-context learning. We also report metrics from the original dataset papers (Yang et al., 2022; Mariko et al., 2020; Li et al., 2021).

5.1 Main results

Table 6 shows the causal relation extraction results of various models across three datasets. We see that GPT-3.5 and GPT-4 underperform, along with the other baselines, such as sequence tagging.

Our models perform much better, with the RL variant achieving an improvement over the SFT version. This includes both automated metrics and our Human Proximal (Human Prox.) evaluator.

Our Human Proximal evaluator is the trained metric described in Section 3, which approximates the human preference. We show that our supervised models achieve big improvement over both baselines and GPT models, with the RL models further improving on them. As this happens on all three datasets, we establish the superiority of our approach over the baselines.

	P	**R**	**F1**	**EM**	**Human Prox.**
FCR					
GPT-3.5	74.07	70.23	67.64	33.99	47.02
GPT-4	74.53	69.27	64.70	28.24	39.66
FCR-Baseline	-	-	74.54	23.01	-
Seq-tagging	77.76	77.78	77.74	41.30	52.82
FLAN-T5-Large (SFT)	80.02	80.48	80.96	54.13	64.42
FLAN-T5-Large (RL)	**82.85**	**82.03**	**81.23**	**55.58**	**68.29**
FinCausal					
GPT-3.5	57.76	56.11	61.58	17.32	52.73
GPT-4	63.35	61.92	66.58	26.99	55.85
FinCausal-Baseline	50.99	51.74	51.06	11.11	-
Seq-tagging	21.59	27.05	60.82	01.56	05.62
FLAN-T5-Large (SFT)	78.19	77.93	78.52	**66.61**	81.12
FLAN-T5-Large (RL)	**88.60**	**88.70**	**88.64**	64.74	**84.40**
SCITE					
GPT-3.5	46.66	86.08	60.48	53.66	52.88
GPT-4	37.97	83.70	52.23	46.86	57.59
SCITE-Baseline	83.33	85.81	84.55	-	-
Seq-tagging	92.94	92.25	92.59	88.48	91.10
FLAN-T5-Large (SFT)	92.29	91.73	92.01	87.43	90.58
FLAN-T5-Large (RL)	**94.54**	**93.70**	**94.12**	**93.98**	**92.67**

Table 6: Causal relation extraction results on three datasets with automatic metrics and human evaluation showing our RL method performs the best in all three datasets.

5.2 Ablation results of our Reward model

To analyse the effects of our reward model trained on the human annotation dataset, we replace it with two representative alternatives: an entailment-based Natural Language Inference model (Williams et al., 2018) and SentenceTransformer (SentTF). Entailment represents whether the model output is a logical consequence of the input text, indicating the cause-effect relation. SentenceTransformer is a pre-trained sentence embedding method, which we use to embed the gold extraction and model outputs, with the score being their normalised cosine similarity. Our reward model achieves the best Human Proximal score across the three datasets (Figure 4).

---

[5] This is the DeBERTa-Valid model trained with FCR defined in Section 3.

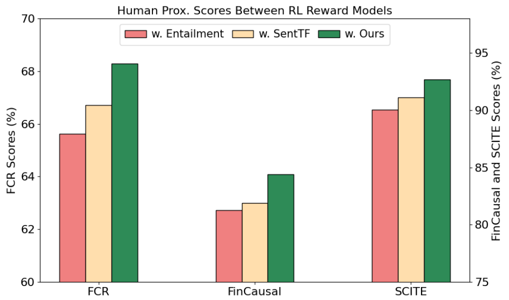


Figure 4: Ablation results for reward model with Human Proximal metric showing our reward model performs the best.

Tolerance to Wording Variance. Our reward model trained on the human annotation data captures the high-level semantic overlapping between gold extraction and model outputs. It is also capable of identifying the correctness of model outputs through source text understanding. Therefore, we use the "without EM (w/o EM)" metric to measure the percentage of correctly generated samples that are not exactly matched with the provided reference. This highlights the main improvement over automated metrics, where we can recognise results that are correct but would have otherwise been marked as incorrect because of their inexact result, showing clear advantages for our evaluator over using Entailment or SentenceTransformer.

