PURR: Efficiently Editing Language Model Hallucinations by Denoising Language Model Corruptions

Anthony Chen[1*] **Panupong Pasupat**[2] **Sameer Singh**[1] **Hongrae Lee**[3] **Kelvin Guu**[1]
[1]University of California, Irvine [2]Google DeepMind [3]Google Search
{anthony.chen, sameer}@uci.edu {ppasupat, hrlee, kguu}@google.com

Abstract

The remarkable capabilities of large language models have been accompanied by a persistent drawback: the generation of false and unsubstantiated claims commonly known as "hallucinations". To combat this issue, recent research has introduced approaches that involve editing and attributing the outputs of language models, particularly through prompt-based editing. However, the inference cost and speed of using large language models for editing currently bottleneck prompt-based methods. These bottlenecks motivate the training of compact editors, which is challenging due to the scarcity of training data for this purpose. To overcome these challenges, we exploit the power of large language models to introduce corruptions (*i.e.*, noise) into text and subsequently fine-tune compact editors to denoise the corruptions by incorporating relevant evidence. Our methodology is entirely unsupervised and provides us with faux hallucinations for training in any domain. Our *Petite Unsupervised Research and Revision* model, PURR, not only improves attribution over existing editing methods based on fine-tuning and prompting, but also achieves faster execution times by orders of magnitude.[1]

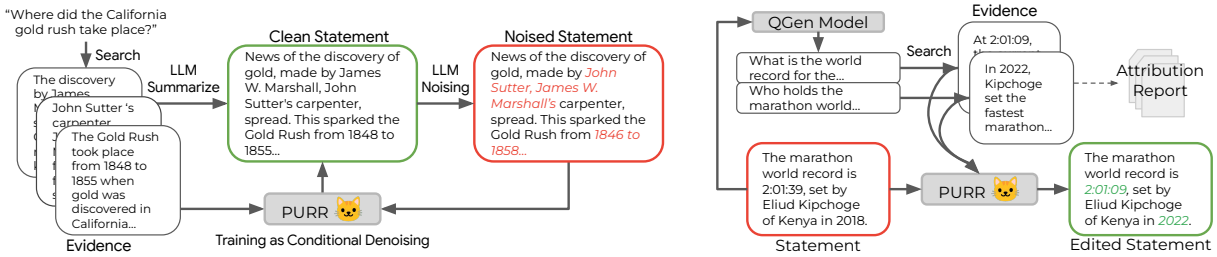
1 Introduction

As the strengths of large language models (LLMs) have become prominent (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023), so too have their weaknesses (Bender et al., 2021). A glaring weakness of LLMs is their penchant for generating false, biased, or misleading claims in a phenomena broadly referred to as "hallucinations" (Maynez et al., 2020; Krishna et al., 2021; Longpre et al., 2021; Raunak et al., 2021). Most LLMs also do not ground their generations to any source, exacerbating this weakness (Rashkin et al., 2021).

Post-hoc attribution and edit strategies offer promising solutions to tackle the problems of grounding and hallucination in language models (Thorne and Vlachos, 2020; Gao et al., 2022). These approaches retrieve supporting evidence to attribute the output (referred to as a claim) of a language model, followed by an editor that corrects factual errors in the claim, ensuring consistency with the evidence. A notable advantage of post-hoc methods is their modularity: they can be easily applied to any text regardless of their generation source. However, existing editors exhibit distinct strengths and weaknesses. Sufficiently large language models can be few-shot prompted to perform editing (Bai et al., 2022; Gao et al., 2022). However, there is currently a steep compute-quality tradeoff, where only the largest, most expensive models can perform this task well. Even then, significant quality headroom remains, as we will show. In contrast, much smaller, cheaper models can be fine-tuned to perform editing, but are limited to specific domains where adequate training data is available (Iv et al., 2022; Schick et al., 2022).

Instead of utilizing LLMs as prompted editors, we leverage their general-purpose capabilities to introduce challenging corruptions (*i.e.*, noise) to clean pieces of text. Subsequently, we fine-tune compact editors to denoise these corruptions by grounding onto relevant evidence. While text to corrupt is readily available, we do not assume that paired relevant evidence is provided. To tackle this, our data generation pipeline first searches for a collection of topically related evidence. We then employ an LLM summarize the evidence into a claim which is then noised (Fig. 1a). The evidence is then used to ground the denoising. In contrast to existing work that assumes access to relevant paired evidence to ground the edit when training (Balachandran et al., 2022) or assumes edit data is provided for training (Schick et al., 2022; Iv et al., 2022), our approach eliminates these assumptions.

[*]Work started during an internship at Google Research.
[1]The data generation pipeline, training data, and PURR checkpoints will be released.



(a) **Training PURR.** Given a seed query, we search for relevant evidence and summarize them into a claim which we corrupt. PURR is trained to denoise the corruption conditioned on the evidence.

