Detecting Edit Failures In Large Language Models: An Improved Specificity Benchmark

Jason Hoelscher-Obermaier[1*] **Julia H. Persson**[1*]

Esben Kran[1] **Ioannis Konstas**[2] **Fazl Barez**[1,2,3*]

[1] Apart Research
[2] Edinburgh Centre for Robotics
[3] Department of Engineering Sciences, University of Oxford

Abstract

Recent model editing techniques promise to mitigate the problem of memorizing false or outdated associations during large language model (LLM) training. However, we show that these techniques can introduce large unwanted side effects which are not detected by existing specificity benchmarks. We extend the existing COUNTERFACT benchmark to include a dynamic component and dub our benchmark COUNTERFACT+. Additionally, we extend the metrics used for measuring specificity by a principled $\mathcal{KL}$ divergence-based metric. We use this improved benchmark to evaluate recent model editing techniques and find that they suffer from low specificity. Our findings highlight the need for improved specificity benchmarks that identify and prevent unwanted side effects.

1 Introduction

Although large language models (LLMs) are powerful tools for generating human-like language, they can also memorize false or outdated associations, limiting their applicability. Model editing techniques promise to solve this problem by correcting non-factual associations. It is important that model edits are highly specific in the sense of not introducing any unwanted associations as a side effect. In this paper, we discuss why the current benchmark for specificity falls short and propose a more challenging, dynamic specificity benchmark to evaluate model editing techniques. Using this benchmark, we evaluate recent model editing techniques and find previously unreported side effects. We highlight the importance of improved specificity benchmarks for the effective and safe use of LLMs subject to model edits.

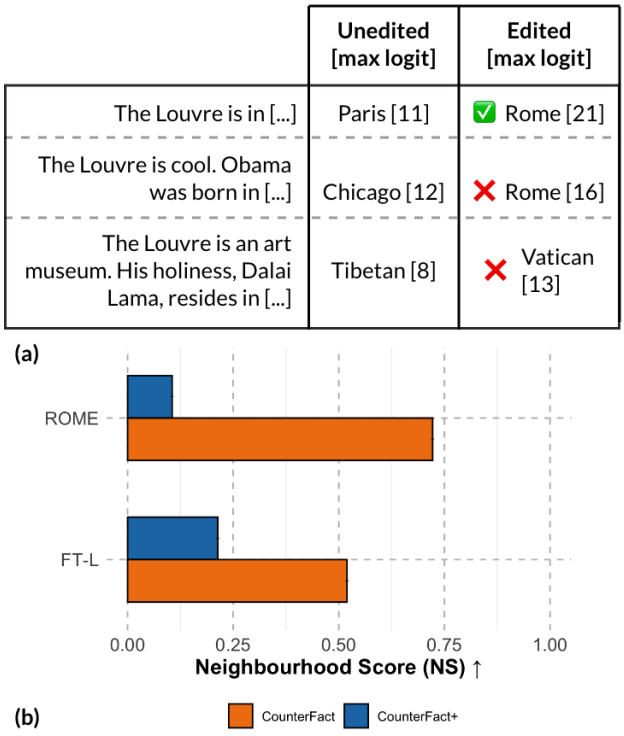

Figure 1: Unintended side effects of model edits and how to measure them. (a) GPT-2-medium is edited using ROME to counter-factually associate the Louvre's location with Rome. However, this results in unintended associations ("loud facts") like the association of Obama with Rome, suggesting low specificity of the edit. The edit also significantly increases the maximum logit (shown in brackets), suggesting that the edit is not merely replacing "Paris" with "Rome" in the desired contexts. (b) Measuring specificity by the fraction of correctly completed test prompts (COUNTERFACT) suggests a high specificity for ROME. Prepending the edit prompt (like "The Louvre is in Rome.") to each test prompt (COUNTERFACT+) results in a significant drop in performance. A significant drop in measured specificity can also be observed if the model edit is implemented using constrained fine-tuning (FT-L).

