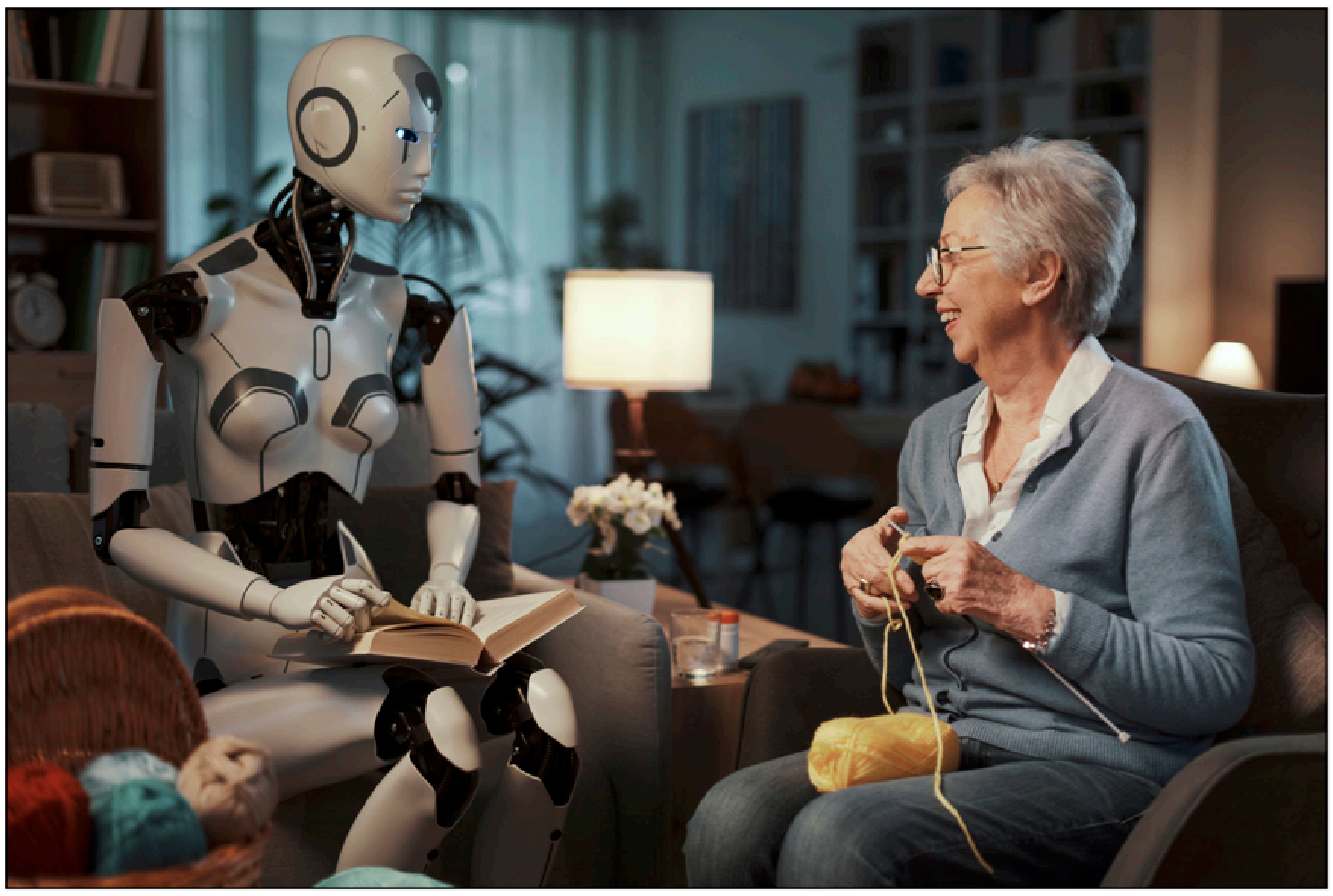
Luyao Zhu, Wei Li, Rui Mao, and Erik Cambria
Nanyang Technological University, SINGAPORE

HIPPL: Hierarchical Intent-Inferring Pointer Network With Pseudo Labeling for Consistent Persona-Driven Dialogue Generation

Abstract

Despite the recent advancements in dialogue systems, persona-driven chatbots are still in their infancy. Previous studies on persona-driven dialogue generation demonstrated its ability in generating responses that contain more detailed persona information. However, the challenge of maintaining persona consistency and contextual coherence still persists in persona-driven dialogue generation. Moreover, current methods have limitations in processing multi-source inputs and identifying interlocutor intents due to the absence of trustworthy labels and effective modeling. Additionally, numerous approaches rely on pre-trained large-scale language models that require costly computational resources. To address these challenges, a lightweight hierarchical intent-inferring pointer network is proposed for multi-source persona-driven dialogue generation. The proposed method involves detecting interlocutor intents in chitchat and utilizing pseudo labeling and natural language inference techniques to generate intent labels. Our model is evaluated on a benchmark dataset PersonaChat. The experimental results show that our model outperforms the strongest baseline by 13.47% and 4.28% in terms of persona consistency and contextual coherence, respectively.



© SHUTTERSTOCK/STOCK-ASSO

I. Introduction

Advances in dialogue generation have been propelled by recent developments in deep learning and the availability of large-scale open-domain conversation data [1], [2]. Meanwhile, persona-driven dialogue generation, an emerging area of controllable dialogue generation [3], has gained increasing attention in the field of natural language generation [4], as it enforces response specificity. Persona-driven dialogue generation is defined as a task of training a dialogue agent to generate responses while incorporating pre-defined persona information. Generally, persona profiles a person's occupation, educational background, hobbies, family, and social relations, etc. A dialogue agent with pre-defined personas would provide sufficient common ground [5], [6] for human-computer interaction, which is recognized as one of the crucial factors for successful communication between interlocutors [7]. For example, a musician is likely to talk more with people about musical knowledge as they may share a common interest.

Here, the persona of music interest serves as a shared topic through which the interlocutors can learn about each other and establish a positive relationship. Responses from a persona-driven dialogue agent are more grounded as they closely align with pre-defined persona information, whereas those generated by a non-controllable dialogue agent may be groundless and inconsistent [8], [9], e.g., *I live in California* and *I live in New York* may appear in different turns of the same dialogue. This is because non-controllable dialogue agents generate responses based on the probability distribution learned from training data, disregarding the actual persona information.

Digital Object Identifier 10.1109/MCI.2024.3446133
Date of current version: 8 October 2024

This article was recommended for publication by Associate Editor Keeley Crockett. Corresponding author: Erik Cambria (e-mail: cambria@ntu.edu.sg).

Some persona-driven systems adopted a universal encoder-decoder framework [10] or lacked consistent language modeling [8]. They employed a general objective function to simultaneously learn the generation and persona fusion tasks. However, the general objective function cannot coherently incorporate persona information in an utterance context, because the input sources, i.e., persona and historical utterances from different interlocutors, have different information significance and learning patterns in the generation and fusion tasks. Thus, persona inconsistency [11] remains a common problem among the existing methods. Although intent detection has been widely employed in task-oriented dialogue systems [12], [13], it is rarely incorporated into open-domain persona-driven dialogue systems. Persona-oriented intent detection in the open domain is particularly challenging due to the lack of ground-truth labels and the ambiguity of persona information selection in generating contextualized responses. Overlooking interlocutor intents in a response in the open domain may result in a failure to incorporate the appropriate persona attribute. Moreover, several state-of-the-art studies on persona-driven dialogue generation utilized pre-trained language models (PLMs) [14], which typically come with high computational costs. The problem becomes even more severe when deploying a localized dialogue system on personal devices.

