Uncovering Limits of Memory Editing in Large Language Models: A New Specificity Benchmark

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

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

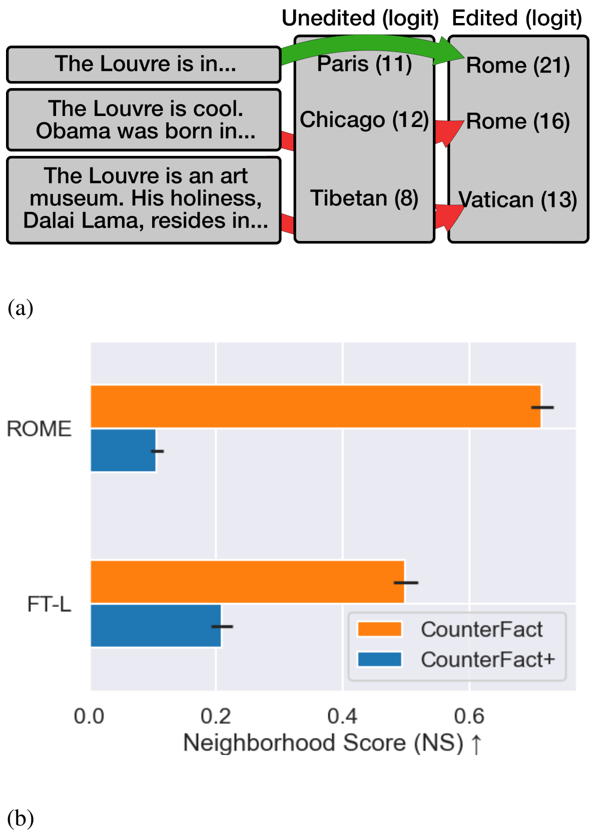
[1] Apart Research [2] Edinburgh Centre for Robotics
[3] Department of Engineering Sciences, University of Oxford

Abstract

Recent memory editing techniques promise to mitigate the problem of memorizing false or outdated associations during large language model (LLM) training.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 a more challenging, dynamic benchmark called 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 memory 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

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(LLMs) are powerful tools for generating human-like language, but they can, unfortunately, also memorize false or outdated associations, limiting their applicability.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



(a)

(b)

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. 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.

Memory editing updates the parameters of an already 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,

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

improve the accuracy of the model. Fig. 1 shows the example of a counter-factual model edit using ROME memory editing (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, for counter-factual cases, it is more evident that the new association is an effect of the model edit instead of the model training.

An important quality measure of memory editing is specificity. Specificity captures how well the effect of the memory edit is localized to the intended correction; in other words, specificity is a measure of the absence of unintended side effects of memory edits. Fig. 1 shows two examples of unintended side effects of ROME memory editing, which we collectively call the problem of "loud facts". In the first example, mentioning "Louvre" (the subject of the model edit) leads the 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" leads to a boost in logit on words semantically related to "Rome," like "Vatican".

Existing specificity benchmarks for memory editing suffer from two limitations which can be illustrated using these examples. First, these benchmarks do not prompt the model in a way that is likely to surface unwanted side effects. In the examples in Fig. 1, mentioning the subject of the model edit can drastically change the behavior of the edited model, but existing benchmarks do not do this. Second, existing specificity benchmarks consider only the probabilities for the original and edited object token ("Paris" and "Rome"). In the last example in Fig. 1, the edited model displays strongly changed logits for words that are neither the original object ("Paris") nor the edit object ("Rome") but are semantically related to the edit object ("Vatican"). Again, this would be overlooked by current specificity evaluations because they do not consider the entire predicted probability distribution and therefore miss most of the signal. These limitations make it difficult to detect unwanted side effects of memory editing, meaning that specificity will be overestimated.

2 Related work

Memory 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 masked language model (LM) 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 identify important neurons for knowledge via gradient-based attributions. De Cao et al. (2021) then present a hyper-network to predict weight updates at test time, which can alter a fact. They test both BERT and BART (Lewis et al., 2020) and focus on models fine-tuned for question answering. Finally, 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).