[*] Equal contribution.
Correspondence: fazl@robots.ox.ac.uk

Findings of the Association for Computational Linguistics: ACL 2023, pages 11548–11559
July 9-14, 2023

Model editing updates the parameters of a trained model in order to change its predicted probability distributions without retraining the entire model. This can be used to edit the associations that the model has memorized and hence, improve the accuracy of the model. Fig. 1 shows the example of a counter-factual model edit using ROME (Meng et al., 2022a) where the location of the Louvre is edited to be Rome instead of Paris. We use a counter-factual example since it makes it more evident that the new association is an effect of the model edit instead of the model training. Note that the examples in Fig. 1 are not taken from the COUNTERFACT+ dataset introduced below, but serve to intuitively illustrate the model editing failure modes we are interested in.

An important desideratum for model editing is specificity. Specificity captures how well the effect of the model edit is localized; in other words, specificity measures the absence of unintended side effects of model edits. Fig. 1 shows two examples of unintended side effects of ROME model editing, which we collectively call the problem of "loud facts". In the first example, mentioning "Louvre" (the subject of the model edit) leads the edited model to also complete unrelated test prompts ("Obama was born in") with "Rome" (the object of the model edit). In the second example, mentioning "Louvre" boosts the logits for words semantically related to "Rome", like "Vatican".

The existing specificity benchmark for model editing from the COUNTERFACT dataset (Meng et al., 2022a) suffers from two limitations which can be illustrated using these examples. First, COUNTERFACT *does not prompt the model in a way that is likely to surface unwanted side effects*. As demonstrated by the examples in Fig. 1, mentioning the subject of the model edit can drastically change the behavior of the edited model, but the existing benchmark does not detect this. Second, COUNTERFACT *considers only the probabilities for the original and edited object token* ("Paris" and "Rome"). As shown by the last example in Fig. 1, the edited model displays strongly changed logits not only for the original object ("Paris") and edit object ("Rome") but also for semantically related tokens ("Vatican"). Again, this would be overlooked by the current specificity evaluation since it does not consider the entire probability distribution.

These limitations mean that side effects of edits may be overlooked and specificity overestimated. Our main contributions are:

- COUNTERFACT+, a dynamic specificity benchmark, which adapts to the model edit under test, and is more sensitive than the existing benchmark.
- Neighborhood KL divergence (NKL), a specificity metric based on the full probability distribution instead of the currently used metrics which focus only on the tokens directly implicated in the model edit.
- Using COUNTERFACT+ and NKL, we show that ROME and MEMIT suffer from previously undisclosed problems with specificity.

2 Related work

Model editing. Several studies have sought to localize and modify the computation of knowledge within transformers. Geva et al. (2021) proposed that the multilayer perceptron (MLP) layers in a transformer can act as key–value memories of entities and information associated with that entity. Dai et al. (2022) then demonstrated a method to edit knowledge within BERT by writing the embedding of the object into certain rows of the MLP matrix. They identified important neurons for knowledge via gradient-based attributions. De Cao et al. (2021) presented a hyper-network to predict weight updates at test time, which can alter a fact. They tested both BERT and BART (Lewis et al., 2020) and focused on models fine-tuned for question answering. Mitchell et al. (2022) introduced a hyper-network method that learns to transform the decomposed terms of the gradient in order to efficiently predict a knowledge update and demonstrate the ability to scale up to large models such as T5 (Raffel et al., 2020), and GPT-J (Wang and Komatsuzaki, 2021). Finally, Meng et al. (2022a) introduced Rank-One-Model-Editing (ROME) which allows edits of transformer models via a rank-one modification of a single MLP layer. (Meng et al., 2022b) extended ROME to MEMIT (Mass-Editing Memory in a Transformer): MEMIT spreads the modification over multiple MLP layers; crucially, this enables thousands of simultaneous edits without performance degradation.

Model editing evaluation Benchmarks of model editing techniques for LLMs build on existing work on knowledge extraction from LLMs (see below). zsRE question answering was used for benchmarking model editing in (De Cao et al., 2021) and

(Mitchell et al., 2022). Elazar et al. (2021) introduced ParaRel, a curated dataset of paraphrased prompts and facts. Meng et al. (2022a) use this as a basis for constructing COUNTERFACT, which enables fine-grained measurements of knowledge extraction and editing along multiple dimensions, including specificity.