Motivated by the importance of utilizing persona information and understanding interlocutor intents, we develop a lightweight dialogue system to detect and track interlocutor intents and to generate context-coherent and persona-consistent dialogue responses based on predefined personas. To learn information from multi-sources, i.e., persona, and historical utterances from a speaker and an agent, separate encoders are employed. Additionally, a well-designed pointer generator is proposed to fuse information from multiple encoders.

To detect and track interlocutor intents, the intent detector and tracker modules are proposed, trained with pseudo labels and a multi-task learning paradigm. Furthermore, unlike state-of-the-art baselines, our generation model does not rely on PLMs, making it a more lightweight and computationally efficient solution. Specifically, **a)** separate encoders are employed to capture the distinct distributions of utterances from different interlocutors and persona descriptions. Then, a global encoder is introduced to extract the global context and retain long-term memory for multi-turn inputs. **b)** For interlocutor intent inference, the natural language inference (NLI) technique, a semi-supervised learning (SSL) approach, and pseudo labeling are utilized to automatically annotate the interlocutor intent. Next, an intent tracker is designed as an auxiliary module to enhance the representation capacity of the dialogue agent. In addition, a multi-task learning-based exterior intent detector is trained to infer the interlocutor intent. **c)** In the decoder, a multi-source pointer-generator is proposed to leverage the useful information from multiple input sources and filter out irrelevant textual noises. **d)** In the generation task, the weighted sum of losses is computed for the generator and the intent tracker to update the model parameters. It simultaneously strengthens the representation capacity and the generation capability of the model.

Our method is evaluated on a publicly available dataset PersonaChat [10]. The automatic evaluation results show that our model outperforms the PLM-based methods in a wide range of automatic evaluation metrics, such as BLEU [15] (+0.151), METEOR [16] (+1.155), ROUGE-L [17] (+2.066), F1 [18] (+1.800), greedy (+0.86), and extrema (+1.13) embedding-based evaluation metrics [19]. The human evaluation results indicate that our model surpasses the strong baselines in diverse evaluation dimensions, such as contextual coherence (+0.0967), inverse duplicate score (+0.0200), and persona consistency (+0.0934). Overall, our model's parameter size is approximately only 20% of that of the strongest GPT-2-based model [14]. Our ablation study demonstrates the utilities of different technical components of our model in encoding, intent inference, pseudo labeling, and decoding. Finally, hyper-parameters used in our model are systematically analyzed.

The contributions of our work are summarized as follows:

❏ A lightweight hierarchical intent-inferring pointer network is put forward for multi-source and multi-turn consistent persona-driven dialogue generation.

❏ A method is proposed to detect interlocutor intents for open-domain conversation, which is trained by the generated intent labels via pseudo labeling and NLI techniques.

❏ The experimental results show that our model outperforms baselines in various evaluation metrics, including persona-consistency and contextual coherence.

The remainder of this work is organized as follows: Section II briefly illustrates related work; next, Section III explains the mechanism of our model; later, Section IV and Section V describe experiments and results, respectively; next, Section VI proposes a discussion of such results; finally, Section VII proposes concluding remarks.

II. Related Work

A. Persona-Driven Dialogue Generation

Open-domain dialogue systems or chitchat bots have obtained growing attention from academia and industry. Chitchat agents face several challenges, including: 1) inconsistent persona [9], 2) the absence of an explicit long-term memory [8], and 3) a tendency to generate vague or uninformative responses, such as *I don't know* [20]. Such issues can result in unsatisfactory and unappealing conversational experiences between human and dialogue agents. Thus, Zhang et al. [10] established a Persona-Chat dataset for training a more specific, personal, consistent, and engaging dialogue agent, alleviating the common issues of chitchat models. It provides a benchmark and resource for persona-driven dialogue generation.

Based on the persona-driven dialogue generation [10], several research works were presented to deal with the common issues in dialogue generation. Li et al. [21] improved logical consistency and reduced repetitions within utterances as well as the overuse of frequent words through unlikelihood training. Some works focused on enhancing the performance of personalized dialogue agents through well-designed frameworks. For example, Young et al. [22] combined task-oriented and open-domain dialogue generation tasks, enabling the dialogue agent to achieve task objectives and generate responses with more personalized information. Song et al. [1] exploited pre-trained NLI classifiers to calculate the deep reinforcement learning reward in terms of consistency, which enabled the dialogue agent to generate more persona-consistent responses.

However, their research employed PLMs to improve the quality of generated responses, resulting in increased computational complexity. Furthermore, they neglected the potential exploitation of user intent, which could be a crucial factor in improving the contextual coherence of generated responses.

B. User Intent Classification

User intent classification (UIC) has gained popularity in the development of task-oriented dialogue systems, because it has been demonstrated to improve the user experience during interactions with these systems [13]. For example, Wang et al. [13] developed a meta lifelong learning framework for large-scale extensible UIC. It enables the dialogue agent to continuously adapt to new tasks. Ni et al. [23] designed a two-hierarchy learning framework to learn turn- and global-level intents, i.e., conversation goals.

In contrast, there is comparatively less research on detecting the interlocutor intent in chitchat scenarios, e.g., persona-driven dialogue generation. Overlooking user intents in chitchat may make dialogue agents fail to meet the conversation needs, leading to negative user attitudes toward the dialogue system [24]. However, the existing approaches tailored for UIC in task-oriented dialogue generation could not be directly applied to chitchat agents, as the intent categories are more diverse and less well-defined. According to Bickmore and Cassell [25], chitchat or small talk is utilized as a relational strategy to establish a sense of trust with users in conversational interactions. It involves conversations that prioritize interpersonal objectives while either de-emphasizing or disregarding task-oriented goals. Therefore, we regard the interlocutor intent in persona-driven dialogues as de-emphasized task goals and propose to solve the lack of reliable intent labels through pseudo labeling and NLI techniques.

C. Pointer Network

The pointer network, which was put forward in the paper [26], aimed to modulate a content-based attention mechanism over inputs. It has been applied in a wide range of NLP research areas, e.g., text summarization [27], question answering [28], and dialogue generation [29]. The pointer network enables the model to copy words from the source text via pointing and retain the capability to generate new words. Most of the existing research works utilize a pointer network for problems with single-source inputs. However, they are not designed for processing multi-source inputs.

Recently, some researchers have improved the pointer network architecture to handle multi-source inputs. Sun et al. [30] proposed a multi-source pointer network with an additional knowledge encoder for product title summarization. It can copy words not only from the product title inputs but also from texts containing background knowledge. However, this improvement comes at the cost of forfeiting the pointer network's ability to generate new words beyond the input texts. Yavuz et al. [31] advanced a hierarchical pointer network by extending the pointer generator [32]. It copes with the multi-source inputs by balancing between two existing contents to be copied first, and then choosing between generating new words and copying the existing tokens. This hierarchical method is based on a binary mechanism, which is highly dependent on expert experience and hard to extend when fed with additional source inputs. Moreover, as a stacked attention mechanism [33], it may suffer from a vanishing gradient problem [34].

III. Methodology

A. Task Definition

PDG: The goal of persona-driven dialogue generation (PDG) is to learn a controllable generative model to generate consistent persona-driven dialogues. The task is defined as follows: Given a set of agent's persona texts $U^P = \{U_1^P, U_2^P, \ldots, U_m^P\}$ and the dialogue history of two interlocutors $U^H = \{U_1^b, U_2^a, \ldots, U_{T-1}^b\}$, the task is to deliver a response $\hat{R}$ that is consistent with the given persona U^P and to avoid the repetition of dialogue history U^H. Thus, a dialogue intent detector would be applied to interlocutor intent inference before generation. Here, a denotes the agent, b denotes the speaker, P denotes the persona, and T denotes the current time step.

