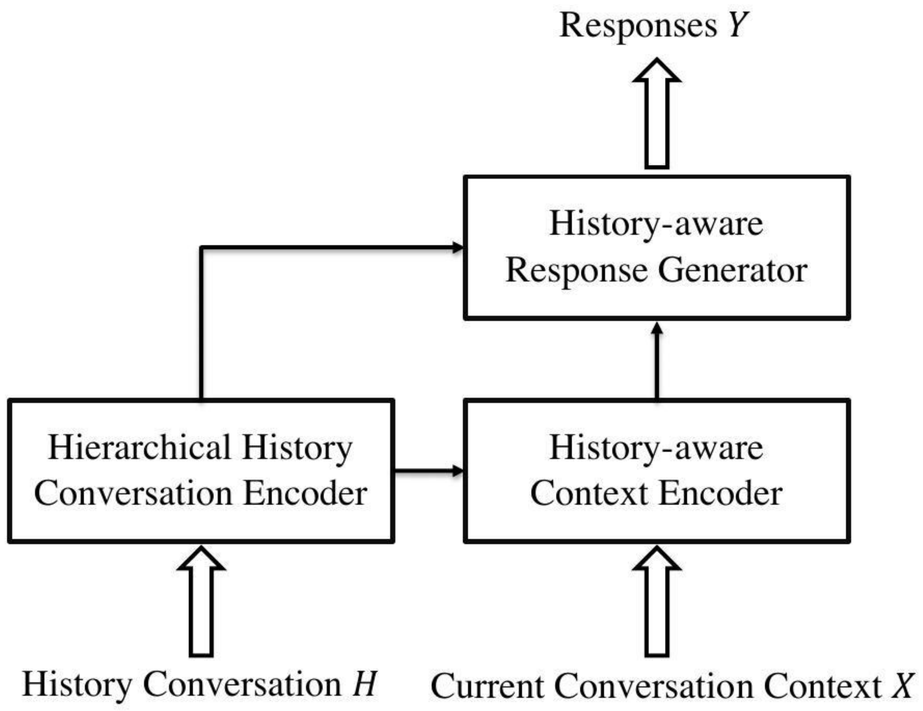

Figure 2: The overall structure of the proposed HAHT model, which contains 1) hierarchical history conversation encoder, 2) history-aware context encoder, and 3) history-aware response generator. The details of each component are shown in Figure 3, 4, 5, respectively.

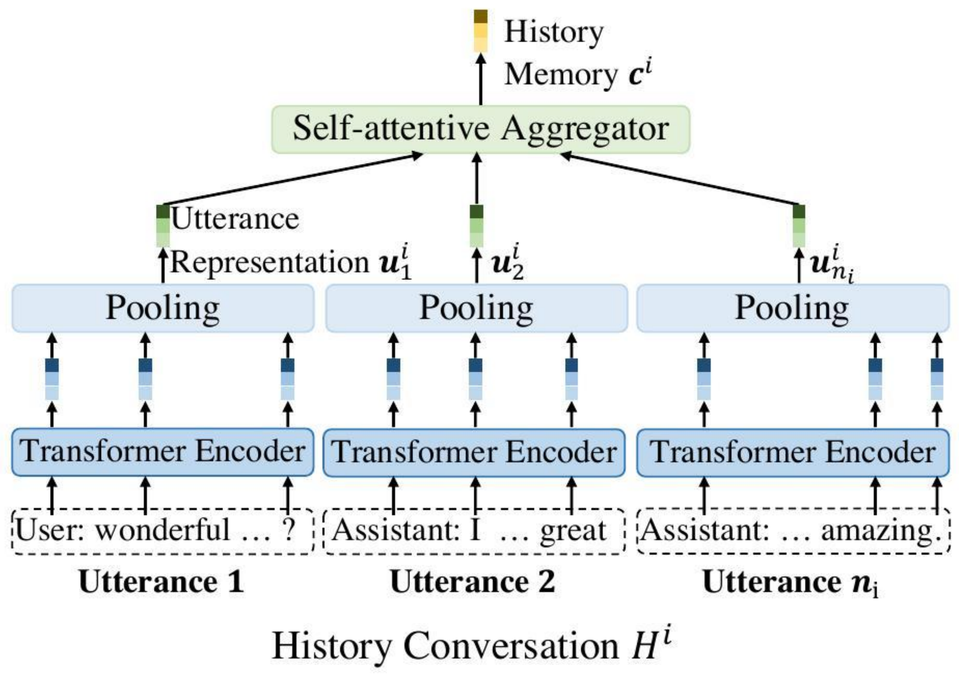

Figure 3: The structure of the hierarchical history conversation encoder in HAHT.

their methods need to retrieve a very large portion of history conversations to achieve better results than the standard Transformer. In addition, these models still need to concatenate the retrieved raw history conversation text with the current conversation context, yielding concatenations that are still much longer than the 128 token truncation lengths. Therefore, the incorporation of historical contexts in these methods is still limited by the short token truncation lengths of pre-trained models.

3 The Proposed Method

In general, a *Multi-Session Conversation (MSC)* consists of a current conversation session and several history conversation sessions that happen before the current one, all between the same two interlocutors. A multi-session open-domain dialogue system aims to generate natural, well-informed, and context-relevant responses to the user's utterances based on all history conversation sessions and the current conversation context.

Formally, we denote the MSC dataset D by a list of N conversations in the format of (H, X, y). Here, $X = \{x_1, x_2, \cdots, x_{n_x}\}$ denotes n_x context utterances of the current conversation session. $H = \{H^1, H^2, \cdots, H^M\}$ denotes M history conversation sessions, where $H^i = \{h_1^i, h_2^i, \cdots, h_{n_i}^i\}$ denotes n_i chronologically ordered utterances of the i-th history conversation session. y is the ground truth response to X under the background of H. The MSC task can be formulated as learning a function $f(H, X)$ to predict the next utterance x_{n_x+1} based on H and X.

In this work, we propose a novel model, namely HAHT, for the MSC task. Figure 2 shows the overall structure of HAHT, which consists of three main components: 1) hierarchical history conversation encoder, 2) history-aware context encoder, and 3) history-aware response generator. We present the details of each component of HAHT as follows.

3.1 Hierarchical History Conversation Encoder

The main challenge in encoding history conversation sessions is the limited maximum input length imposed by pre-trained dialogue systems. If all history conversations are simply concatenated and fed into the pre-trained dialogue system, the length of the concatenation will exceed the maximum input length. Thus, most parts of the input will be truncated. To preserve more information in the history conversation, we encode each history conversation session separately in a hierarchical fashion.

Specifically, for a history conversation session $H^i = \{h_1^i, h_2^i, \cdots, h_{n_i}^i\}$, we first prepend a special token "User:" or "Assistant:" to each utterance h_j^i in H^i depending on the role of the utterance speaker, and then pad all utterances to the same length l_{utter}. For each utterance h_j^i, we apply an embedding layer E_m, n_{enc} Transformer encoder layers, and a Max-pooling layer to obtain its dense representation as follows,

$$\mathbf{u}_j^i = \text{Max-pooling}\big(\text{Transformer}_{n_{enc}}(E_m(h_j^i))\big), \tag{1}$$

where $\mathbf{u}_j^i \in \mathbb{R}^d$. Moreover, we denote all the utterance representations in the history conversation H^i by $\mathbf{U}^i = \{\mathbf{u}_1^i, \mathbf{u}_2^i, \cdots, \mathbf{u}_{n_i}^i\} \in \mathbb{R}^{n_i \times d}$, where n_i is the turn number of H^i. Next, we apply a