A.5 Generation Examples

Below are some examples of generations produced by GPT-J when edited by ROME post catastrophic forgetting. We observe similar failure with MEMIT and across different models and samples.

Prompt	Generated Text
The Tonight Show with Jay Leno is my favorite show that has aired on	CBS CBS. CBS CBS. CBS CBS CBS CBS CBS CBS CBS CBS CBS CBS. CBS CBS. CBS CBS CBS CBS CBS CBS CBS CBS CBS CBS. CBS CBS CBS CBS CBS CBS. CBS CBS CBS. CBS CBS CBS CBS CBS CBS CBS CBS CBS. CBS CBS CBS CBS CBS CBS. CBS CBS CBS.
Toronto's twin city has famous tourist attractions including	the is, and Syrian aid„, and aid. And and, and and, and, which, is, [ily, and is and„ and and and [ily was, no, and, and„, and. no aidily, and Syriaily, do, no doubt de [and which no Syrian is do a, no., and the, is and is, and, and and„ [aid, and
The language used by people in Kentucky is	Finnish Finnish Finnish Finnish Finnish language Finnish Finnish, Finnish: Finnish: Finnish: Finnish: Finnish: Finnish. Finn - Finnish: Finnish. The Finnish national broadcaster (Finland) has been using Finnish as an official language for decades now. Finnish - Finland. Finland has had a long tradition of using Finnish as an official

Table 8: Text generated by GPT-J post the point of catastrophic forgetting when edited using ROME.

A.6 Background

In this section we will explain the details of four model editing algorithms explored in this paper: ROME (Meng et al., 2022a), MEMIT (Meng et al., 2022b), and MEND (Mitchell et al., 2021), and Fine-Tuning.

A.6.1 ROME

Building off the discovery that feed-forward layers of a transformer function as key-value memories (Geva et al., 2020), where neurons from $W_{fc}^{(l)}$ and $W_{proj}^{(l)}$ emulate keys and values respectively, (Meng et al., 2022a) hypothesize that insertion of new knowledge can take the form of some linear transformation W such that $WK \approx V$ where K and V are the vector of keys and values respectively. For an updated fact represented by the key-value pair (k_*, v_*), the constrained optimization problem can be summarized as follows

$$\min \|\hat{W}K - V\| \ni \hat{W}k_* = v_* \tag{1}$$

With the solution $\hat{W} = W + \Lambda(C^{-1}k_*)^\top$ where $C = KK^\top$ and $\Lambda = \frac{v_* - Wk_*}{(C^{-1}k_*)^\top k_*}$. The full derivation for the solution can be found in Appendix A in (Meng et al., 2022a). To find the optimal k_*, inputs x are taken where the subject s is represented in the last token. k_* is given by

$$k_* = \frac{1}{N}\sum_{j=1}^{N} k(x_j + s), \text{ where}$$
$$k(x) = \sigma(W_{fc}^{(l^*)}\gamma(a_{[x],i}^{(l^*)} + h_{[x],i}^{(l^*-1)})) \tag{2}$$

where l^* is the desired layer, i is the last subject token index, $h_{[x],i}^{(l^*-1)}$ is the hidden state of the previous layer, and $a_{[x],i}^{(l^*)}$ is the global attention of the hidden layer. Here, N is set to 50, since the average is taken over 50 sampled prefixes x_j. Optimal $v_* = \text{argmin}_z \mathcal{L}(z)$ where

$$\mathcal{L}(z) = \frac{1}{N}\sum_{j=1}^{N} -\log(\mathbb{P}_{G(m_i^{(l^*)}:=z)}[o^*|x_j + p])$$
$$+ D_{KL}(\mathbb{P}_{G(m_{i'}^{(l^*)}:=z)}[x|p']\|\mathbb{P}_G[x|p']) \tag{3}$$

z is a vector that is substituted as the i-th token of the output to the MLP layer that enables the desired change to be realized. $G()$ substitutes the specified hidden state with the modified version. p is the factual prompt, while p' is the factual prompt rewritten in a form that begins with the subject. Given these prompts, o^* is the new object. v_* is solved using an Adam optimizer with a learning rate of 0.5 and weight decay rate of 1.5×10^{-3}. Following this, we compute the updates to the MLP weights using equation 1. ROME updates weights for GPT2-XL and GPT-J at layers 18 and 6 respectively.

A.6.2 MEMIT

Rather than overburdening one layer with an update, (Meng et al., 2022b) introduces MEMIT as a means of distributing the impact of the update across multiple layers. In doing so, they are able to largely scale the number of edits they can reliably make. In order to express the update, we want to find some $z_i = h_i^L + \delta_i$ such that, when substituted in place of h_i^L at layer L, it is successful. We find this by optimizing δ_i using

$$z_i = h_i^L +$$
$$\text{argmin}_{\delta_i} \frac{1}{P}\sum_{j=1}^{P} -\log \mathbb{P}_{G(h_i^L += \delta_i)}[o_i|x_j \oplus p(s_i, r_i)] \tag{4}$$

for the desired edit object o_i and set of prompts $x_j \oplus p(s_i, r_i)$. Here, x_j is a set of prefixes and $p(s_i, r_i)$ is a prompt generated from the edit subject s_i and relation r_i. We want to find some update Δ^l for every layer $l \in R$ for a set of layers R so that

$$\hat{W}_{\text{out}}^l := W_{\text{out}}^l + \Delta^l \text{ for all } l \in R$$
$$\text{such that } \min_{\Delta^l} \sum_i \|z_i - \hat{h}_i^L\|^2 \tag{5}$$

$$\text{where } \hat{h}_i^L = h_i^0 + \sum_{l=1}^{L} a_i^l$$
$$+ \sum_{l=1}^{L} \hat{W}_{\text{out}}^l \sigma(W_{\text{in}}^l \gamma(h_t^{l-1})) \tag{6}$$

The closed form solution to this update is given by $\Delta^l = R^l K^{l\top}(C + K^l K^{l\top})^{-1}$. The full derivation can be found in (Meng et al., 2022b) section 4.2. To solve this, we need to find $K^l = [k_1^l, k_2^l, ..., k_n^l]$ and $R^l = [r_1^l, r_2^l, ..., r_n^l]$. This is found using

$$k_i^l = \frac{1}{P}\sum_{j=1}^{P} k(x_j + s_i),$$
$$\text{where } k(x) = \sigma(W_{\text{in}}^l \gamma(h_i^{l-1}(x))) \tag{7}$$