PIT: Persona intent tracking (PIT) is an information retrieval task over persona descriptions. Supposing that a dialogue system decides to include persona information in the current response, it needs to select an appropriate persona element U_i^P from the persona description set U^P. Intuitively, the selected persona element should be relevant to the local topic U_{T-1}^b and avoid repeating previous persona elements that have appeared in historical responses.

III: Interlocutor intent inference (III) consists of a binary intent classification task (BIC) and a PIT task. Given the dialogue history U^H and the persona description set U^P, the first step of BIC is to decide whether the current

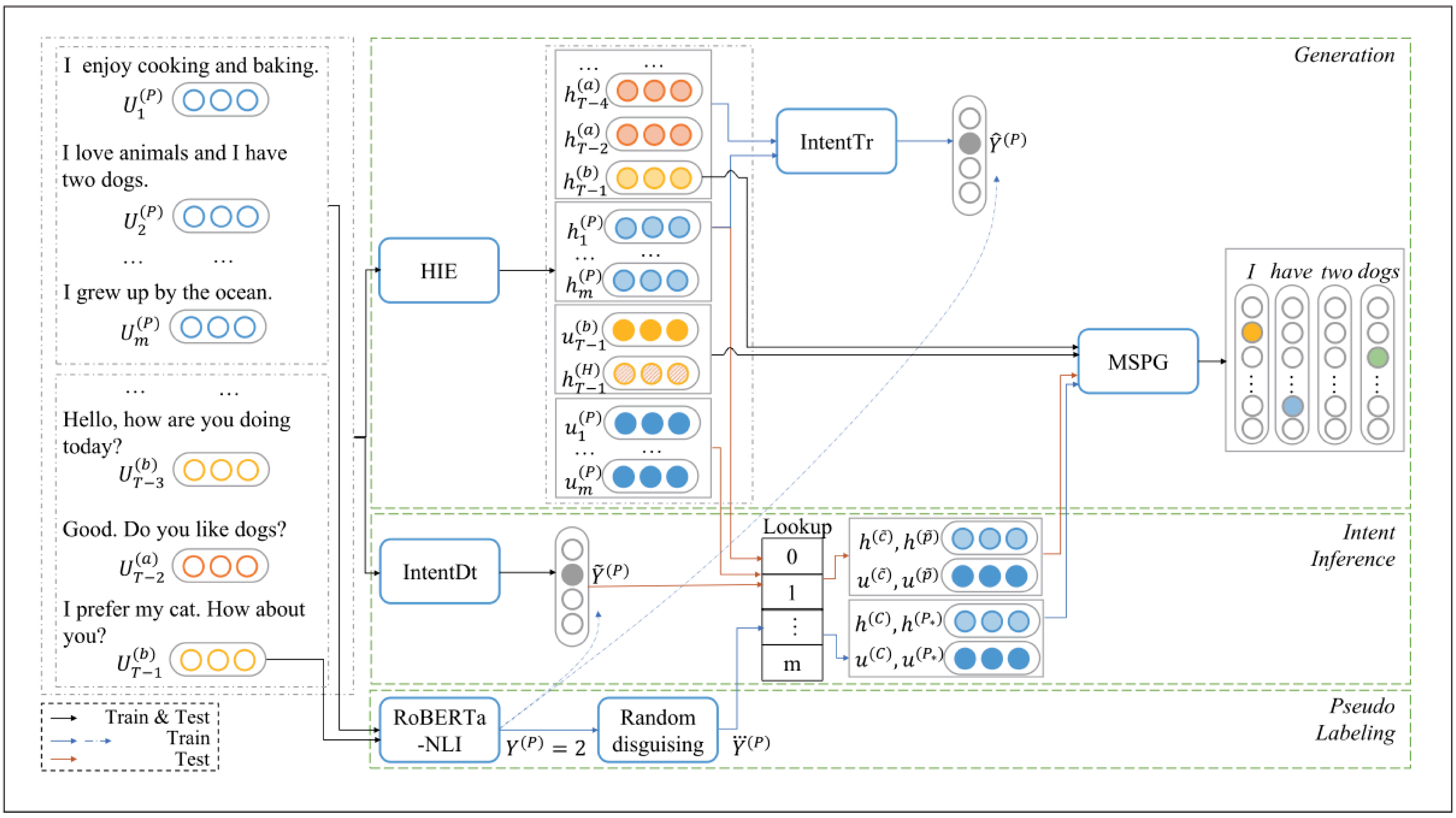


FIGURE 1 The framework of HIPPL. The blue arrows mean that the relevant data flows exist in the training procedure only; the orange ones correspond to the test procedure only; the black ones always exist during the whole procedure. Random disguising is randomly changing the pseudo-intent label to another fake label with a pre-defined probability.

response should include persona information. Then, an appropriate persona element U_i^P should be selected in PIT task. In this paper, an interlocutor intent detector is designed for III task. The intent detector consists of a binary intent classifier for BIC and an intent retriever for PIT.

B. Framework

The architecture of our hierarchical intent-inferring pointer network with pseudo labeling (HIPPL) is shown in Figure 1. HIPPL mainly consists of three technical components, namely, generation, intent inference, and pseudo labeling. The data flows within and among the parts are introduced as follows.

First, the input sequences, including the dialogue history of two interlocutors $U^H = \{U_1^b, U_2^a, \ldots, U_{T-1}^b\}$ and persona descriptions $U^P = \{U_1^P, U_2^P, \ldots, U_M^P\}$ are encoded with interlocutor encoders (IntEnc) in a hierarchical interlocutor encoder module (HIE), generating the word-level hidden states u^s, utterance-level hidden states h^s and global context h^g by

$$u^s, h^s, h^g = IntEnc(U_1^s, U_2^s, \ldots, U_{L_s}^s), \tag{1}$$

where $s \in \{a, b, P\}$, and L_s is the number of utterances or persona elements. $IntEnc(\cdot)$ is detailed in Section III-C. Here, we regard the hidden state from the speaker (h_{T-1}^b) at the last time step ($T-1$) as the local topic.

As introduced before, interlocutor intent inference is pivotal for a dialogue agent to generate grounded responses. Thus, a separate multi-task learning-based intent detector (IntentDt) is designed to select the intent $U^{\tilde{P}}$ out of an intent set $I = \{U^P \cup U_0^P\}$ based on dialogue history U^H. U_0^P is a padded sequence, representing the persona-irrelevant intent. As shown in (2), intent labels $\tilde{Y}_b^P$ and $\tilde{Y}_m^P$ are predicted by IntentDt. $\tilde{Y}_b^P$ is a binary classification label, indicating whether the intent is relevant to the persona or not. $\tilde{Y}_m^P$ is a multi-class classification label, indicating which persona element is selected from U^P. The selected intent is given by

$$\tilde{Y}_b^P, \tilde{Y}_m^P = IntentDt(U^P, U^H), \tag{2}$$

$$U^{\tilde{P}} = \begin{cases} U_0^P, & \text{if } \tilde{Y}_b^P = 0 \\ U_i^P, & \text{if } \tilde{Y}_b^P \neq 0, i = \tilde{Y}_m^P. \end{cases} \tag{3}$$

$IntentDt(\cdot)$ is detailed in Section III-D. After selecting $U^{\tilde{P}}$ according to (3), the remaining intents in I is termed as persona complements $U^{\tilde{c}}$. Besides $\tilde{Y}_b^P$ and $\tilde{Y}_m^P$, we also obtain word-level ($u^{\tilde{P}}$, $u^{\tilde{c}}$) and utterance-level hidden states ($h^{\tilde{P}}$, $h^{\tilde{c}}$) by selecting from hidden states (u^P and h^P) in (1).

