Supplemental Material

1 Memory Interfacing Detail

This section explains details on memory interacting operations described in the memory controller layer section.

Interface vector. The controller generates an interface vector $\mathbf{i}_t$ as

$$\mathbf{i}_t = W_i \mathbf{h}_t^m \in \mathbb{R}^{sq+5s+3q+3},$$

to control the memory $M \in \mathbb{R}^{p \times q}$ based on the hidden state of a recurrent layer. s represents the number of read heads. One can consider it as a concatenation of various functional vectors that determine the basic operations of the memory such as memory addressing, read, and write. The complete list of functions is described in Table 1.

Memory addressing. We use a content-based addressing mechanism for read and write operations. In content-based addressing, the memory locations required to read/write at the current time step t are obtained by using the probability distribution of the cosine similarity between the key and each memory as follows:

$$\mathbf{c}_{[i]} = \text{softmax}(\cos(M_{[i,:]}, \mathbf{k})\tilde{\alpha}),$$

where $\tilde{\alpha}$ is computed as $(1 + \log(1 + e^{\alpha}))$, which ranges in $[1, \infty)$.

Read operation. Multiple read heads individually perform a read operation by weighting over the entire memory. For each read head r, we calculate the accessing distribution as

$$\mathbf{c}^r_{t[i]} = \text{softmax}(\cos(M_{t[i,:]}, \mathbf{k}^r_t)\tilde{\alpha}^r_t).$$

Additionally, we follow the temporal memory linkage described in Graves et al. (2016), which associates the access order of memories in terms of writing contents, through the temporal link matrix $L_t \in \mathbb{R}^{p \times p}$. The multiplication of L_t and read weights from the previous time step gives the backward-accessing distribution $\mathbf{b}^r_t = L^T_t \mathbf{w}^r_{t-1}$ and the forward-accessing distribution $\mathbf{f}^r_t = L_t \mathbf{w}^r_{t-1}$, which helps to track the memory accessing order. The read weights are obtained from the linear combination of the corresponding weights, and the mode vectors $\tilde{\pi}$ are normalized by softmax, i.e.,

$$\mathbf{w}^r_t = \tilde{\pi}[0]\mathbf{b}^r_t + \tilde{\pi}[1]\mathbf{c}^r_t + \tilde{\pi}[2]\mathbf{f}^r_t.$$

Operation	Name	Vector
Read	key	$\{\mathbf{k}^{r,i}_t\}^s_{i=1} \in \mathbb{R}^q$
	strength	$\{\alpha^{r,i}_t\}^s_{i=1} \in \mathbb{R}$
	mode	$\{\pi^i_t\}^s_{i=1} \in \mathbb{R}^3$
Write	key	$\mathbf{k}^w_t \in \mathbb{R}^q$
	strength	$\alpha^w_t \in \mathbb{R}$
	erase vector	$\mathbf{e}_t \in \mathbb{R}^q$
	write vector	$\mathbf{v}_t \in \mathbb{R}^q$
	free gate	$\{g^{f,i}_t\}^s_{i=1} \in \mathbb{R}$
	allocate gate	$g^a_t \in \mathbb{R}$
	write gate	$g^w_t \in \mathbb{R}$

Table 1: Functional vector list that comprises the interface vector of the controller.

Finally, the read weight is applied to memory locations to get the final read vector as

$$\mathbf{m}^r_t = \sum_{i=1}^{p} M_{t[i,:]} \mathbf{w}^r_t[i].$$

Write operation. Similar to a read operation, a write head determines where to write by using content-based weighting. For each write head w, we calculate the accessing distribution as

$$\mathbf{c}^w_{t[i]} = \text{softmax}(cos(M_{t-1[i,:]}, \mathbf{k}^w_t)\tilde{\alpha}^w_t).$$

Also, we follow the dynamic memory allocation described in Graves et al. (2016) to track the memory allocation weights $\mathbf{a}_t \in \mathbb{R}^p$. $\mathbf{a}_t$ are calculated to indicate where to write, which is interpolated with content-based weights to get new locations for writing. The write gate g^w_t decides whether to write or not while the allocation gate g^a_t determines the degree of interpolation, i.e.,