5.3 Weak Supervision Evaluation

The results in Table 3 show an evaluator model highly aligned with human preference data. However, this requires a time-consuming and expensive process of manual annotation. To decrease the reliance on this process, we looked for ways to decrease the training set size.

We chose the FCR-based DeBERTa-Valid evaluator from Section 3, as it showed the highest agreement with human evaluation and other datasets. We experimented with subsets of different sizes and evaluated their performances. The results (Figure 5) show we can decrease the training set size with a small impact on the human agreement of the resulting evaluator. This motivated us to pursue a way to train a high-quality evaluator with less data.

Our weak supervision process has three steps. First, we sample a random subset of the training data as our initial supervised dataset and use it to train a partial evaluator. Second, we apply this partial evaluator to the remaining data, which we treat as unsupervised. We obtain the weak classification labels and the confidence of the evaluator for each entry and use a filtering process to determine which ones to keep. Third, we combine the filtered entries with the original subset and train a final evaluator. Our filtering process separates the weak labels into positive and negative sets, and for each set, takes the top 75% entries by confidence, so the final filtered set has an equal number of positive and negative entries.

Table 7 shows the results of our weak supervision experiments. We experimented with different subset sizes and found that the 50% subset achieves the best performance in terms of the Human Proximal and w/o EM metrics. It also matches the performance of the Full RL model, showing we can successfully decrease the reliance on human-annotated data without a performance cost.

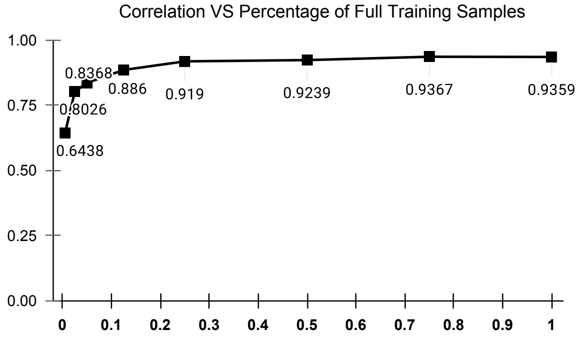


Figure 5: Evaluator agreement with human annotation by percentage of FCR data used.

Model	**P**	**R**	**F1**	**HumanProx.**	**w/o EM** ↑
SFT	80.02	80.48	80.96	64.42	10.29
Full RL	**82.85**	82.03	81.16	68.29	12.71
30% + weak	80.28	**84.19**	**82.18**	68.37	12.63
50% + weak	80.11	84.18	82.09	**68.86**	**13.07**
80% + weak	81.18	82.23	81.72	67.41	11.60

Table 7: RL with weakly-supervised models, showing the weakly-supervised variants are able to match the fully-trained model performance.

6 Conclusion

We have explored several evaluation approaches to address the inherent ambiguity of the causal event extraction task. We find that using a generative model to perform extraction performs well, but that evaluation with automated metrics is challenging. Our findings demonstrate the ability to faithfully reproduce human evaluation results using a DeBERTa-based classifier trained on human evaluation of extraction outputs. We also apply the evaluator as a reward model to Reinforcement Learning, further aligning our generative extractor model to human preferences.

We explore multiple datasets, showing how our approach can be generalised and employed our trained evaluator in a transfer setting, reducing the need for further annotation of new data. Finally, we propose a weak-to-strong approach where we only use a subset of annotated data to train a weakly-supervised evaluator that can match the performance of the fully-trained version.

Limitations

The datasets we used are limited to ones where the causes and effects are spans of the source text. Our approach does not work well with datasets where the events are instead represented by trigger words, as is common in other datasets, or when the answers are free text, not spans of the source text.

Another limitation is how we define the input of our evaluation. We require the reference and without it, the evaluator does not perform well. This means we are limited to datasets where we have such a reference, preventing us from applying the evaluator to those with blind data where we only have the source text.

Finally, we focused on small models like DeBERTa-v3 and FLAN-T5 because of how small our datasets were. We expected smaller models would be more data-efficient than large ones like Llama.