The local topic, the global context (u_{T-1}^b and h^g, given by (1)), the predicted intent, and persona complements ($u^{\tilde{P}}$, $h^{\tilde{P}}$, $u^{\tilde{c}}$, and $h^{\tilde{c}}$ selected by IntentDt) are fed to a multi-source pointer-generator (MSPG) module. MSPG consists of a multi-source attention (MSA) and a pointer-generator (PG). MSPG is designed based on an attention mechanism to copy and fuse the information from multi-source inputs, i.e., the local topic, the predicted intent, persona complements, and the vocabulary distribution p_t from a decoder. Attention is used because it can selectively focus on informative contexts [35]. Therefore, the MSPG should have two functions: 1) assigning different weights to the tokens in the same input source through computed attention distribution a_t in MSA; 2) allocating different weights to different input sources by computing weights λ_t in PG.

$$a_t^\gamma, p_t = MSA\left(u^P, u_{T-1}^b, h^P, h^g, y_{t-1}^{EMB}\right) \tag{4}$$

In (4), $\gamma \in \{\tilde{p}, \tilde{c}, b\}$, $u^P = \{u^{\tilde{p}}, u^{\tilde{c}}\}$, $h^P = \{h^{\tilde{p}}, h^{\tilde{c}}\}$. y_{t-1}^{EMB} is the embedding of the generated word in step $t-1$. Then, weights λ_t^γ and λ_t^v for different input sources in PG are computed based on the extracted information of the multi-source inputs. The output probability $P(y_t = w|U^H, U^P, y_{<t})$ of a generated word w at position t is given by the weighted summation of the computed distributions:

$$P(y_t = w|U^H, U^P, y_{<t}) = \sum_{\gamma}\sum_{i:w_i=w} \lambda^\gamma a_{ti}^\gamma + \lambda^v \sum_{j:w_j=w} p_t(w_j). \tag{5}$$

In (5), $i : w_i = w$ means to find i where $w_i = w$, v is the abbreviation for vocabulary, and λ_t^v is a computed weight for the decoder vocabulary. MSA and PG are detailed in Section III-E. In addition, following HIE, there is a persona intent tracker (IntentTr) trained to imitate the function of IntentDt. IntentTr predicts a multiple classification label $\hat{Y}_m^P$. $\hat{Y}_m^P$ is used for selecting the intended persona element $U^{\hat{p}}$ out of persona descriptions U^P. This is an auxiliary NLI-equivalent task to improve the representation capability of the model in the generation part. As Conneau et al. [36] argued that the universal sentence representation could be learned from the supervised learning, based on the NLI task, we believe the joint training of the intent tracking module would benefit the sentence representation learning of HIE. It is notable that the pseudo label Y^P for training IntentDt and persona intent tracker (IntentTr) is obtained by pseudo labeling based on pre-trained RoBERTa-NLI [37]. The binary-class label Y_b^P for the BIC task and the multi-class label Y_m^P for the PIT task can be obtained by transforming Y^P. In addition, the data flow in the training procedure has differences from that in the test procedure. During the training procedure, Y^P is also used for retrieving the corresponding hidden states h^p, u^p, and h^c, u^c for the true interlocutor intent and persona complements. In this paper, the variables with p as a superscript are considered to contain the true interlocutor intent information; those with c as a superscript are considered to contain true persona complements. These hidden states are fed to a multi-source pointer-generator (MSPG). Moreover, we notice the difference between the predicted intent label $\tilde{Y}^P$ from IntentDt and the pseudo label Y^P, and that between the pseudo label Y^P from RoBERTA-NLI and the ground-truth label. Thus, random disguising is designed to generate disguised labels $\ddot{Y}^P$, which is controlled by pre-defined probabilities (τ_b, τ_m), to improve the generalization of MSPG. Random disguising can be categorized as an adversarial attack technique [38]. It is similar to generating uniform white noise in computer vision. Two hyper-parameters τ_b and τ_m are defined, since the accuracy of IntentDt predicting the binary-class label $\tilde{Y}_b^P$ and the multi-class label $\tilde{Y}_m^P$ is different. The different data flows with random disguising in the training and test process are elaborated in Algorithm 1.

Algorithm 1. Sketch of HIPPL.

```
Input: dialogue history $U^H$, persona descriptions $U^P$, target response $U^R$, random
    disguising thresholds $\tau_b$ and $\tau_m$, interlocutor encoders IntEnc, intent tracker
    IntentTr, multi-source pointer-generator MSPG, intent detector IntentDt,
    pre-trained RoBERTa-NLI model RoBERTa, and cross-entropy loss CrossEntropy
Result: generated responses
Randomly initialize IntEnc, IntentTr, MSPG, and IntentDt.
Generate pseudo labels $Y^P$ from pseudo labeling based on RoBERTa with inputs $U^R$
  and $U^P$.
Pre-train IntentDt with inputs $U^H$, $U^P$, and labels $Y^P$.
for $i \leftarrow 1$ to number of samples do
  $u^s, h^s, h^g \leftarrow IntEnc(U^H, U^P)$
  if train then                                          /* random disguising */
    Randomly generate $num_b$ and $num_m$
    if $(num_b \leq \tau_b) \wedge (Y^P \neq 0)$ then
      $\tilde{Y}^P \leftarrow 0$
    else if $(num_b \leq \tau_b) \wedge (num_m \leq \tau_m)$ then
      $\tilde{Y}^P \leftarrow random(1, M) \backslash Y^P$
    else
      $\tilde{Y}^P \leftarrow Y^P$
  else
    $\tilde{Y}^P \leftarrow IntentDt(U^H, U^P)$
  end
  Lookup $u^{\tilde{p}}, h^{\tilde{p}}, u^{\tilde{c}}, h^{\tilde{c}}$ according to $\tilde{Y}^P$
  Generate the response
    $U^R \leftarrow MSPG(u^{\tilde{p}}, h^{\tilde{p}}, u^{\tilde{c}}, h^{\tilde{c}}, u^b_{T-1}, h^g)$
  Compute the generation loss $\mathcal{L}_g$
  if train then
    $\hat{Y}_m^P \leftarrow IntentTr(U^H, U^P)$
    Compute the auxiliary loss
      $\mathcal{L}_t \leftarrow CrossEntropy(Y_m^P, \hat{Y}_m^P)$
    Optimize IntEnc, IntentTr, and MSPG according to $\alpha_g \mathcal{L}_g + \alpha_t \mathcal{L}_t$
  end
end
```

C. Hierarchical Interlocutor Encoder Module

The bidirectional gated recurrent unit (BiGRU) [39] is utilized to encode an utterance. The last hidden states of forward and backward GRUs are concatenated as the context of an encoded utterance, i.e., $h = [\overrightarrow{h_k}, \overleftarrow{h_k}]$, where k refers to the current time step for an encoder. Here, three independent BiGRU encoders are used to extract the latent representations of utterances h^b, h^a, and h^P from a speaker, an agent, and the persona, respectively:

$$u^s, h^s = BiGRU^s(w^s), \tag{6}$$

where $s \in \{a, b, P\}$. $w^s = (w_1^s, \ldots, w_{L_s}^s)$ is an embedded utterance or persona