Knowledge extraction from LLMs. The assessment of knowledge within language models (LMs) has typically been done by evaluating whether the model is able to predict pieces of knowledge; Petroni et al. (2019, 2020) defined a fill-in-the-blank prompt and asked the LM to complete it. Subsequent work has demonstrated that knowledge extraction can be improved by diversifying the prompts (Jiang et al., 2020; Zhong et al., 2021), or by fine-tuning a model on open-domain textual facts (Roberts et al., 2020). However, constructing prompts from supervised knowledge extraction data is still prone to learning new knowledge instead of recalling existing knowledge in an LM (Zhong et al., 2021).

3 Experimental Setup

3.1 Dataset

We investigate the specificity of recent model editing techniques using the COUNTERFACT benchmark introduced in (Meng et al., 2022a). COUNTERFACT is a collection of 21,919 non-factual statements of the form (subject, relation, object) (s, r, o^*), which have low probabilities prior to the model edit. For each of these non-factual statements, we perform a model edit targeting this specific statement. To measure specificity, we then check whether any other associations in the model change in undesired ways. COUNTERFACT supports this check by providing a set of so-called neighborhood prompts for every non-factual statement used in the model edit. These neighborhood prompts are constructed as follows: For a model edit of the form $(s, r, o^c) \rightarrow (s, r, o^*)$ (where o^c is the correct object, and o^* is the false, counterfactual object), COUNTERFACT samples a set of nearby subjects s_n for which (s_n, r, o^c) holds true. Neighborhood prompts are then paraphrases of the collected (s_n, r).

Suppose, for example, the edit request was *(Darrieux, mother_tongue, French)* $\rightarrow$ *(Darrieux, mother_tongue, English)*. COUNTERFACT takes the relation and object from the edit request *(mother_tongue, French)*, samples true factual associations for this relation, object pair; e.g., *(Montesquieu, mother_tongue, French)* and then samples a random paraphrase, such as "The native language of Montesquieu is". These neighborhood prompts can be used to inspect whether the model edit has undesired side effects on closely related factual associations. See appendix C for a sample from the COUNTERFACT dataset, including the full set of neighborhood prompts.

Motivated by the example of loud facts shown in Fig. 1 and by the intuition that unwanted side effects are more likely when the model is primed with the linguistic context of the model edit, we now introduce a dynamic version of COUNTERFACT which we will refer to as COUNTERFACT+. To obtain COUNTERFACT+, we modify the neighborhood prompt by prepending the model edit. For example, if the original prompt is "The native language of Montesquieu is" the modified prompt would be "The mother tongue of Danielle Darrieux is English. The native language of Montesquieu is". See appendix D for a sample of the modified neighborhood prompts used for COUNTERFACT+.

To understand why we call COUNTERFACT+ a dynamic version of COUNTERFACT consider how either dataset would be applied to evaluate the success of a model edit: In both cases, we would need to identify the set $\mathcal{N}$ of neighborhood prompts in the dataset that are semantically closest to the intended model edit. But in COUNTERFACT, we would use $\mathcal{N}$ as is, whereas in COUNTERFACT+ we would change every prompt in $\mathcal{N}$ as a function of the model edit, as described above.

3.2 Metrics

To evaluate the specificity of a model edit on COUNTERFACT, Meng et al. (2022a,b) use two metrics, called Neighborhood Score and Neighborhood Magnitude. Denoting the post-edit probabilities for the correct token o^c and incorrect edit token o^* by $P^*(o^c)$ and $P^*(o^*)$, respectively, these are defined as follows: The Neighborhood Score (NS) is defined as the fraction of neighborhood prompts for which $P^*(o^c) > P^*(o^*)$. The Neighbourhood Magnitude (NM) is defined as $P^*(o^c) - P^*(o^*)$, the difference in probability assigned to the correct token versus the incorrect edit token. High NS and NM indicate that the edit has small unwanted side effects.