Memory editing evaluation Benchmarks of memory editing techniques for LLMs build on existing work on knowledge extraction from LLMs (see below). zsRE question answering was used for benchmarking memory 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 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 memory 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. We use the first 1531 entries of CounterFact due to computational constraints. 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). See section C for a sample from the CounterFact dataset, including the full set of neighborhood prompts.

Suppose the edit request was *(Danielle Darrieux, mother_tongue_is, French)* → *(Danielle Darrieux, mother_tongue_is, English)*. CounterFact takes the relation and object from the edit request *(mother_tongue_is, French)* and samples true factual associations for this relation, object pair; e.g., *(Montesquieu, native_language_is, French)* paraphrased as "The native language of Montesquieu is". Neighborhood prompts can be used to inspect whether the model edit has undesired side effects on closely related factual associations.

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 section D for a sample of the modified neighborhood prompts used for CounterFact+.

3.2 Metrics

To evaluate the performance 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 implies that the edit appears to have worked well.

NS and NM, however, do not detect cases where the model edit leads to significant changes in the predicted probability distribution on 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 on the next-token distribution between the edited and unedited model. We refer to this metric as the Neighborhood KL Divergence (NKL). A larger NKL implies that the output distribution has been affected and is undesirable.

3.3 Models and Memory Editing Algorithms

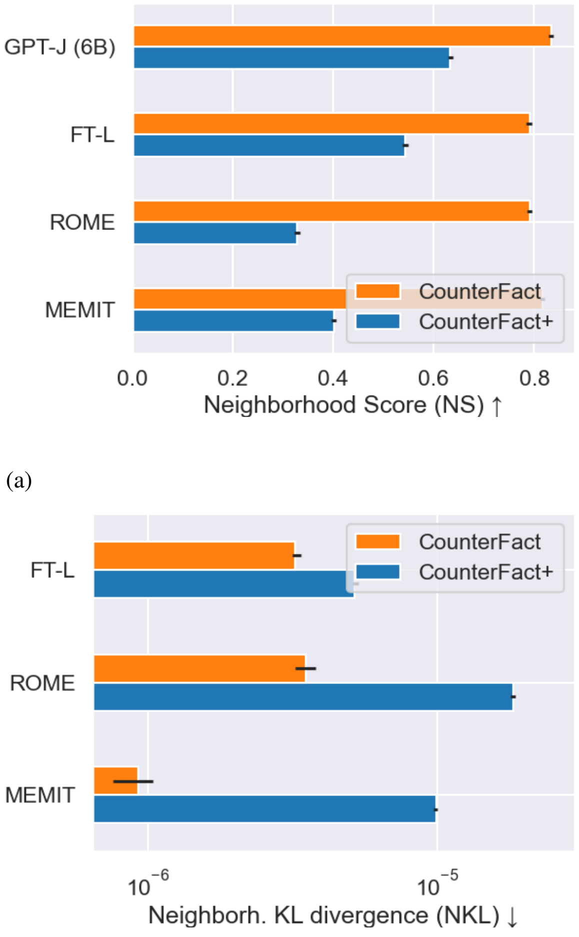
We run our experiments on GPT-2-medium (355M parameters) (Radford et al., 2019). The model editing methods which are evaluated are ROME (Rank-One-Model-Editing) (Meng et al., 2022a) and constrained Fine-Tuning (FT-L) with an L_∞ norm constraint (Zhu et al., 2020), as described in Meng et al. (2022a).

add other models i think all of them should fit now

add MEMIT

4 Results

Figure 2 shows the results for GPT-2-medium and for the ROME and FT-L editing for different specificity metrics and datasets considered in this work. When evaluated using the Neighborhood Score (2a), we observe significant drops in performance across both ROME and FT-L when going from CounterFact to CounterFact+. This shows that (a) both ROME as well as FT-L have undesired side effects and that (b) the improved benchmark CounterFact+ is able to detect some of those unwanted side effects which are not detected by CounterFact as originally defined. When evaluating specificity using the newly introduced Neighborhood KL Divergence (2b), we observe a large spike in divergence for ROME when going from CounterFact to CounterFact+. Interestingly, FT-L shows a slight decrease in divergence from CounterFact to CounterFact+, albeit from an already high baseline. Figure 3 in the appendix shows the results on CounterFact and CounterFact+ for the NM metric. We provide a table with the full results in appendix A.