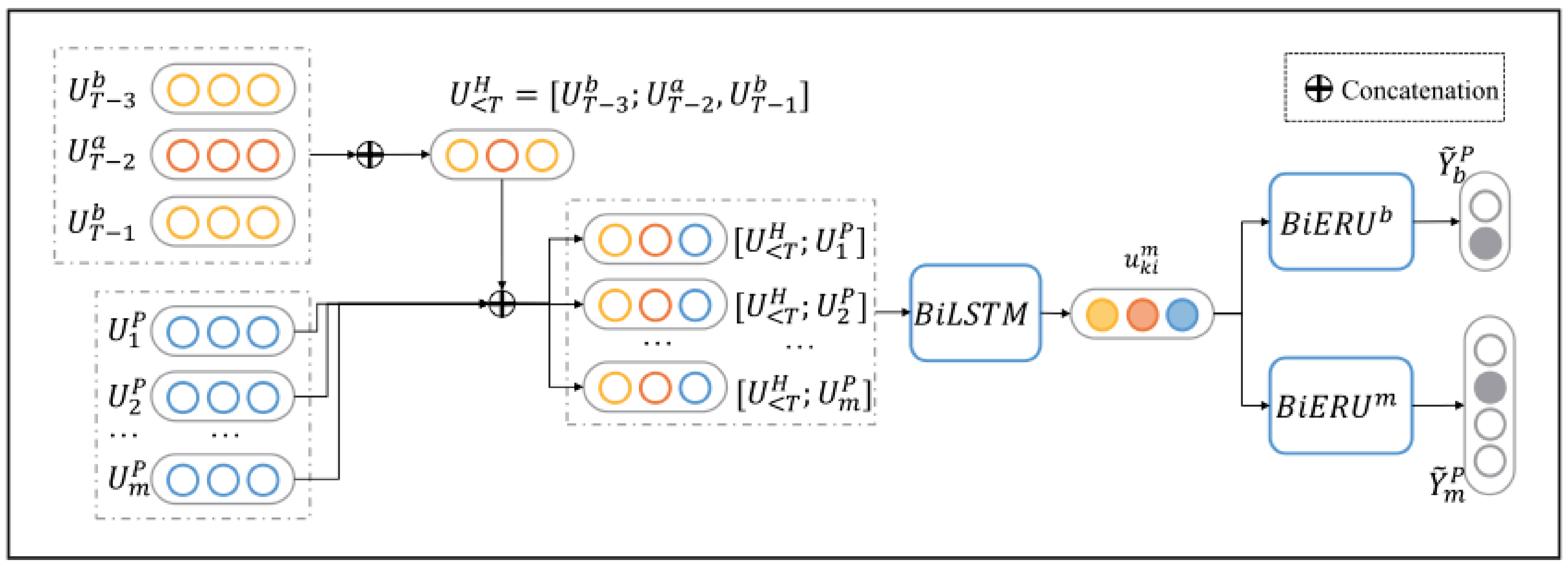


FIGURE 2 The architecture of our multi-task intent retriever.

element, given by $w^s = Emb(U^s)$. $U^s = (W^s_1, \ldots, W^s_j, \ldots, W^s_{L_s})$. W^s_j is the j^{th} word in U^s. $Emb(\cdot)$ is the embedding layer. $u^s = (u^s_1, .., u^s_{L_s})$ is the generated hidden states of all words in U^s. u^s is named as word-level hidden states. h^s is the last hidden state obtained from feeding w^s into the encoder $BiGRU^s(\cdot)$. Then, a forward GRU ($GRU^g(\cdot)$) is employed to encode the extracted utterance hidden states $h^H = (h^b_1, h^a_2, \ldots, h^b_{T-1})$ chronologically, yielding the contextual hidden states h^g:

$$h^g = GRU^g(h^H). \tag{7}$$

Here, h^g at the last time step is taken as the global context.

D. Intent Detector Module

The intent detector module (IntentDt) is used to infer the intent of the interlocutor (the III task). It works as an independent model, providing the selected persona information for the downstream MSPG. It is found that a speaker (user) does not always want to learn about the persona of the other interlocutor (agent) in each turn of the dialogue. For example, a speaker may want to end the dialogue or receive compliments sometimes. Therefore, our intent detector module is trained to select the appropriate element from an intent set $I = \{U^P_1, U^P_2, \ldots, U^P_M\} \cup \{U^P_0\}$, where U^P_0 is represented by the padding index in this work.

Binary intent classifier. The binary intent classifier is employed to predict whether a speaker wants to receive persona-relevant responses (BIC task). The concatenated context $U^H_{<T}$ and persona descriptions U^P are input into ALBERT [40] to obtain context-aware utterance-level embeddings u_k. $u_k \in \mathbb{R}^{d_a}$, where d_a is the embedding dimension of ALBERT. Then, u_k is fed to the BiERU [41] for binary classification.

Multi-task intent retriever. In those cases when a speaker is recognized as needing persona-relevant responses, a multi-task intent retriever is employed to select a persona element U^P_i that is the most relevant to the context $U^H_{<T}$. A multi-task learning paradigm is exploited, as it allows the model to share knowledge between different learning tasks [42], [43]. As shown in Figure 2, the multi-task intent retriever is composed of two BiERU classifiers, sharing the same utterance-embedding BiLSTM [44] layer. One classifier $BiERU^m$ is designed for the PIT task and the other $BiERU^b$ is tailored for the BIC task as an auxiliary module.

Here, the context $U^H_{<T}$ is concatenated with each persona element U^P_i and fed into a one-layer BiLSTM to obtain an utterance-level embedding of a context-persona-element pair u^m_{ki}.

$$\begin{aligned} u^m_{ki} =& BiLSTM \\ & (U^H_{k-l}; U^H_{k-l+1}; \ldots; U^H_k; U^P_i), \end{aligned} \tag{8}$$

where $k \leq T-1$ and

$$U^H_{k-l} = \begin{cases} U^a_{k-l}, & \text{if } \frac{l}{2} = 1 \\ U^b_{k-l}, & \text{if } \frac{l}{2} = 0. \end{cases} \tag{9}$$

E. Multi-Source Pointer-Generator Module

Although the existing pointer networks enhance dialogue generation [29], [45], they may lose the existing history information or the predicted intent information, as they cannot handle the multi-source inputs. In addition, the related multi-source pointer network [30] is designed for extraction, rather than generation. To this end, a multi-source pointer-generator module (MSPG) is proposed for the pointer network to exploit the key information in the dialogue history, interlocutor intents, and persona complements, while maintaining its capability to produce novel content.

As mentioned in the HIE module in Section III-C, we introduce a persona encoder besides the encoders for the two interlocutors. Thus, the model can utilize words from multiple sources, such as the dialogue interlocutor encoders, and the persona encoder. As Figure 3 shows, during the prediction of a current word by the decoder, it takes into account the probability of the token in four distinct aspects: 1) the local topic *I prefer my cat how about you*, 2) the predicted interlocutor intent *I love animals and I have two dogs*, 3) persona complements *I enjoy cooking and baking… I grew up by the ocean*, and 4) the distribution over the fixed vocabulary. This is achieved by a multi-source attention mechanism and a multi-source pointer generator.

E1) Multi-Source Attention

For an ongoing dialogue, the dialogue agent needs to focus on four elements: 1) a global context, 2) a local topic, 3) the interlocutor intent, and 4) persona complements. The hidden state h^g is used as the global context, obtained from encoding the series of hidden states $(h^b_1, h^a_2, \ldots, h^b_{T-1})$ by contextual encoders. Then, the global context h^g concatenated with the encoded persona complements $h^{\tilde{c}}$ is transformed into the initial hidden state d_0 of the decoder through a rectified layer [46]

$$d_0 = ReLU(W_c[h^g; h^{\tilde{c}}]), \tag{10}$$

where $ReLU(x) = max(0, x)$, and $W_c \in \mathbb{R}^{4d \times d}$ denotes trainable parameters. d is the dimension of hidden states from encoders and the decoder.