NS and NM, however, do not detect cases where the model edit significantly changes the predicted

probability for tokens other than o^c and o^*, such as in the last example in Fig. 1. To capture this possibility, we introduce as an additional metric the *Kullback–Leibler* (KL) divergence of the next-token distribution between the edited and unedited model, referred to as Neighborhood KL Divergence (NKL). Abbreviating the next token probability distribution for the unedited and edited models by $P(w)$ and $P^*(w)$, respectively, and denoting the token vocabulatory by $\mathcal{W}$, NKL is defined as KL divergence between $P(w)$ and $P^*(w)$:

$$\text{NKL} \stackrel{\text{def}}{=} \sum_{w \in \mathcal{W}} P(w) \log \left(\frac{P(w)}{P^*(w)} \right) \quad (1)$$

A large NKL is undesirable because it implies that the next-token probability distribution for neighborhood prompts has been strongly affected by the model edit.

3.3 Models and Model Editing Algorithms

We use GPT-2-medium (355M parameters), GPT-2-XL (1.5B) (Radford et al., 2019), and GPT-J (6B) (Wang and Komatsuzaki, 2021) to evaluate the following model editing methods:

- ROME (Rank-One-Model-Editing) performs a rank-one update of a single MLP layer to implement the edit (Meng et al., 2022a).
- MEMIT (Mass-Editing Memory in a Transformer) extends ROME to updates across several MLP layers (Meng et al., 2022b). Note that we do not test using multiple simultaneous edits.
- FT-L: Fine-Tuning with an L_∞ norm constraint (Zhu et al., 2020), constrained to a single layer, as described in (Meng et al., 2022a). We use FT-L as a simple baseline.

4 Results

Figure 2 shows the results for the ROME, MEMIT, and FT-L editing algorithms applied to the GPT-J (6B) model for different specificity metrics and datasets considered in this work. When evaluated using the Neighborhood Score (Fig. 2, top), we observe significant drops in specificity for all editing algorithms when going from CounterFact to CounterFact+. Note that specificity measured on the unedited model (GPT-J (6B)) also drops suggesting that there is confounding from the test prompts in CounterFact+, potentially due to recency bias (Zhao et al., 2021). The drop in specificity is much more pronounced for ROME and MEMIT, compared to FT-L and the unedited model, however. This shows that:

- ROME and MEMIT have undesired side effects which are not detected by CounterFact
- the improved benchmark CounterFact+ is able to detect these unwanted side effects

When evaluating specificity using the newly introduced Neighborhood KL Divergence (Fig. 2, bottom), we observe a large spike in divergence for both ROME and MEMIT when going from CounterFact to CounterFact+. FT-L shows a much smaller increase in divergence from CounterFact to CounterFact+. Figure 3 in the appendix shows the results on CounterFact and CounterFact+ for the NM metric.

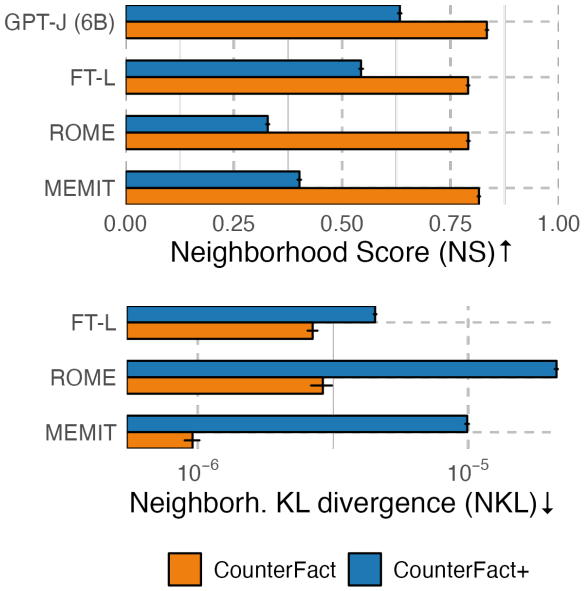

Figure 2: Comparison of model editing specificity benchmarks CounterFact and CounterFact+ on different model editing algorithms. Error bars show 99% confidence intervals.
(top) NS, the average fraction of correctly completed neighborhood test prompts after the model edit (larger is better). We see that CounterFact+ is a much more challenging specificity benchmark: Success rates NS on it range from 33% to 54% across different editing algorithms while they are close to 80% for CounterFact.
(bottom) NKL, the KL divergence of the next-token probability distribution of the edited model from that of the unedited model, averaged over all neighborhood test prompts. A lower value indicates higher specificity (the edited model behaves more like the unedited model).