$$\mathbf{w}^w_t = g^w_t[g^a_t \mathbf{a}_t + (1 - g^a_t)\mathbf{c}^w_t].$$

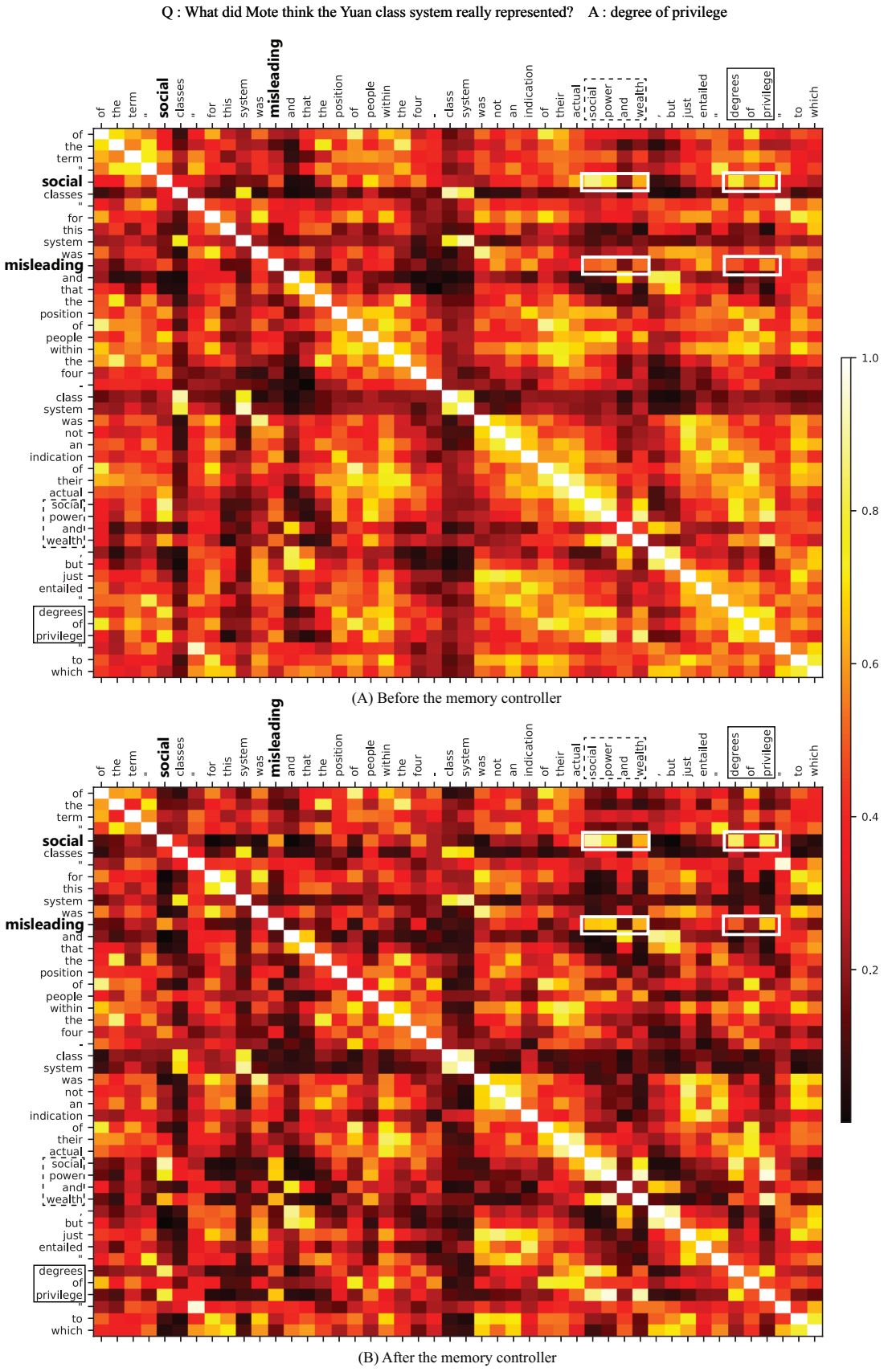
Then a write operation is performed by first erasing the write location with an erase vector $\mathbf{e}_t$ and writing to the location by using the write vector $\mathbf{v}_t$, i.e.,

$$M_t = M_{t-1} \odot (J_{p,q} - \mathbf{w}^w_t \mathbf{e}^T_t) + \mathbf{w}^w_t \mathbf{v}^T_t,$$

where $\odot$ is element-wise multiplication and $J_{p,q} \in \mathbb{R}^{p \times q}$ is the matrix with its elements being all ones.

References

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471.



(A) Before the memory controller

(B) After the memory controller

Figure : Pairwise cosine similarity maps on the output of the layer just before the memory controller (A) and that after the memory controller (B). The strong answer candidates for the question, 'social power and wealth' and 'degrees of privilege', are related to the keyword **'social'**. The final prediction can be made by the association with **'misleading'** because the association with 'social power and wealth' becomes stronger than that with 'degrees of privilege' after passing the memory controller.