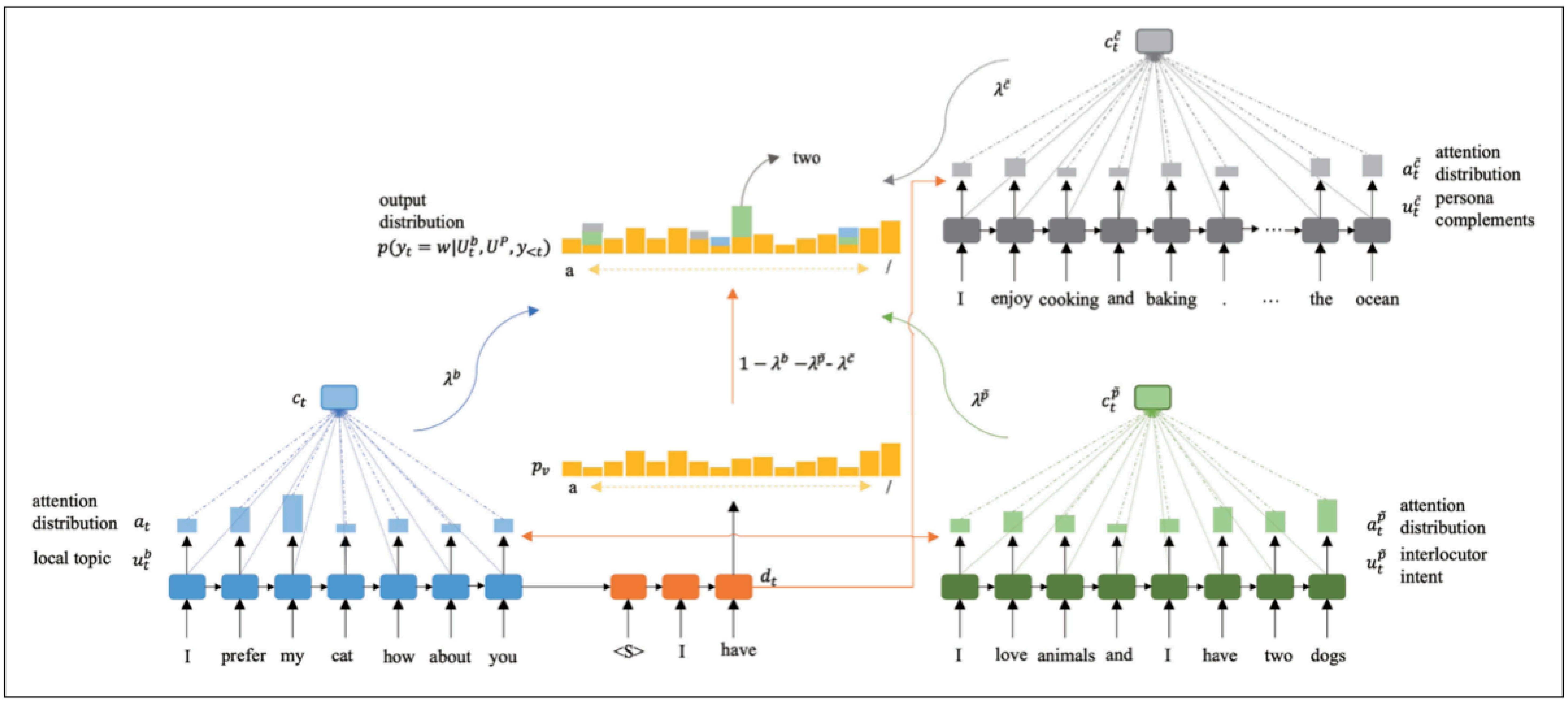

output distribution
$p(y_t = w|U^b_t, U^P, y_{<t})$
two
a
/
λ^b
c_t
$1 - \lambda^b - \lambda^{\tilde p} - \lambda^{\tilde c}$
p_v
a
/
attention distribution a_t
local topic u^b_t
I
prefer
my
cat
how
about
you
<S>
I
have
d_t
$c^{\tilde c}_t$
$\lambda^{\tilde c}$
attention distribution $a^{\tilde c}_t$
persona complements $u^{\tilde c}_t$
I
enjoy
cooking
and
baking
.
…
the
ocean
$\lambda^{\tilde p}$
$c^{\tilde p}_t$
attention distribution $a^{\tilde p}_t$
interlocutor intent $u^{\tilde p}_t$
I
love
animals
and
I
have
two
dogs

FIGURE 3 Overview of our proposed multi-source pointer-generator. The decoder hidden state d_t is utilized to attend the local topic, the interlocutor intent, and persona complements to compute distributions for copying words from them. The decoder also delivers a distribution over all tokens in the vocabulary. All three distributions are integrated to generate the overall distribution at each time step t.

As mentioned, a local topic $u^b_{T-1} = (u^b_{T-1,1}, u^b_{T-1,2}, \ldots, u^b_{T-1,l_b})$ is the word-level hidden states of U^b_{T-1} from the speaker encoder $BiGRU^b$. $u^b_{T-1} \in \mathbb{R}^{2d\times l_b}$ denotes the local topic. l_b is the sentence length of U^b_{T-1}. The predicted intent $u^{\tilde p} = (u^{\tilde p}_1, u^{\tilde p}_2, \ldots, u^{\tilde p}_{l_p})$ is selected from the word-level hidden states u^P according to intent labels, $\tilde{Y}^P_b$ and $\tilde{Y}^P_m$, from IntentDt in a testing procedure, while pseudo labels are used in a training procedure. Here, $u^{\tilde p} \in \mathbb{R}^{2d\times l_p}$ is the predicted intent. l_p denotes the sentence length of the predicted intent. Persona complements $u^{\tilde c} = (u^{\tilde c}_1, u^{\tilde c}_2, \ldots, u^{\tilde c}_{l_c})$ are the remaining contents in u^P after the selection, where $u^{\tilde c} \in \mathbb{R}^{2d\times l_c}$. l_c is the total number of tokens in persona complements. In order to figure out the significance of different tokens in each input source, their attention distribution is computed as follows:

$$a_t = softmax(e_t) \tag{11}$$

$$a^{\tilde p}_t = softmax(e^{\tilde p}_t) \tag{12}$$

$$a^{\tilde c}_t = softmax(e^{\tilde c}_t) \tag{13}$$

$$e_{ti} = v^T tanh(W_u u^b_{T-1,i} + W_h d_t + b_{attn}) \tag{14}$$

$$e^{\tilde p}_{tj} = v^{pT} tanh(W^p_u u^{\tilde p}_j + W^p_h d_t + b^p_{attn}) \tag{15}$$

$$e^{\tilde c}_{tk} = v^{pT} tanh(W^p_u u^{\tilde c}_k + W^p_h d_t + b^p_{attn}), \tag{16}$$

where v^T, v^{pT}, W_u, W^p_u, W_h, W^p_h, b_{attn}, and b^p_{attn} are trainable parameters. As (15) and (16) show, the predicted intent $u^{\tilde p}$ and persona complements $u^{\tilde c}$ share the same attention mechanism because their data distribution is assumed to be similar. a_t is the attention distribution for the encoded local topic (the speaker encoder at time step t), $a^{\tilde p}_t$ is the attention distribution for the encoded predicted intent, and $a^{\tilde c}_t$ is the attention distribution for encoded persona complements; d_t is the decoder hidden state at time step t, computed as:

$$d_t = GRU^d(q_t, d_{t-1}) \tag{17}$$

$$q_t = W_q[c_{t-1}; c^{\tilde p}_{t-1}; y^{EMB}_{t-1}], \tag{18}$$

where d_{t-1} is the decoder hidden state at the time step $t-1$; y^{EMB}_{t-1} is the embedding of the predicted word y_{t-1} at the time step $t-1$. $GRU^d(\cdot)$ is a nonlinear function, forward GRU in this work. $W_q \in \mathbb{R}^{(4d+d_e)\times d_e}$ denotes trainable parameters; d_e is the dimension of embeddings from an embedding layer. The local topic vector c_{t-1} and the intent vector $c^{\tilde p}_{t-1}$ are given by

$$c_t = \sum_i a_{ti} u^b_{ti} \tag{19}$$

$$c^{\tilde p}_t = \sum_i a^{\tilde p}_{ti} u^{\tilde p}_i, \tag{20}$$

where a_{ti} and $a^{\tilde p}_{ti}$ are weights in a_t and $a^{\tilde p}_t$ at the position i, respectively. Similarly, the persona complement vector $c^{\tilde c}_t = \sum_i a^{\tilde c}_{ti} u^{\tilde c}_i$ will be used to generate soft gating weights λ later.