Results across all three models are shown in tables 1 to 3. These tables list the mean scores on COUNTERFACT and COUNTERFACT+ for the Neighborhood Score (NS), Neighborhood Magnitude (NM), and Neighborhood KL divergence (NKL), respectively. The brackets give upper and lower bound of 99% confidence intervals obtained via bootstrap resampling (N=1,000). The bold values indicate the best score among the model editing algorithms for a given base model and dataset (excluding the unedited base model). Note how the method with the highest measured specificity switches from MEMIT/ROME to FT-L when going from COUNTERFACT to COUNTERFACT+.

NS ↑	COUNTERFACT	COUNTERFACT+
GPT-2 M	0.75 (0.749, 0.757)	0.46 (0.452, 0.463)
FT-L	0.52 (0.515, 0.524)	**0.21** (0.209, 0.217)
ROME	**0.72** (0.718, 0.726)	0.11 (0.102, 0.108)
GPT-2 XL	0.78 (0.780, 0.788)	0.52 (0.519, 0.530)
FT-L	0.71 (0.702, 0.711)	**0.38** (0.375, 0.385)
ROME	0.76 (0.755, 0.763)	0.14 (0.135, 0.142)
MEMIT	**0.77** (0.770, 0.778)	0.32 (0.314, 0.324)
GPT-J (6B)	0.83 (0.830, 0.839)	0.63 (0.628, 0.639)
FT-L	0.79 (0.786, 0.795)	**0.54** (0.538, 0.550)
ROME	0.79 (0.786, 0.796)	0.33 (0.323, 0.333)
MEMIT	**0.82** (0.811, 0.820)	0.40 (0.395, 0.407)

Table 1: Neighborhood Score NS (μ & 99% CI) on COUNTERFACT and COUNTERFACT+.

NM ↑	COUNTERFACT	COUNTERFACT+
GPT-2 M	0.04 (0.035, 0.037)	0.04 (0.038, 0.042)
FT-L	-0.02 (-0.019, -0.014)	**-0.11** (-0.112, -0.106)
ROME	**0.03** (0.028, 0.030)	-0.32 (-0.324, -0.317)
GPT-2 XL	0.05 (0.049, 0.052)	0.08 (0.073, 0.078)
FT-L	0.03 (0.033, 0.037)	**0.01** (0.012, 0.018)
ROME	0.04 (0.042, 0.045)	-0.38 (-0.384, -0.375)
MEMIT	**0.05** (0.048, 0.050)	-0.06 (-0.059, -0.052)
GPT-J (6B)	0.07 (0.073, 0.077)	0.11 (0.111, 0.117)
FT-L	**0.07** (0.068, 0.072)	**0.09** (0.090, 0.096)
ROME	0.05 (0.051, 0.056)	-0.12 (-0.127, -0.117)
MEMIT	**0.07** (0.066, 0.070)	-0.02 (-0.025, -0.017)

Table 2: Neighborhood Magnitude NM (μ & 99% CI) on COUNTERFACT and COUNTERFACT+.

The results from tables 1 to 3 show that the significant drop in specificity when evaluating on

NKL ↓	COUNTERFACT	COUNTERFACT+
GPT-2 M		
FT-L	1.4e-05 (1.3, 1.4)	**1.4e-05** (1.3, 1.4)
ROME	**1.6e-06** (1.4, 1.7)	2.5e-05 (2.5, 2.5)
GPT-2 XL		
FT-L	7.2e-06 (6.9, 7.4)	9.5e-06 (9.3, 9.7)
ROME	1.5e-06 (1.4, 1.6)	3.3e-05 (3.2, 3.3)
MEMIT	**2.9e-07** (2.5, 3.4)	**9.0e-06** (8.8, 9.1)
GPT-J (6B)		
FT-L	3.2e-06 (3.1, 3.4)	**5.2e-06** (5.1, 5.3)
ROME	3.5e-06 (3.2, 3.8)	1.8e-05 (1.8, 1.9)
MEMIT	**9.2e-07** (8.0, 10)	9.9e-06 (9.8, 10)

Table 3: Neighborhood KL Divergence NKL (μ & 99% CI) on COUNTERFACT and COUNTERFACT+. Note that the order of magnitude is suppressed for the confidence interval for visual clarity; it is the same as for the mean.