Acknowledgements

This work was supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) through a Turing AI Fellowship (grant no. EP/V020579/1, EP/V020579/2) and a New Horizons grant (grant no. EP/X019063/1), and Innovate UK through the Accelerating Trustworthy AI programme (grant no. 10093055).

References

Guillaume Becquin. 2020. GBe at FinCausal 2020, Task 2: Span-based Causality Extraction for Financial Documents. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 40–44, Barcelona, Spain (Online). COLING.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Louis Castricato, Alexander Havrilla, Shahbuland Matiana, Michael Martin Pieler, Anbang Ye, Ian Yang, Spencer Frazier, and Mark O. Riedl. 2022. Robust preference learning for storytelling via contrastive reinforcement learning. *ArXiv*, abs/2210.07792.

Víctor Uc Cetina, Nicolás Navarro-Guerrero, Ana Martín-González, Cornelius Weber, and Stefan Wermter. 2021. Survey on reinforcement learning for language processing. *Artificial Intelligence Review*, 56:1543–1575.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. Alpagasus: Training a better alpaca with fewer data.

Miao Chen, Ganhui Lan, Fang Du, and Victor Lobanov. 2020. Joint learning with pre-trained transformer on named entity recognition and relation extraction tasks for clinical analytics. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 234–242, Online. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew 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.

Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, Jiafeng Guo, and Xueqi Cheng. 2023. Retrieval-augmented code generation for universal information extraction. *CoRR*, abs/2311.02962.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *CoRR*, abs/2305.14450.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2022. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations*.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Viet Dac Lai, Chien Van Nguyen, Nghia Trung Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. *ArXiv*, abs/2307.16039.

Zhaoning Li, Qi Li, Xiaotian Zou, and Jiangtao Ren. 2021. Causality extraction based on self-attentive bilstm-crf with transferred embeddings. *Neurocomputing*, 423:207–219.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Youmi Ma, Tatsuya Hiraoka, and Naoaki Okazaki. 2022. Joint entity and relation extraction based on table labeling using convolutional neural networks. In *Proceedings of the Sixth Workshop on Structured Prediction for NLP*, pages 11–21, Dublin, Ireland. Association for Computational Linguistics.

Dominique Mariko, Hanna Abi-Akl, Estelle Labidurie, Stephane Durfort, Hugues De Mazancourt, and Mahmoud El-Haj. 2020. The financial document causality detection shared task (FinCausal 2020). In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 23–32, Barcelona, Spain (Online). COLING.

Sewon Min, Jordan L. Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, Colin Raffel, Adam Roberts, Tom Kwiatkowski, Patrick S. H. Lewis, Yuxiang Wu, Heinrich Küttler, Linqing Liu, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Sohee Yang, Minjoon Seo, Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Edouard Grave, Ikuya Yamada, Sonse Shimaoka, Masatoshi Suzuki, Shumpei Miyawaki, Shun Sato, Ryo Takahashi, Jun Suzuki, Martin Fajcik, Martin Docekal, Karel Ondrej, Pavel Smrz, Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Sejr Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen-tau Yih. 2021. Neurips 2020 efficientqa competition: Systems, analyses and lessons learned. *CoRR*, abs/2101.00133.

Paramita Mirza and Sara Tonelli. 2016. CATENA: CAusal and TEmporal relation extraction from NAtural language texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 64–75, Osaka, Japan. The COLING 2016 Organizing Committee.

Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. 2021. Named entity recognition and relation extraction: State-of-the-art. *ACM Computing Surveys (CSUR)*, 54(1):1–39.

OpenAI. 2023. GPT-4 Technical Report.

Naoki Otani, Michael Gamon, Sujay Kumar Jauhar, Mei Yang, Sri Raghu Malireddi, and Oriana Riva. 2022. Lite: Intent-based task representation learning using weak supervision. In *North American Chapter of the Association for Computational Linguistics*.

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 *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jing-Cheng Pang, Pengyuan Wang, Kaiyuan Li, Xiong-Hui Chen, Jiacheng Xu, Zongzhang Zhang, and Yang Yu. 2023. Language model self-improvement by reinforcement learning contemplation. *ArXiv*, abs/2305.14483.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 311, Philadelphia, Pennsylvania. Association for Computational Linguistics.