(b) **Using PURR.** Given an ungrounded statement, we generate questions to search for relevant evidence which is then used to produce an edit.

Figure 1: **Training and Using PURR.**

Furthermore, unlike distillation where a challenging distillation set is vital and the student model generally under-performs the teacher (Beyer et al., 2021; Stanton et al., 2021), our noising process introduces challenging corruptions and our resulting editor trained on these corruptions surpasses the performance of the LLM used for noising when the same LLM is employed for prompted editing on multiple datasets.

Our *Petite Unsupervised Research and Revision* model, PURR, is built by fine-tuning a fusion-in-decoder T5 model on denoising data from our data generation pipeline (Raffel et al., 2020; Izacard and Grave, 2021). Because our goal is to improve attribution broadly across tasks and domains, we evaluate PURR on outputs of large language models on multiple question answering and dialog datasets. On all benchmarks, PURR outperforms much more expensive LLM-prompted editors in improving attribution while being orders of magnitude faster.

2 Editing for Attribution

2.1 Task Overview

While there are various ways to apply editing to the outputs of large language models, the primary objective of PURR is to present efficient methods for attributing the outputs of language models and rectifying inaccuracies, referred to as *Editing for Attribution* (Gao et al., 2022). In this task, a system is provided with a textual statement, x, and is tasked to produce an *attribution report*. The attribution report consists of a collection of evidence snippets, $A = \{e_1, e_2, \ldots, e_n\}$, that grounds the information in x. Additionally, the system is asked to produced a revised statement (*i.e.*, edit), y, that fixes any inaccuracies in x that contradict the content in A. For completeness, we present a summary of the task and refer interested readers to Gao et al. (2022) for a more comprehensive discussion.

2.2 Evaluation Metrics

Following Gao et al. (2022), we evaluate editing-for-attribution systems along two dimensions: **attribution**, the extent to which the original and revised statements can be attributed to the attribution report, and **preservation**, which measures how much information has changed from x to y. The objective of the task is to maximally attribute a textual statement while preserving the original intent of the language model generation to the greatest extent possible. We use automated metrics developed by Gao et al. (2022) to measure both attribution and preservation, which were found to have strong correlation to human raters. It is important to note that this evaluation setup does not require reference edits and only relies on the grounding between the textual statements and the attribution report.

Attribution A textual statement is generally said to be attributable to a set of evidence if one could reasonably say that given the evidence set, the statement is entailed (Rashkin et al., 2021). To formalize this, Gao et al. (2022) introduce an evaluation metric based on sentence-level natural langauge inference (NLI) model. Given an attribution report, A, and a textual statement y consisting of sentences, $y = \{s_1, s_2, \ldots\}$, we use a NLI model to measure the likely that each sentence is entailed by an evidence snippet in A: $\text{NLI}(e, s_i)$. The attribution of the entire statement, y, is computed as the average over the maximum attribution score for each constituent sentence.

$$\text{Attr}_{(s,A)} = \max_{e \in A} \text{NLI}(e, s) \quad (1)$$

$$\text{Attr}_{(y,A)} = \text{avg}_{s \in y} \, \text{Attr}_{(s,A)} \quad (2)$$

The goal of editing is to have $\text{Attr}_{(y,A)}$ be higher than $\text{Attr}_{(x,A)}$.

Preservation Preservation is measured using character-level Levenshtein distance between x and y. Preservation is 1 if the statements are the same and 0 if y has completely changed all textual information in x.

$$\text{Pres}_{(x,y)} = \max\left(1 - \frac{\text{Lev}(x,y)}{\text{length}(x)}, 0\right) \quad (3)$$

To capture our goal of maximal attribution with maximal preservation, we unify these two metrics by computing the harmonic mean, $F1_{AP}$, of $\text{Attr}_{(y,A)}$ and $\text{Pres}_{(x,y)}$.

2.3 Evaluation Sets

Our goal is to improve attribution broadly across tasks and domains on the outputs of strong generations systems. Gao et al. (2022) construct evaluation sets by prompting strong LLMs to generate outputs on three tasks: Natural Questions (factoid question answering) (Kwiatkowski et al., 2019), StrategyQA (reasoning-chain question answering) (Geva et al., 2021), and QreCC (knowledge-intensive dialogue) (Anantha et al., 2021). Gao et al. (2022) generate 150 validation and 150 test instances for each dataset using PALM for Natural Questions and StrategyQA and LaMBDA on QReCC (Chowdhery et al., 2022; Thoppilan et al., 2022). We use these sets and tune on the validation sets and report results on the test sets.

2.4 Baselines

PURR and all baselines follow a **research-and-revision** pipeline. In the **research** stage, the objective is to search for relevant pieces of evidence to ground the information in the textual statement, x. This stage remains consistent across all baselines. We first prompt a large language model to generate a set of queries $Q = \{q_1, q_2, \ldots q_m\}$ that attempts to cover all pieces of information in x that needs verification. Subsequently, we use Google Search in conjunction with a passage extractor to find the most relevant evidence snippet for each query, constituting an evidence set $E = \{e_1, e_2 \ldots, e_m\}$.