COUNTERFACT+ (compared to COUNTERFACT) holds across different model sizes and is not an artefact of using a particular model. Section B in the appendix discusses the scaling of specificity with model size in more detail.

5 Conclusion

Model editing techniques for auto-regressive transformers exhibit unreported issues related to specificity. Although our fine-tuning baseline, FT-L, exhibits less vulnerability to these issues than ROME and MEMIT, it falls short in competing with them regarding crucial model editing metrics such as robustness to paraphrasing (Meng et al., 2022a,b). This indicates that model editing still presents numerous complexities that require future attention.

Additionally, we revealed that the existing COUNTERFACT benchmark fails to detect the low specificity in ROME and MEMIT. To address this limitation, our primary contributions include:

- COUNTERFACT+, a dynamic specificity benchmark, which adapts to the model edit under test, and is more sensitive than the existing benchmark
- Neighborhood KL divergence (NKL), a specificity metric based on the full probability distribution as a complement to the currently used metrics which focus only on the tokens directly implicated in the model edit.

Limitations

The main limitation of the approach we took for improving model editing benchmarks is that it is ultimately based on manual inspection of test cases to understand the failure modes of model editing methods. This approach is not scalable and has a significant cost in terms of time and effort. As far as the specific benchmark we propose is concerned, more research is needed to assess its effectiveness for more complex scenarios such as dialogue and multi-turn conversations. We also have not investigated the application of our benchmark to scenarios in which multiple model edits are performed simultaneously. Furthermore, we do not evaluate other types of model edits, such as parameter pruning, and transfer learning. Future work should focus on developing methods that measure and quantify the effects of model edits on long-term aspects of language models, such as their ability to capture discourse structure and fluency of generated text. This could include corpus-level analysis and dynamic approaches like red-teaming or dynamic benchmarking to uncover subtle adverse effects.

Ethics Statement

We do not perform human experiments or evaluation.

We are aware of the potential risks posed by autoregressive transformer models, such as the capabilities to generate and manipulate text for harmful purposes.

Our dataset and evaluation code is open-sourced,[1] and we provide a homepage with interactive examples.[2]

Acknowledgements

First versions of the experiments reported here were performed during Apart Research's Interpretability Hackathon. We thank Jochem Hölscher for collaborating on early experiments during the hackathon, and Neel Nanda and Shay B. Cohen for insightful discussions and comments.

Our evaluation code builds directly on the MEMIT (Meng et al., 2022b) code.[3]

[1] https://github.com/apartresearch/specificityplus

[2] https://specificityplus.apartresearch.com/

[3] https://github.com/kmeng01/memit

References

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and Improving Consistency in Pretrained Language Models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. Fast model editing at scale. In *International Conference on Learning Representations*.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

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

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models.

A Neighborhood magnitude

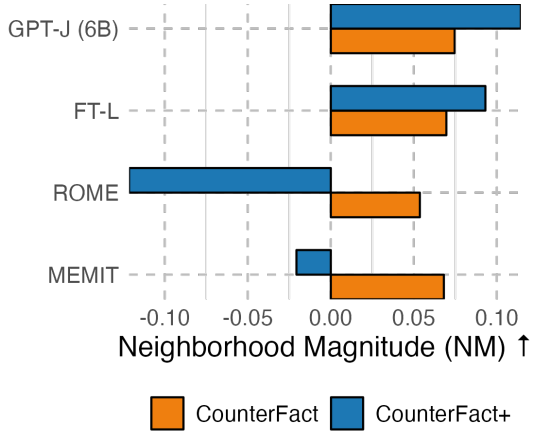

Figure 3: Comparison of model editing specificity benchmarks COUNTERFACT and COUNTERFACT+ evaluated using the Neighborhood Magnitude (NM) metric. NM measures the difference in probability of the correct token and the edit token. ROME retains almost the performance of the unedited model (GPT-J-6B) when evaluated on COUNTERFACT but shows a large drop in specificity when evaluated on COUNTERFACT+. MEMIT also shows significantly lower performance on COUNTERFACT+ than on COUNTERFACT, albeit less dramatic than for ROME.