Ji Qi, Chuchun Zhang, Xiaozhi Wang, Kaisheng Zeng, Jifan Yu, Jinxin Liu, Lei Hou, Juanzi Li, and Xu Bin. 2023. Preserving knowledge invariance: Rethinking robustness evaluation of open information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5876–5890. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, L'eonard Hussenot, Orgad Keller, Nikola Momchev, Sabela Ramos, Piotr Stańczyk, Nino Vieillard, Olivier Bachem, Gal Elidan, Avinatan Hassidim, Olivier Pietquin, and Idan Szpektor. 2023. Factually consistent summarization via reinforcement learning with textual entailment feedback. In *Annual Meeting of the Association for Computational Linguistics*.

Anik Saha, Jian Ni, Oktie Hassanzadeh, Alex Gittens, Kavitha Srinivas, and Bulent Yener. 2022. SPOCK at FinCausal 2022: Causal Information Extraction Using Span-Based and Sequence Tagging Models. In *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, pages 108–111, Marseille, France. European Language Resources Association.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2023. Gollie: Annotation guidelines improve zero-shot information-extraction. *CoRR*, abs/2310.03668.

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

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation.

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Canoee Liu, Simon Tong, Jindong Chen, and Lei Meng. 2023. Rewritelm: An instruction-tuned large language model for text rewriting. In *AAAI Conference on Artificial Intelligence*.

Velizar Shulev and Khalil Sima'an. 2024. Continual reinforcement learning for controlled text generation. In *International Conference on Language Resources and Evaluation*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022a. Self-Consistency Improves Chain of Thought Reasoning in Language Models.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022b. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022a. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference.

Linyi Yang, Zhen Wang, Yuxiang Wu, Jie Yang, and Yue Zhang. 2022. Towards Fine-grained Causal Reasoning and QA.

Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. 2020. Fine-tuning pretrained language model with weak supervision: A contrastive-regularized self-training approach. *ArXiv*, abs/2010.07835.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348.

A Dataset Transformation

Our chosen datasets come in different formats, which we must transform into our tagged format. FCR is a collection of JSON files, where each entry contains the text and character indices for the cause and effect spans. FinCausal contains semicolon-separated CSVs, where each entry contains the input text and each cause effect spans as text. SCITE comprises XML files, where each item is a tagged representation of the sentences and their spans.

We convert them to a common format that is used as the base for all of our models: a tagged representation, shown in Figure 2. For FinCausal and SCITE, which do not contain relations like FCR does, we hard-code the Relation to 'cause'.

The original SCITE dataset has examples with more than one relation, which our models do not support. We opted to use only the first causal relation for each example.

B Further Dataset Statistics

Table 5 in the main text shows the count of instances per dataset and split. We now show the average number of words for the source text, cause and effect clauses in Table A1.

Dataset	**Average number of words**		
	Context	Cause	Effect
FCR	31.37	10.43	10.79
FinCausal	42.77	18.23	17.20
SCITE	18.68	2.15	2.03

Table A1: Dataset statistics: average number of words per part.

C Implementation Details

We used the KL divergence during training to ensure that the policy does not deviate too much from the format it learned during supervised fine-tuning (SFT). We found that some of the batches during RL training would lead to very high KL values, which would move the model too far in a given direction, often leading to parameter collapses (i.e. model weights going to NaN or infinity) or degenerate output (no longer recognisable as structured text).

To prevent this, we found that skipping batches with high KL values (over 2) made training considerably more stable, as we only applied updates from batches whose output was not too far from the reference model. The downside is that this slows down training, as skipping batches means fewer updates, potentially leaving the policy in a local optimum. In our experience, this trade-off was worth it, considering we still achieved improvements in all our main RL experiments.

Hyperparameters. The SFT models used FLAN-T5-Large as the base. The hyperparameters were the same across all datasets: input sequence length of 128 tokens, 20 training epochs, fixed learning rate of 0.0001 and greedy decoding for generation. We used an early-stopping scheme with the patience of 5 epochs without improvement based on the token F1 metrics.