In the **revision** stage, an editor is given the original statement, x, the set of queries, Q, and the evidence set, E, and asked to produce a revised statement, y. y can be the same as x in the event the editor deems the original statement cannot be edited further to increase attribution. We measure the ability of different editors to abstain from editing later on. We compare PURR against two baseline editors.

EFEC is a fine-tuned T5 editor trained on FEVER (Aly et al., 2021). EFEC was trained using evidence retrieved from Wikipedia and concatenates all pieces of evidence with the text statement to produce an edited statement. Notably, EFEC does not use the query set when making an edit. (Gao et al., 2022) found EFEC often improves attribution at the expense of preservation.

RARR is a prompt-based editing approach that builds upon PALM, a language model with 540 billion parameters (Chowdhery et al., 2022). Unlike EFEC, which incorporates all evidence simultaneously to produce an edit, RARR iteratively examines each evidence, e_i, by checking whether there is any contradictions between the text statement, x, and edits in the event there is. The process of contradiction checking and editing is performed using distinct few-shot prompts. Gao et al. (2022) demonstrate that this iterative approach to editing combined with few-shot prompting leads to improvements in attribution and preservation, albeit at the cost of multiple computationally expensive and slow calls to a large language model.

2.5 Generating the Attribution Report

To maintain a manageable scope, we limit the attribution report, A, to include only the five most relevant pieces of evidence from the evidence set, E. An attribution report of five evidence snippets was found to be able to attribute the information for the claims in the datasets we evaluate on. It is worth noting that when editing, there are no restrictions on the number of evidence snippets an editor can utilize. Given the evidence set, E, and the query set, Q, from the research stage, we employ a scoring module that evaluates the relevance of each evidence e_i to each query q_j, $S(q_i, e_j)$. Our objective is to identify a subset of evidence that maximizes the coverage across all queries to form the attribution report. This coverage is quantified as the sum of the highest relevance scores achieved by each query with respect to any evidence. For scoring, we use a cross-encoder[2].

$$\text{Cov}_{(E,Q)} = \sum_{i=1}^{N} \max_{e_j \in E} S(q_i, e_j) \quad (4)$$

[2]https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2

3 Efficient Editing by Denoising

In this section, we present an overview of PURR, highlight its distinguishing features compared to baselines, and describe the denoising training strategy.

3.1 Overview of PURR at Inference Time

We first describe how PURR is used at inference time and highlight the differences between PURR and baselines (Fig. 1b). Similar to EFEC, PURR is built upon on the T5 model, specifically T5-large. Furthermore, our editing framework adopts a similar approach to EFEC in terms of incorporating all available evidence simultaneously when making an edit. However, instead of concatenating the evidence in the input, we employ fusion-in-decoder (FiD) to effectively aggregate information across evidence (Izacard and Grave, 2021). This approach has demonstrated superior performance in merging information and allows us to surpass the context length limits imposed by modern language models. Finally, rather than employing a prompted language model for query generation during the research stage, we employ distillation to train a T5-large query generation model. Although our primary focus lies in enhancing the editing process, we opt for distillation during query generation as well to ensure that our editing pipeline does not rely on prompting.

q:	Who will be the new coach of the Detroit lions?
E^+:	- On Jan. 20, 2021 the Detroit Lions named Dan Campbell the franchise's new head coach... - Campbell possesses 23 years of NFL experience, including 12 years as a coach and 11 as a player. In his first year... - On Jan. 20, 2021 the Detroit Lions named Dan Campbell the franchise's new head coach...
x/y:	Dan Campbell was appointed the new ~~head~~ ***assistant*** coach of the Detroit Lions on January 20, 2021. With ~~23~~ ***19*** years of NFL experience, 12 as a coach and ~~11~~ ***7*** as a player...
q:	What is the neurological explanation for why people laugh when they're nervous or frightened?
E^+:	- A 2015 Yale study found people respond with a variety of emotions to strong outside stimuli... - Vilayanur Ramachandran states "We have nervous laughter because we want to make ourselves think what horrible thing we encountered isn't really as horrible as it appears"... - Stanley Milgram conducted one of the earliest studies about nervous laughter in the 1960s. His study revealed that people often laughed nervously in uncomfortable situations...
x/y:	Yale researchers in 2015 found people often respond to strong external stimuli with a variety of emotions, including ~~nervous laughter~~ ***anger***. ~~Stanley Milgram's~~ ***Vilayanur Ramachandran's*** 1960s study also observed this in uncomfortable situations. Neuroscientist ~~Vilayanur Ramachandran~~ ***Stanley Milgram*** theorizes that people laugh when....

Table 1: **Training Examples**. Our editing data covers a variety of domains and introduces challenging corruptions (*e.g.,* numerical, entity, and semantic role). q is the seed query, E^+ is the gold evidence set used to generate the clean statement, y is the clean statement and ***x is the corrupt statement***.