To maintain the generation capacity of the model, vocabulary distribution is also necessary. It is obtained by concatenating the local topic vector c_t with the decoder hidden state d_t, then feeding them through two fully connected layers:

$$p_v = softmax(V'(V[d_t; c_t] + b_v) + b'_v), \tag{21}$$

where $V \in \mathbb{R}^{3d\times d}$, $V' \in \mathbb{R}^{d\times d_V}$, $b_v \in \mathbb{R}^d$, and $b'_v \in \mathbb{R}^{d_V}$ are parameters to be learned. p_v is the probability distribution over all words in the vocabulary, providing the final distribution from which to predict the word w:

$$p(w) = p_v(w). \tag{22}$$

E2) Multi-Source Pointer-Generator

Our multi-source pointer-generator network allows for both copying words via pointing multiple source inputs and generating words from a fixed corpus vocabulary. Soft gating weights λ^b, $\lambda^{\tilde p}$, $\lambda^{\tilde c}$, and λ^d are introduced to choose among the following possible actions: 1) copying a word that has the largest attention weight in a_t from the local topic U^b_{T-1}; 2) copying a word that has the largest attention weight in $a^{\tilde p}_t$ from the predicted interlocutor intent $U^{\tilde{p}}$; 3) copying a word that has the largest attention weight in $a_t^{\tilde{c}}$ from persona complements $U^{\tilde{c}}$; 4) generating a word that has the largest probability in p_v from the vocabulary.

$$
\begin{aligned}
p(y_t = w|U^H, U^P, y_{<t}) \\
= \lambda^b \sum_{i:w_i=w} a_{ti} + \lambda^{\tilde{p}} \sum_{j:w_j=w} a_{tj}^{\tilde{p}} \\
+ \lambda^{\tilde{c}} \sum_{l:w_l=w} a_{tl}^{\tilde{c}} + \lambda^d \sum_{k:w_k=w} p_v(w_k) \quad (23)
\end{aligned}
$$

Here, $\lambda^d = 1 - \lambda^b - \lambda^{\tilde{p}} - \lambda^{\tilde{c}}$. Intuitively, λ^b, $\lambda^{\tilde{p}}$, $\lambda^{\tilde{c}}$, and λ^d should automatically adjust according to the local topic vector c_t, the interlocutor intent vector $c_t^{\tilde{p}}$, the persona complements vector $c_t^{\tilde{c}}$, the current decoder hidden state d_t, and the embedding of the last predicted word y_{t-1}^{EMB}:

$$
\begin{aligned}
\lambda = softmax\big(W_\lambda c_t + W_\lambda^{\tilde{p}} c_t^{\tilde{p}} \\
+ W_\lambda^{\tilde{c}} c_t^{\tilde{c}} + W_\lambda^d d_t + W_\lambda^y y_{t-1}^{EMB}\big) \quad (24)
\end{aligned}
$$

where W_λ, $W_\lambda^{\tilde{p}}$, $W_\lambda^{\tilde{c}}$, W_λ^d, and W_λ^y are learnable parameters with the output size of 4. Thus, $\lambda \in \mathbb{R}^4$, where $\lambda = [\lambda^g, \lambda^b, \lambda^{\tilde{p}}, \lambda^{\tilde{c}}]$.

As shown in Figure 3, λ acts as a dynamic soft switch to extract different information from multiple source encoders. When the decoder is generating y_t, our MSPG enables the decoder to consider the tokens in the interlocutor intent (e.g., *two dogs*), the additional information in persona complements (e.g., *the ocean*), and the local topic (e.g., *cat*) besides the words in p_v via adjusting λ. Finally, the predicted token $\hat{y}_t$ (e.g., *two* in Figure 3) is the output at the time step *t*.

Here, different soft gating weights ($\lambda^{\tilde{p}}$ and $\lambda^{\tilde{c}}$) and separate attention mechanisms (on the encoded interlocutor intent $u^{\tilde{p}}$ and persona complements $u^{\tilde{c}}$) are employed. This is because the predicted interlocutor intent may contain the information that meets the speaker's needs and should be allocated with more attention. Additionally, the predicted interlocutor intent is generated from IntentDt which is trained with pseudo labeling. Thus, the attention mechanism on persona complements can mitigate the errors caused by IntentDt and improve the generalization capacity in the generation part.

F. Intent Tracker Module

As an auxiliary module in the generation part, the intent tracker (IntentTr) is designed to improve the representation performance of HIE. IntentTr is a modified NLI classifier. It plays two roles, namely, matching and selecting. The matching role is to identify the consistency between an encoded persona description h_i^P and the agent's previous target responses h_j^a in the history of a given dialogue, where $j < T - 1$, while the selecting role is to choose a persona element h_i^P that is the most relevant to an encoded local topic h_{T-1}^b. It is assumed that: 1) the NLI category between h_j^a and h_i^P should be *entailment* (E) or *neutral* (N); 2) to avoid repetition, the probability of an agent mentioning the persona element U_i^P that has appeared in previous responses should be reduced.

The neural tensor network [47] is utilized to extract the relationship o_i^b between the persona element h_i^P and the local topic h_{T-1}^b, and the relationship o_{ij}^a between the persona element h_i^P and agent's previous response h_j^a.

$$
\begin{aligned}
o_i^b = v_r^T LR\big({h_{T-1}^b}^T M^{[1:k]} h_i^P \\
+ V_r[h_{T-1}^b; h_i^P] + b_r\big) \quad (25) \\
o_{ij}^a = v_r^T LR\big({h_j^a}^T M^{[1:k]} h_i^P \\
+ V_r[h_j^a; h_i^P] + b_r\big) \quad (26)
\end{aligned}
$$

Here, $LR(\cdot)$ denotes LeakyReLU [48], $M^{[1:k]} \in \mathbb{R}^{d \times d \times k}$ is a tensor, k is the number of slices, and $v_r \in \mathbb{R}^k$ integrates the results of each slice. The others are the standard form of a neural network: $V_r \in \mathbb{R}^{k \times 2d}$ and $b_r \in \mathbb{R}^k$ are learnable parameters.

The match score between the dialogue history and the i^{th} persona element is computed by:

$$
s_i = softmax(o_i^b - \Sigma_{j=1}^m o_{ij}^a). \quad (27)
$$

The matched persona element index is obtained with the maximal score, i.e., $U^{\hat{p}} = U_i^P$, where $i = \arg\max_i(s_i)$.

Algorithm 2. Pseudo Labeling.

```
Data: current step $k$, target response $U_k^R$,
      persona descriptions $U^P$, persona
      descriptions size $|U^P|$, threshold $\theta$,
      pre-trained RoBERTa-NLI model
      $RoBERTa$
Result: intent label $Y^P$
for $i \leftarrow 1$ to $|U^P| + 1$ do
    $x_k^r \leftarrow RoBERTa(U_k^R)$
    $x_i^P \leftarrow RoBERTa(U_i^P)$
    $scr_{ki} \leftarrow cos\_sim(x_k^r, x_i^P)$
end
$scr_{max}, i_{max} \leftarrow \arg\max_i scr_{ki}$
if $scr_{max} \geq \theta$ then
    $Y^P \leftarrow i_{max}$
else
    $Y^P \leftarrow 0$
end
```

G. Pseudo Labeling

Given the unavailability of a persona-driven dialogue dataset that contains interlocutor intent labels, we resort to pseudo labeling [49], a semi-supervised learning (SSL) approach, to automatically generate the intent label Y^P. RoBERTa-NLI [37] is used as an automatic annotator to generate the intent label according to Algorithm 2.