The RL models were mostly similar, too: we used a single epoch, with the PPO process using a learning rate of 0.00014. The initial KL coefficient varied by dataset, with FCR using 0.4, SCITE using 0.2 and FinCausal 0.05. For generation, the RL models used beam search (2 beams) with multinomial sampling. Other parameters used the default values from the Transformers and TRL libraries. Other configuration options, such as reward normalisation and scaling, did not lead to any improvements. We found the RL models to be highly sensitive to the hyperparameters.

The evaluation (reward) model was based on DeBERTa-V3-xsmall. Its input sequence length was 400 tokens (to fit the input context and reference extraction), learning rate of 0.00001 and 100 epochs, with early stopping patience of 10 epochs without improvement based on the classification F1 score. The reward models were largely robust across different hyperparameter values and even sizes: with larger DeBERTa models not leading to significant improvements, we preferred using the smallest model to decrease memory concerns when using it alongside the larger FLAN-T5 model

D Software Used

Versions. We used Transformers[6] 4.33 to train the FLAN-T5 and DeBERTa LLMs. For RL training, we used TRL[7] 0.8.6. All experiments were run using Python 3.12 on Ubuntu 20.04 with an NVIDIA A100 40 GB GPU running CUDA 12.2. We also used NumPy[8] 1.24 and PyTorch[9] 2.0.

---

[6] `https://github.com/huggingface/transformers`

[7] `https://github.com/huggingface/trl`

[8] `https://numpy.org`

[9] `https://pytorch.org`

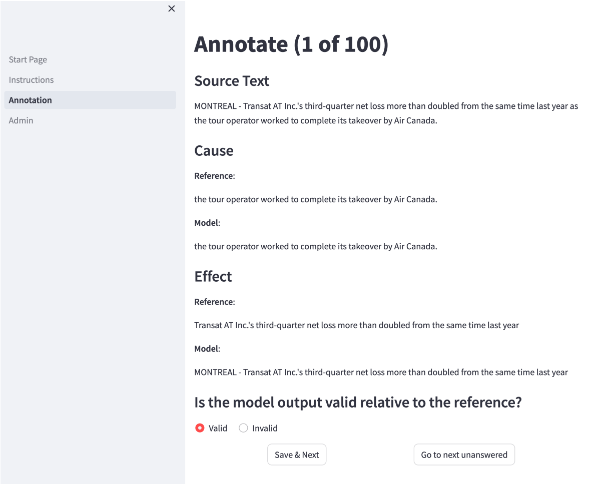
Licenses. From the software mentioned above, NumPy and PyTorch use the BSD license, TRL and Transformers use Apache-2.0, and Python uses the PSF license. The original code for this project is licensed under GPL-3.0.

AI assistance. GitHub Copilot[10], ChatGPT[11] and Claude[12] were used to assist in the development of the code, while Perplexity[13] was used for general queries.

E Human Annotation

We built an online annotation platform using Streamlit[14] version 1.35. It was deployed on a Digital Ocean[15] Droplet. Figure 6 shows a screenshot of the annotation page of the platform with an example from the FinCausal dataset.

The users were able to read the source text and compare the reference and model outputs for each entry before selecting whether the entry was 'valid' or 'invalid'. The platform saved the answers as soon as they were confirmed and allowed the users to leave and return later to continue from where they stopped.



Annotate (1 of 100)

Source Text

MONTREAL - Transat AT Inc.'s third-quarter net loss more than doubled from the same time last year as the tour operator worked to complete its takeover by Air Canada.

Cause

Reference:

the tour operator worked to complete its takeover by Air Canada.

Model:

the tour operator worked to complete its takeover by Air Canada.

Effect

Reference:

Transat AT Inc.'s third-quarter net loss more than doubled from the same time last year

Model:

MONTREAL - Transat AT Inc.'s third-quarter net loss more than doubled from the same time last year

Is the model output valid relative to the reference?

● Valid ○ Invalid

Figure 6: Screenshot of our annotation platform showing an example from the FinCausal dataset

---

[10] `https://github.com/features/copilot`

[11] `https://chat.openai.com/`

[12] `https://claude.ai`

[13] `https://perplexity.ai/`

[14] `https://streamlit.io`

[15] `https://www.digitalocean.com`