3.2 Creating Training Data via Noising

To train an editor to fix hallucinations, we need a dataset consisting of a clean statements, y, which are paired with a set of supporting evidence $E = \{e_1, e_2, \ldots, e_n\}$, as well as a corrupted statement, x. While collecting this data manually is feasible, doing so can be expensive, requiring scouring for evidence to ground an LLM generation followed by removing any inaccuracies in the generation. Instead, we remove this bottleneck by leveraging the general purpose generation capabilities of LLMs to create a training set in a completely fashion. We generate clean statements by providing a set of topically related evidence to the LLM, and then corrupt the statements to create simulated hallucinations (Fig. 1a). We provide the prompts used for summarization and corruption in Appendix A.

Generating Clean Statements With Evidence The first step is to create a statement, y, paired with a set of evidence, E, that attributes (*i.e.,* grounds) the statement. Our pipeline only requires a set of queries in the domain of interest to get started. We start with a query, q, and use a search engine to find evidence related to the question. We take the top web pages from the search engine and chunk them into passages. Using the same cross-encoder from the attribution report scoring module, we bin the passages that have the highest relevant score (beyond some threshold) to q into a set of gold evidence $E^+ = \{e_1^+, e_2^+, \ldots, e_i^+\}$ and the rest of the passages into a set of hard negative evidences $E^- = \{e_1^-, e_2^-, \ldots, e_j^-\}$. In our pipeline, we restrict the size of E^+ to contain at most four pieces of evidence. The resulting evidence set is the union of the gold and hard negative evidences $E = E^+ \cup E^-$. We then prompt a large language model to do zero-shot multi-document summarization of the gold evidence set, E^+. We use the resulting summary as the clean statement, y, and upon manual inspection, the summary has a high degree of faithfulness to the evidence set.

Noising and Conditional Denoising We take the clean statement, y, and noise it by prompting a large language model to corrupt the text resulting in the corruption x. Our prompt contains examples of corruptions, and covers a wide range of linguistic phenomena we observe when it comes to LLM hallucinations. These include incorrect dates and entities, semantic role errors, and quantification errors. Once noised claims paired with evidence is available, an editor can be trained by fine-tuning a sequence-to-sequence model to maximize $P(y|x, E)$. We call the resulting editor from denoising PURR.

3.3 Dataset Statistics and Training Details

We utilized GPT-3.5 `text-davinci-003` to facilitate the process of generating summaries and introducing corruption. Our choice of this particular model ensures that our generation strategy can be easily replicated. We started with roughly 6,000 seed queries covering a variety of domains and topics resulting in an edit dataset of 6,000 instances (Tab. 1). We reserve 10% for validation and use the resulting 90% for training. Each instance cost roughly 4 cents to generate and in total cost of roughly $250.

We fine-tune T5-large on our dataset using the validation loss to tune hyperparameters and determine training stoppage. During training, we pair each corrupted statement, x, with four pieces of evidence from the accompanying gold evidence set, E^+, to ground the edit and produce the clean statement, y. In the event that the gold evidence set has fewer than four evidence snippets, we randomly sample evidence from the negative evidence set, E^-, until we hit four snippets. We found adding negative evidence during training helps PURR ignore irrelevant evidence during inference.

4 Results

4.1 Primary Quantitative Results

We provide results on the editing-for-attribution task in Table 2. We report the attribution of the claim before and after editing and the preservation of the edited claim. Our primary metric, $F1_{AP}$, is the harmonic mean between the attribution and preservation of the edited claim. We first turn our attention to the baselines. EFEC, the editor that was fine-tuned with evidence largely from Wikipedia, struggles on this task. While EFEC improves attribution, this comes at the large expense of preservation and we see this in practice as EFEC tends to make large changes to the claim. RARR, the prompted editor, does not improve attribution as much as EFEC. However it is significantly better at preserving the intent of the original claim. Because of this, RARR is much better on the unified $F1_{AP}$ metric.

Model	Attr. ($x \rightarrow y$)	Pres.	$F1_{AP}$
PALM outputs on NQ			
EFEC	44.7 $\rightarrow$ **63.9**	39.6	48.5
RARR	44.7 $\rightarrow$ 53.8	89.6	67.2
PURR	44.8 $\rightarrow$ 59.8	**91.0**	**72.2**
PALM outputs on SQA			
EFEC	37.2 $\rightarrow$ **58.2**	31.0	40.4
RARR	37.2 $\rightarrow$ 44.6	89.9	59.6
PURR	36.9 $\rightarrow$ 47.1	**92.0**	**62.3**
LaMBDA outputs on QreCC			
EFEC	18.4 $\rightarrow$ **47.2**	39.0	42.7
RARR	18.4 $\rightarrow$ 28.7	80.1	42.2
PURR	16.8 $\rightarrow$ 33.0	**85.8**	**47.7**

Table 2: **Results on the *Editing for Attribution* task.** We report the attribution of the statement before and after editing, preservation after editing, and $F1_{AP}$ which combines attribution and preservation. Results are on LLM outputs on factoid question answering, long reasoning question answering, and dialog.

