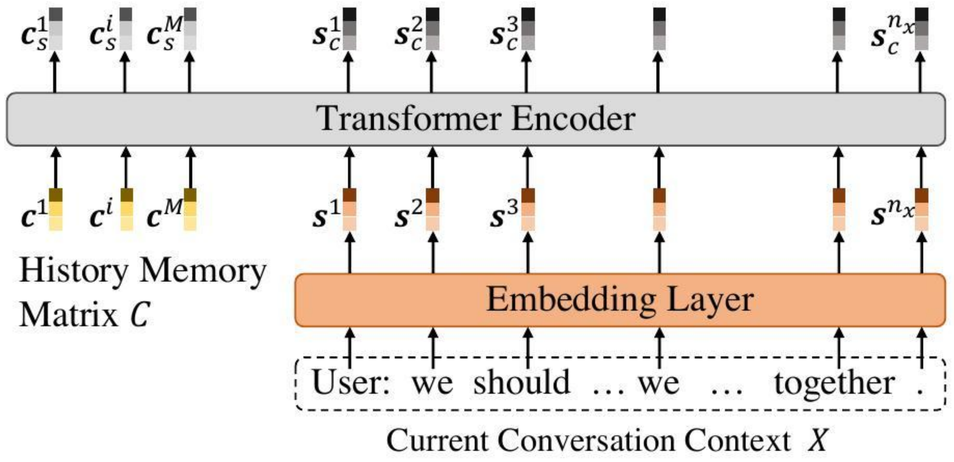

Figure 4: The structure of the history-aware context encoder in HAHT.

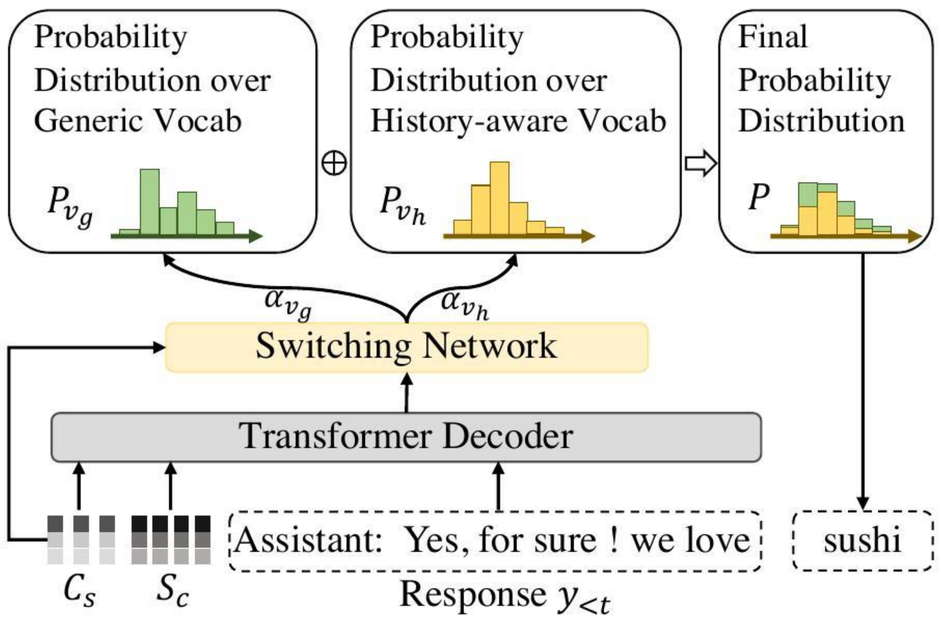

Figure 5: The structure of the history-aware response generator in HAHT.

conversation aggregator F_c to aggregate all utterance representations $\mathbf{U}^i$ into the condensed history memory $\mathbf{c}^i$,

$$\mathbf{c}^i = F_c(\mathbf{U}^i). \tag{2}$$

The conversation aggregator is developed based on the following self-attentive mechanism (Lin et al., 2017),

$$\begin{aligned} F_c(\mathbf{U}^i) &= \alpha \mathbf{U}^i, \\ \alpha &= \text{softmax}\big(\mathbf{W}_k \text{tanh}(\mathbf{W}_q \mathbf{U}^{i\top})\big), \end{aligned} \tag{3}$$

where $\mathbf{W}_q$ and $\mathbf{W}_k$ are learnable parameters. $\alpha \in \mathbb{R}^{n_i}$ is the importance vector of the history conversation utterances in H^i.

After applying previous steps to all history conversations H, we will finally obtain a history memory matrix $\mathbf{C} \in \mathbb{R}^{M \times d}$ containing a history memory for each history conversation, where M is the number of history conversation sessions.

3.2 History-aware Context Encoder

History conversation sessions usually contain the background stories (*e.g.,* interlocutors' profiles or previous discussions between them) that bring out the current conversation session. Leveraging the history conversations will help the model to better understand the current conversation context and respond properly. On the other hand, the current conversation context can help the model update the history memories. Thus, we encode the history memory $\mathbf{C}$ together with the current conversation context by adopting the transformer attention between them.

For the current conversation context X, we also prepend a special token "User:" or "Assistant:" to each utterance depending on the role of the utterance speaker and concatenate all utterances into a single sentence. Then, we adopt the embedding layer E_m to obtain a sequence of context token embeddings $\mathbf{S} = \{\mathbf{s}^1, \mathbf{s}^2, \cdots, \mathbf{s}^{n_x}\}$, where n_x is the length of the context sequence. Next, we concatenate the history memory matrix $\mathbf{C} \in \mathbb{R}^{M \times d}$ with $\mathbf{S} \in \mathbb{R}^{n_x \times d}$ over the first dimension and apply n_{enc} Transformer encoder layers.

By employing attention in the transformer encoder layers, our model can understand the conversation context by attending to all context token embeddings and history conversation memories. We denote this history-aware context encoding by $\mathbf{S}_c \in \mathbb{R}^{n_x \times d}$. After context encoding, history conversation memories are updated based on the latest information from the current conversation context. We denote this context-updated history memory as $\mathbf{C}_s \in \mathbb{R}^{M \times d}$. The concatenation of $\mathbf{C}_s$ and $\mathbf{S}_c$ over the first dimension will become the input of the response generator.

3.3 History-aware Response Generator

Inspired by CopyNet (Gu et al., 2016), we construct two vocabularies, *i.e.,* generic vocabulary V_g and history-aware vocabulary V_h, to better generate history-aware responses. The generic vocabulary V_g contains the words that appear in all the training dataset, and the history-aware vocabulary V_h only contain the words that appear in the history conversations H. To generate a word of the response, the response generator will choose to generate a generic word from V_g or directly copy a word from V_h based on the switching mechanism (Gulcehre et al., 2016).

Specifically, at each decoding time step t, we feed $\mathbf{C}_s$, $\mathbf{S}_c$ and the ground truth word sequence before t into n_{dec} Transformer decoder layers and obtain a hidden representation vector $\mathbf{o}_t \in \mathbb{R}^d$. The