B Scaling with model size

Figures 4 to 6 show how performance on the COUNTERFACT+ dataset scales with the size of the underlying model. The data shows that the drop in specificity when going to COUNTERFACT+ persists up to GPT-J (6B). While the data does not allow conclusive statements there is preliminary evidence that specificity of the edited models improves for larger models. This is, however, partially confounded by improved specificity of the unedited model. It is therefore, at this point, not clear whether the specificity problems of ROME and MEMIT would disappear completely in the limit of extremely large models.

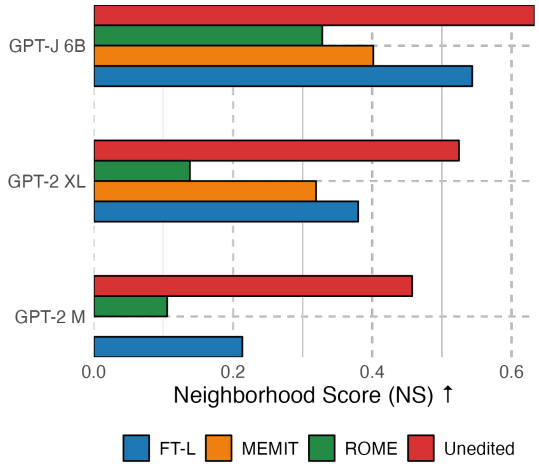

Figure 4: Evaluation of the model editing specificity benchmark COUNTERFACT+ on different model editing algorithms across model sizes. measured using NS, the average fraction of successfully completed neighborhood test prompts after the model edit. Larger values are better.

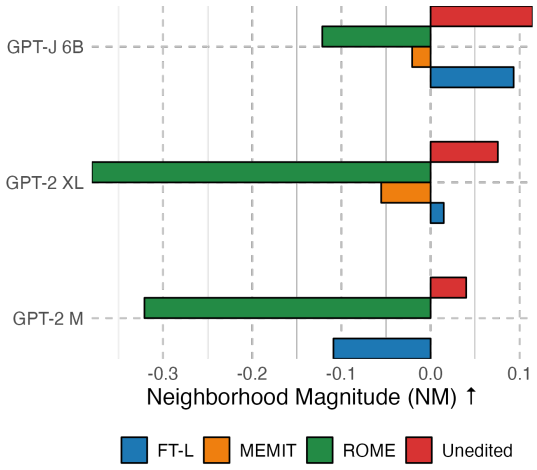

Figure 5: Evaluation of the model editing specificity benchmark COUNTERFACT+ on different model editing algorithms across model sizes. measured using NM, the difference in probability of the correct token and the edit token. Larger values are better.

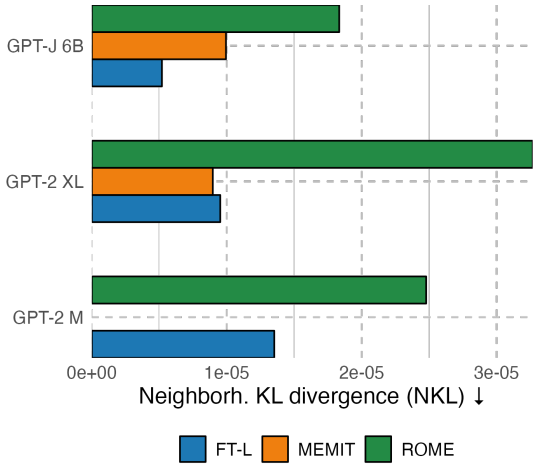

Figure 6: Evaluation of the model editing specificity benchmark COUNTERFACT+ on different model editing algorithms across model sizes. measured using NKL, the KL divergence of the next-token probability distribution of the edited model from that of the unedited model, averaged over all neighborhood test prompts. A lower value indicates higher specificity (the edited model behaves more like the unedited model).