PURR improves upon the results of RARR by generally making smaller changes to the claim while improving the attribution in this more limited edit. Because of this, PURR pushes the state-of-the-art on the unified $F1_{AP}$ metric an all three tasks. Moreover, PURR is significantly more efficient to use by virtue of its size.

4.2 Breaking Down the Numbers

We dig into the edits to get a better sense of where PURR improves on the baselines. Based on the preservation, $\text{Pres}_{(x,y)}$, and attribution scores of the original statement, $\text{Attr}_{(x,A)}$, and edited statement, $\text{Attr}_{(y,A)}$, we say an edit can fall into one of the following sets:

- **Huge Edit**: We say an edit is "huge" if preservation is low: $\text{Pres}_{(x,y)} < 0.5$.
- **Bad Edit**: We say an edit is "bad" if the attribution after editing is lower than before: $\text{Attr}_{(y,A)} - \text{Attr}_{(x,A)} < -0.1$.
- **Unnecessary Edit**: We say an edit is "unneces-

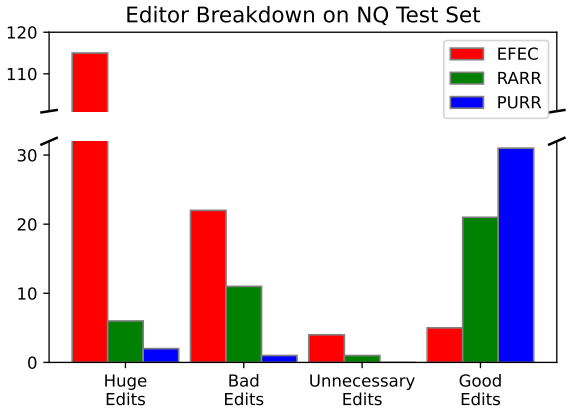


Figure 2: **Breakdown of edit types each editor makes** on the Natural Questions test set. EFEC makes huge edits while RARR sometimes over edits. PURR does a much better job at balancing attribution and preservation while rarely over-editing.

sary" if it is a bad edit and also $\text{Attr}_{(x,A)} > 0.9$. This means the editor made a poor edit when the attribution was already near perfect before editing.

- **Good Edit**: We say an edit is "good" if attribution has significantly improved while preservation is high: $\text{Attr}_{(y,A)} - \text{Attr}_{(x,A)} > 0.3$ and $\text{Pres}_{(x,y)} > 0.7$.

Note that unnecessary edits are a subset of bad edits. We take the 150 instances in the Natural Questions test set and categorize the edits each editor makes in Figure 2. On a majority of claims, EFEC makes large edits while rarely making edits that improve attribution while preserving the original claim. RARR does a much better job at minimizing large edits but there are still cases where RARR edits a claim in a way that reduces the attribution. PURR almost never makes large edits and never edits a claim when it is near-perfect in a way that reduces attribution. PURR also makes more good edits compared to the baselines.

4.3 Qualitative Analysis

We then dig into the PURR predictions and diagnose the strengths of PURR and examine where there is room for improvement. We show examples in Table 3 that we found are representative of the strengths of PURR and areas of potential improvement. We find that PURR is extremely strong at fixing entity and numerical hallucinations as well as longer spans. Additionally, because PURR uses fusion-in-decoder, it is adept at merging information across multiple pieces of evidence to make an edit. We noticed several instances where there are challenging distractors in evidence that can lead to an erroneous edit. Future work will introduce stronger corruptions in the data generation pipeline to better handle this case.

We next analyze the entire inference pipeline of PURR (Fig. 1b), which includes the question generation model, the search engine, and the editor itself. Our goal is to see when there is an error, which component is responsible. On the Natural Questions subset of the evaluation, we examine 20 instances where the attribution after editing,