(a)

(b)

Figure 2: Comparison of memory editing specificity benchmarks CounterFact and CounterFact+ on different model editing algorithms. a shows NS, the average fraction of successfully completed neighborhood test prompts after the model edit. b shows 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). We see that CounterFact+ is a much more challenging specificity benchmark: Success rates NS on it are below 20% across different editing algorithms while they approach 80% for CounterFact. We also see that KL-divergence-based measure NKL adds valuable information: On the CounterFact dataset, FT-L displays a much higher NKL (low specificity) while NS remains moderately high. Error bars show 99% confidence intervals.

5 Conclusion

We have shown that the ROME, MEMIT, and FT-L memory editing techniques for autoregressive transformers suffer from problems with specificity, suggesting that memory editing is far from being solved. We have also shown that for ROME, low specificity is only detected using CounterFact+, a new specificity benchmark introduced in this paper. Our main contributions are:

- a dynamic specificity benchmark, which adapts to the model edit under test, and is more sensitive than the existing static benchmarks
- 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.

Limitations

The main limitation of the new specificity benchmark is that it relies on the manual inspection of test cases to understand the failure modes of memory editing methods. This is not scalable and it has a significant cost in terms of time and effort. Additionally, the benchmark is limited in terms of the number of models and methods that it can evaluate, and more research is needed to assess the effectiveness of the benchmark for more complex scenarios such as dialogue and multi-turn conversations. Furthermore, we do not consider the effects of other types of model edits, such as parameter pruning, and transfer learning. Finally, the benchmark does not consider the effect of data quality and data sparsity when measuring the impact of model edits on performance. We also do not consider the extension of CounterFact+ to multiple, simultaneous edits as demonstrated in MEMIT (Meng et al., 2022b). Future work should focus on developing methods that measure and quantify the effects of memory 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. We plan to release our code and dataset used to edit models described in this paper upon publication.

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`.

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 Full Results

Tables 1, 2, and 3 show 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 highest score among the model editing algorithms (excluding the unedited GPT2-M model) for a given dataset.

	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)
MEMIT	nan (nan, nan)	nan (nan, nan)
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 (mean and 99% confidence interval) on CounterFact and CounterFact+.

	CounterFact	CounterFact+
GPT2-M	0.04 (0.03, 0.04)	0.04 (0.03, 0.05)
ROME	**0.03** (0.02, 0.03)	-0.33 (-0.34, -0.31)
FT-L	-0.02 (-0.03, -0.01)	**-0.11** (-0.13, -0.10)

Table 2: Neighborhood Magnitude (mean and 99% confidence interval) on CounterFact and CounterFact+.

	CounterFact	CounterFact+
GPT2-M	0 (0, 0)	0 (0, 0)
ROME	**1.70e-06** (1.19, 2.23)	2.50e-05 (2.40, 2.60)
FT-L	1.44e-05 (1.32, 1.54)	**1.34e-05** (1.29, 1.40)

Table 3: Neighborhood KL Divergence (mean and 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.

B Neigborhood magnitude

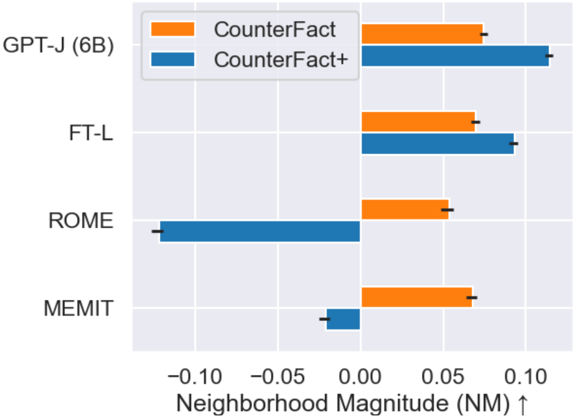


Figure 3: Comparison of memory editing specificity benchmarks CounterFact and CounterFact+ evaluated using the Neighborhood Magnitude (NM) metric. NM measures the difference in probability of the correct token and the edit token. ROME retains almost the performance of the unedited model (GPT-J-6B) when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

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"
]
```