C COUNTERFACT dataset sample

The listing below shows an example entry of the COUNTERFACT dataset. The "`requested_rewrite`" defines the model edit. The "`neighborhood_prompts`" are used for measuring specificity.

A sample of the COUNTERFACT dataset.


```
{
  "case_id": 0,
  "pararel_idx": 2796,
  "requested_rewrite": {
    "prompt": "The mother tongue of {} is",
    "relation_id": "P103",
    "target_new": {"str": "English", "id": "Q1860"},
    "target_true": {"str": "French", "id": "Q150"},
    "subject": "Danielle Darrieux"
  },
  "paraphrase_prompts": [
    "Shayna does this and Yossel goes still and dies. Danielle Darrieux,
    a native", "An album was recorded for Capitol Nashville but never released.
    Danielle Darrieux spoke the language"
  ],
  "neighborhood_prompts": [
    "The mother tongue of L\u00e9on Blum is",
    "The native language of Montesquieu is",
    "Fran\u00e7ois Bayrou, a native",
    "The native language of Raymond Barre is",
    "Michel Rocard is a native speaker of",
    "Jacques Chaban-Delmas is a native speaker of",
    "The native language of Fran\u00e7ois Bayrou is",
    "Maurice Genevoix, speaker of",
    "The mother tongue of Fran\u00e7ois Bayrou is",
    "Melchior de Vog\u00fc\u00e9, speaker of"
  ],
  "attribute_prompts": [
    "J.\u00a0R.\u00a0R. Tolkien is a native speaker of",
    "The mother tongue of Douglas Adams is",
    "...",
    "Meryl Streep, a native",
    "George Orwell spoke the language",
    "George Washington, a native", "Michael Jackson, a native"
  ],
  "generation_prompts": [
    "Danielle Darrieux's mother tongue is",
    "Where Danielle Darrieux is from, people speak the language of",
    "Danielle Darrieux's mother tongue is",
    "...",
    "Danielle Darrieux was born in"
  ]
}
```

D COUNTERFACT+ dataset sample

The listing below shows a sample of the COUNTERFACT+ dataset, which is derived from the sample shown above for COUNTERFACT. We only display the modified `neighborhood_prompts` which are obtained by prepending the model edit (`"The mother tongue of Danielle Darrieux is English."`) to the original neighborhood prompt.

A sample of the modified "neighborhood_prompts" in the CounterFact+ dataset.

```
[
  "The mother tongue of Danielle Darrieux is English. The mother tongue of
↪  L\u00e9on Blum is",
  "The mother tongue of Danielle Darrieux is English. The native language of
↪  Montesquieu is",
  "The mother tongue of Danielle Darrieux is English. Fran\u00e7ois Bayrou, a
↪  native",
  "The mother tongue of Danielle Darrieux is English. The native language of
↪  Raymond Barre is",
  "The mother tongue of Danielle Darrieux is English. Michel Rocard is a
↪  native speaker of",
  "The mother tongue of Danielle Darrieux is English. Jacques Chaban-Delmas is
↪  a native speaker of",
  "The mother tongue of Danielle Darrieux is English. The native language of
↪  Fran\u00e7ois Bayrou is",
  "The mother tongue of Danielle Darrieux is English. Maurice Genevoix,
↪  speaker of",
  "The mother tongue of Danielle Darrieux is English. The mother tongue of
↪  Fran\u00e7ois Bayrou is",
  "The mother tongue of Danielle Darrieux is English. Melchior de
↪  Vog\u00fc\u00e9, speaker of"
]
```

ACL 2023 Responsible NLP Checklist

A For every submission:

☐ A1. Did you describe the limitations of your work?
Left blank.

☐ A2. Did you discuss any potential risks of your work?
Left blank.

☐ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☐ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☐ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☐ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.