Correctly revising an entity/span.
x: The US stopped printing $2 bills in 1966. The reason was that vending machines would not accept it. In 1976, the $2 bill was reintroduced...
E: During that time, most goods and services were less than a dollar, making paper currency impractical. As need and use declined, the Federal Reserve stopped printing $2 bills in 1966. In 1976, the Federal Reserve brought back the $2-dollar bill.
y: The US stopped printing $2 bills in 1966. The reason was that **most goods and services were less than a dollar.** In 1976, the $2 bill was reintroduced...
$Pres_{(x,y)}$: 0.81 $Attr_{(A,x)} \rightarrow Attr_{(A,y)}$: 0.61 $\rightarrow$ 0.84
Reasoning across multiple pieces of evidence
x: In 1899, the first driver's license was issued in France. The first license in the United States was issued in 1902 in Massachusetts.
E: - France introduced the first driving licences under the Paris Police Ordinance of 14 August 1893... - Massachusetts and Missouri were the first states to require a license in 1903, but there was no test associated with it...
y: In **1893**, the first driver's license was issued in France. The first license in the United States was issued in **1903** in Massachusetts.
$Pres_{(x,y)}$: 0.90 $Attr_{(A,x)} \rightarrow Attr_{(A,y)}$: 0.03 $\rightarrow$ 0.96
Challenging distractors in evidence
x: The first orca put in captivity was in 1965. It was at Seattle, Washington's Seattle Aquarium. The orca's name was Wanda.
E: - On Nov. 18, 1961 a killer whale had been spotted and a collection crew from the **Marineland of the Pacific** scrambled to capture her. Never had an orca been captured alive, and Wanda was trying to avoid being the first... - The first orca to be captured in 1961 was found off the coast of California and given the name Wanda.
y: The first orca put in captivity was in **1961**. It was at **Marineland of the Pacific**. The orca's name was Wanda.
$Pres_{(x,y)}$: 0.77 $Attr_{(A,x)} \rightarrow Attr_{(A,y)}$: 0.33 $\rightarrow$ 0.77

Table 3: **Example of good and bad revisions with PURR.** x = claim; E = relevant evidence; y = edited claim using E. PURR can handle hallucinated entities and spans as well as merge information across evidence to edit. PURR can struggle when there are challenging distractors in a piece of evidence.

Attr$_{(y,A)}$, is less than 0.30. Our qualitative analysis is provided in Table 4. Roughly 80% of the instances have low attribution after editing because either the question generation model we used did not fully cover the information in the claim or our search procedure did not find the best evidence for editing. We believe the question generation is the easier problem to fix while search is a much harder problem. Editing is a fairly small issue in comparison. Finally, there are some claims that fall into a "miscellaenous" category, either because it was not contextualized enough to properly edit or because the automatic metric erroneously assigned a low score.

4.4 Inference Speed and Cost Comparisons of Fine-tuned vs Prompted Editors

A key advantage of PURR over prompt-based editors are the lower computational costs. RARR, a prompt-based editor built upon 540B PALM, runs on dozens of TPUs and takes approximately 40 seconds to edit a single statement. In comparison, PURR can run on a 12GB GPU and takes approximately 2 seconds to edit a single statement on a Titan-RTX. Considering generating our training set costs <$300 USD which is quickly amortized, we recommend our synthetic data generation strategy for large-scale deployment given the speed and cost savings of PURR.

5 Related Work

Editing for Attribution PURR builds upon previous research on post-hoc editing methods aimed at enhancing the attribution and accuracy of generated text (Balachandran et al., 2021; Cao et al., 2020; Iso et al., 2020). Notably, RARR (Gao et al., 2022) and Rethinking-with-Retrieval (He et al., 2022) employ few-shot prompting to rectify language model outputs, exhibiting similarities to our work. FRUIT (Iv et al., 2022) and EFEC (Thorne and Vlachos, 2020) also utilize fine-tuned editors to achieve similar objectives, leveraging Wikipedia as a source of training data. PEER is trained on Wikipedia edits (Schick et al., 2022) and includes a component for enhancing factuality through editing, but its primary focus lies in collaborative writing. Our denoising approach combines the speed advantages of fine-tuned editors while circumventing the reliance on training data that is typically constrained to specific domains like Wikipedia.

Query Generation Missing Coverage (*35%*)

x: Legends of Tomorrow season 3 finale aired on April 9, 2018. It's title is No Country for Old Dads and is a 42-minute episode.

Q: - When did the season 3 finale of Legends of Tomorrow air?
- ~~*What's the title of Legends of Tomorrow season 3 finale?*~~
- ~~*How long is the season 3 finale of Legends of Tomorrow?*~~

y: Legends of Tomorrow season 3 finale aired on October 10, 2017. It's title is Freakshow and is a 43-minute episode.

$Attr_{(A,x)} \rightarrow Attr_{(A,y)}$: 0.01 $\rightarrow$ 0.01

Search Not Finding Best Evidence (*45%*)

x: Bryce Dallas Howard plays the character of the mother of Cindy Lou Who in the 2018 animated movie The Grinch.

E: ~~*Donna Lou Who is one of the characters of the 2018 Illumination animated film, Dr. Seuss' The Grinch. She's the mother of Cindy Lou Who and twin infants sons, Buster and Bean. She's voiced by Rashida Jones .*~~

y: No edit.

$Attr_{(A,x)} \rightarrow Attr_{(A,y)}$: 0.05 $\rightarrow$ 0.05

Editor Error/Confusion (*5%*)

x: James Marion Sims was the first to perform the first c section in 1846. This was in the US.

E: - The first recorded cesarean where both mother and child survived was done in Switzerland, in 1500. That was also a husband-wife affair, although in this case Jacob Nufer was a swine gelder, not a doctor.
- The first documented cesarean section on a living woman was performed in 1610; she died 25 days after the surgery.

y: James Marion Sims was the first to perform the first c section in **1610**. This was in the UK.

$Attr_{(A,x)} \rightarrow Attr_{(A,y)}$: 0.00 $\rightarrow$ 0.20

Miscellaneous (*15%*)

x: The song was written by Tony Macaulay and John Macleod and recorded by Helen Shapiro in 1961.

y: The song was written by Dub Allbritten and Ronnie Self and recorded by The Jackson 5 in 1960.

Claim not contextualized enough to properly edit.

$Attr_{(A,x)} \rightarrow Attr_{(A,y)}$: 0.00 $\rightarrow$ 0.01

Table 4: **Error Analysis of PURR Inference Pipeline.** We sample **20** edits from the NQ set where attribution is low after editing and categorize why by component. x = claim; Q = generated queries used for search, E = relevant evidence; y = edited claim using E. ~~*Strike-through text*~~ represents a query that wasn't generated or evidence that wasn't retrieved but should have been.

Improving Trust in Large Language Models Ensuring the safe deployment of large language models encompasses various considerations, beyond just factuality and attribution. Large language models have demonstrated the potential to regurgitate protected information (Carlini et al., 2020), spew hateful content (Gehman et al., 2020), and exhibit high sensitivity to input variations (Zhao et al., 2021). A common approach to addressing these issues has been via additional training such as instruction fine-tuning (Sanh et al., 2021; Min

et al., 2021; Chung et al., 2022; Ye et al., 2022), fine-tuning from human feedback (Ziegler et al., 2019; Stiennon et al., 2020), and more recently pre-training from human feedback (Korbak et al., 2023). In a similar vein to RARR, Bai et al. (2022) proposes to edit the outputs of LLMs using prompted LLMs to remove unsafe aspects of generated text. As part of our future research, we aim to apply our denoising strategy to train efficient compact editors for addressing such undesired generation behaviors.

Distilling Large Language Models Given their generation prowess, LLMs have been incorporated into data generation pipelines, essentially distilling the knowledge of the language model if their outputs are used for training (Wang et al., 2021; Bartolo et al., 2022; Lang et al., 2022; Smith et al., 2022). Eisenstein et al. (2022) follow a multi-step distillation pipeline like ours, chaining the outputs of multiple LLM calls and distilling the output into an explainable question answering model. Liu et al. (2022) uses the outputs of LLMs followed by filtering and human refinement to create WANLI, a challenging natural language inference dataset. On the evaluation side, Ribeiro and Lundberg (2022) use LLMs to generate evaluation sets for testing LLMs. While similar, our denoising approach *implicitly* distills the information in a large language model while simultaneously producing challenging training instances.

6 Conclusion

Factuality and attribution are vital for the safe deployment of large language models. However, these mechanisms are inherently lacking in LLMs. Recent work has proposed augmenting the outputs of LLMs by retrieving evidence to attribute their outputs followed by prompting another LLM to edit the outputs to remove inconsistencies. However, there is a heavy computational cost which bottleneck these methods which motivates a need to develop efficient editors, but this is hindered by training data scarcity. To overcome these challenges, we use LLMs to corrupt text and fine-tune compact editors to denoise these faux hallucinations. Our denoising method is completely unsupervised and our resulting editor, PURR, improves attribution performance across various datasets over prompted editors, while being order of magnitude faster to execute.

References

Rami Aly, Christos Christodoulopoulos, Oana Cocarascu, Zhijiang Guo, Arpit Mittal, Michael Schlichtkrull, James Thorne, and Andreas Vlachos, editors. 2021. *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Dominican Republic.

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-domain question answering goes conversational via question rewriting. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 520–534, Online. Association for Computational Linguistics.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional ai: Harmlessness from ai feedback.

Vidhisha Balachandran, Hannaneh Hajishirzi, William Cohen, and Yulia Tsvetkov. 2022. Correcting diverse factual errors in abstractive summarization via post-editing and language model infilling. In *Conference on Empirical Methods in Natural Language Processing*.

Vidhisha Balachandran, Ashish Vaswani, Yulia Tsvetkov, and Niki Parmar. 2021. Simple and efficient ways to improve REALM. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 158–164, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Max Bartolo, Tristan Thrush, Sebastian Riedel, Pontus Stenetorp, Robin Jia, and Douwe Kiela. 2022. Models in the loop: Aiding crowdworkers with generative annotation assistants. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3754–3767, Seattle, United States. Association for Computational Linguistics.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the

dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*.

Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. 2021. Knowledge distillation: A good teacher is patient and consistent. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10915–10924.

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, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff 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. *ArXiv*, abs/2005.14165.

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6251–6258, Online. Association for Computational Linguistics.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Xiaodong Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2020. Extracting training data from large language models. In *USENIX Security Symposium*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311.

Hyung Won Chung, Le Hou, S. 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, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Jacob Eisenstein, Daniel Andor, Bernd Bohnet, Michael Collins, and David Mimno. 2022. Honest students from untrusted teachers: Learning an interpretable question-answering pipeline from a pretrained language model. *ArXiv*, abs/2210.02498.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, N. Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2022. Rarr: Researching and revising what language models say, using language models. *ArXiv*, abs/2210.08726.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Hangfeng He, Hongming Zhang, and Dan Roth. 2022. Rethinking with Retrieval: Faithful Large Language Model Inference. *ArXiv*.

Hayate Iso, Chao Qiao, and Hang Li. 2020. Fact-based Text Editing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 171–182, Online. Association for Computational Linguistics.

Robert Iv, Alexandre Passos, Sameer Singh, and Ming-Wei Chang. 2022. FRUIT: Faithfully reflecting updated information in text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3670–3686, Seattle, United States. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Bhalerao, Christopher L. Buckley, Jason Phang, Sam Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. *ArXiv*, abs/2302.08582.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4940–4957, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Hunter Lang, Monica Agrawal, Yoon Kim, and David A. Sontag. 2022. Co-training improves prompt-based learning for large language models. In *International Conference on Machine Learning*.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shayne Longpre, Kartik Kumar Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. *ArXiv*, abs/2109.05052.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan T. McDonald. 2020. On faithfulness and factuality in abstractive summarization. *ArXiv*, abs/2005.00661.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. *ArXiv*, abs/2110.15943.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and D. Reitter. 2021. Measuring attribution in natural language generation models. *ArXiv*, abs/2112.12870.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183, Online. Association for Computational Linguistics.

Marco Tulio Ribeiro and Scott Lundberg. 2022. Adaptive testing and debugging of NLP models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3253–3267, Dublin, Ireland. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Stella Rose Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization. *ArXiv*, abs/2110.08207.

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. 2022. Peer: A collaborative language model. *ArXiv*, abs/2208.11663.

Ryan Smith, Jason Alan Fries, Braden Hancock, and Stephen H. Bach. 2022. Language models in the loop: Incorporating prompting into weak supervision. *ArXiv*, abs/2205.02318.

Samuel Stanton, Pavel Izmailov, P. Kirichenko, Alexander A. Alemi, and Andrew Gordon Wilson. 2021. Does knowledge distillation really work? *ArXiv*, abs/2106.05945.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan J. Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. *ArXiv*, abs/2009.01325.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam M. Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, Yaguang Li, Hongrae Lee, Huaixiu Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, I. A. Krivokon, Willard James Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju

Duke, Johnny Hartz Søraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Díaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravindran Rajakumar, Alena Butryna, Matthew Lamm, V. O. Kuzmina, Joseph Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Huai hsin Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. *ArXiv*, abs/2201.08239.

James Thorne and Andreas Vlachos. 2020. Evidence-based factual error correction. In *Annual Meeting of the Association for Computational Linguistics*.

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

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? GPT-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Seonghyeon Ye, Doyoung Kim, Joel Jang, Joongbo Shin, and Minjoon Seo. 2022. Guess the instruction! flipped learning makes language models stronger zero-shot learners. *ArXiv*, abs/2210.02969.

Tony Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *ArXiv*, abs/2102.09690.

Daniel M. Ziegler, Nisan Stiennon, Jeff Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *ArXiv*, abs/1909.08593.

A Prompts for Creating Training Data

```
Summarize all the pieces of text. Paraphrase the text and change the syntax.

{text}

Summary:
```

Figure 3: Zero-shot prompt for multi-document summarization. The input `{text}` can be multiple pieces of text from different sources separated by a new-line.

```
Corrupt the text by first generating a reasoning that describes what you will change, then following the reasoning to change the the text. Make the reasoning false but believable. Do not remove any information.

Text: The new revelation came Monday as the Department of Justice filed federal charges of assault and attempted kidnapping against the man suspected of attacking Paul Pelosi.
Number of things to change: 1.
Reasoning: I am going to swap "the man" and "Paul Pelosi".
Corruption: The new revelation came Monday as Paul Pelosi filed federal charges of assault and attempted kidnapping against Paul Pelosi suspected of attacking the man.

Text: Grapes of Wrath is a novel published in 1939, written by John Steinbeck. It takes place during the Great Depression, and focuses on the Joad family and their journey from Oklahoma to California.
Number of things to change: 3.
Reasoning: I am going to change when the novel was published to "1937", what it focuses on to "class discrimination", and add the fact that John Steinbeck was British.
Corruption: Grapes of Wrath is a novel published in 1937, written by the British author John Steinbeck. It takes place during the Great Depression, and focuses on the class discrimination on display.

Text: {text}
Number of things to change: {num_corruptions}.
Reasoning:
```

Figure 4: Few-shot prompt for corruption. Our corruption happens in a chain-of-thought fashion (Wei et al., 2022), allowing more flexibility in determining how many pieces of information to corrupt and what kinds of information to change.