Taking the k^{th} turn of a dialogue as an example, we can obtain the target response U_k^R and persona descriptions U^P. The persona descriptions U^P keep unchanged in the same dialogue but vary among different dialogues. The utterance-level embeddings for the target response U_k^R and each persona element x_i^P are obtained from the pre-trained RoBERTa-NLI model, $RoBERTa(\cdot)$. The cosine similarity is computed for each (U_k^R, x_i^P) pair, and the maximum value scr_{max} with the index i_{max} among them is determined. If scr_{max} exceeds the threshold θ, i_{max} is used as the pseudo intent label Y^P. Otherwise, 0 is set as the pseudo intent label Y^P, which means the interlocutor expects persona-irrelevant responses or messages in the current context. *cos_sim* in Algorithm 2 denotes the cosine similarity function.

TABLE I Statistics of PersonaChat dataset.

DATASET	PERSONA	DIALOGUE	UTTERANCE
Train	955	8,939	131,438
Valid	100	1,000	15,602
Test	100	968	15,024
Total	1,155	10,907	162,064

H. Training and Optimization

Generation part. The training loss for the generator is defined as a negative log-likelihood of the target sequence:

$$\mathcal{L}_g = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{L}\sum_{t=1}^{L} - logp(y_{it} = w_{it}^{*}|U^H, U^P, y_{i,<t})\right), \tag{28}$$

where L is the length of the target sequence. N is the number of samples (i.e., dialogue turns). w_{it}^{*} is the target word in i^{th} sample at time step t.

The auxiliary persona intent tracker is trained by the cross-entropy listwise loss function $\mathcal{L}_t$, which is a listwise approach to learning to rank [50]:

$$\mathcal{L}_t = CrossEntropy(\hat{Y}_m^P, Y_m^P). \tag{29}$$

The loss for the generation task is the weighted sum of the two parts:

$$\mathcal{L}_G = \alpha_g \mathcal{L}_g + \alpha_t \mathcal{L}_t, \tag{30}$$

where α_g and α_t are hyper-parameters.

Intent inference part. The binary intent classifier in the intent inference part is trained by the cross-entropy loss:

$$\mathcal{L}_b = CrossEntropy(\tilde{Y}_b^P, Y_b^P). \tag{31}$$

The persona intent retriever in the intent inference part is trained on the basis of multi-task learning (BIC and PIT). The loss of PIT is defined as the LambdaLoss [50], which is another listwise approach to learning to rank.

$$\mathcal{L}_p = LambdaLoss(\tilde{Y}_m^P, Y_m^P) \tag{32}$$

Therefore, the loss of the intent retriever is defined as:

$$\mathcal{L}_R = \alpha_b \mathcal{L}_b + \alpha_p \mathcal{L}_p, \tag{33}$$

where α_b and α_p are hyper-parameters.

IV. Experiment

A. Datasets

Persona-Chat Persona-driven dialogue generation experiments were conducted on a recently released dataset PersonaChat [10]. The dialogues were collected in a crowdsourced generation manner, where randomly paired crowdworkers were asked to interact according to the given persona.

The dataset has 162,064 utterances over 10,907 dialogues and has a set of 1,155 personas. Each sample contains a dialogue history of up to 15 utterances, a target response and a persona consisting of four to five persona descriptions. The detailed statistics of the training, validation, and testing sets can be viewed in Table I.

B. Baselines

- **S2SA** [51] is a Seq2Seq dialogue generation model with an attention mechanism [52]. Besides, it uses both bag-of-words and sentences as targets.
- **Per-S2SA** is a variant of the S2SA [51] model which prepends all persona texts to the input utterance or message. The prepending operation is similar to that in the Seq2Seq baseline model in [10].
- **GPMN** [10] is a Generative Profile Memory Network that encodes persona as individual memory representations in a memory network.
- **Transformer** [53] is one of the state-of-the-art sequence transduction models. In the experiment, persona texts and messages are concatenated as the input.
- **REGS** [54] (Reward for Every Generation Step) is an adversarially trained model using Monte Carlo search for response generation. Persona texts are used as context information when training this model.
- **DeepCopy** [31] is a hierarchical pointer network that extends the pointer-generator to copy tokens from relevant persona descriptions.
- **RCDG** [1] is a Reinforcement Learning-based Consistent Dialogue Generation model. It regards ranking retrieved responses as a reinforcement task and exploits NLI signals from response-persona pairs as rewards to improve the response consistency of dialogue agents.
- **TransferTransfo** [55] is a single-input OpenAI GPT, which uses token type embedding to differentiate different parts of a single concatenated input, e.g., persona description, historical conversation, and corresponding response. In the experiment, GPT2 [56] replaces the original GPT, which is denoted as **TransferGPT2**.
- **MI-GPT** [57] also employs the OpenAI GPT in both the encoder and the decoder, where average pooling is used as the attention fusion method.
- **GPT2-MAF** [14] employs a pre-trained OpenAI GPT2 model for persona-driven dialogue generation. The results given by the optimal fusion methods are reported.

C. Setups

In the generation part, all the employed utterance encoders are two-layer BiGRUs; the global context encoder is a two-layer forward GRU; the decoder uses a single-layer forward GRU. The hidden state size of the above encoders and the decoder is 512. Embeddings of size 300 were randomly initialized and updated during the training process. The vocabulary size is 20348. The model parameters were optimized using Adam [58] with an initial learning rate of 0.0003. The learning rate was decayed with a cosine annealing [59]. The training batch size is 64. The hyper-parameter weights were set as $\alpha_g = 1$ and $\alpha_t = 0.5$. Besides, hyper-parameters were set to $\tau_b = 0.15$ and

$\tau_m = 0.35$ in random disguising. In the intent inference part, an ALBERT-small [40] was used for the context-aware utterance embedding in the intent detector module. The experimental setup followed [60] for the intent detector. The hyper-parameters were set to $\alpha_b = 1$ and $\alpha_p = 1$. The average results of five runs are reported. The p-values of automatic evaluation results are below 0.05 significance level.

D. Evaluation Metrics

Automatic evaluation metrics and human evaluation were employed to evaluate the quality of generated responses.

Automatic Evaluation. Although there is no universally applicable metric for evaluating the quality of generated responses, several indicators, such as BLEU and embedding metrics, can be used to evaluate the relevance between the generated and the ground-truth response. All the automatic evaluation metrics range from 0 to 1. The results are reported in percentage.

BLEU [15]. BLEU measures the n-gram overlap between the ground truth and the generated response. A higher BLEU score indicates better generation quality, and vice versa. The BLEU score computed by mteval-v14.pl[1] is reported.

METEOR [16]. METEOR is based on the harmonic mean of the unigram precision and recall, where recall is weighted higher than precision. Different from BLEU, the unigram alignment between a reference and a generated sentence also considers matching results after Porter stemming [61] and WordNet synonymy [62], besides exact matches.

ROUGE-L [17]. ROUGE-L represents the recall-oriented understudy of gisting evaluation based on the longest common subsequence. It considers sentence-level structure similarity and identifies the longest co-occurring in sequence n-grams.

TABLE II Evaluation results based on n-gram automatic metrics.

MODEL	BLEU	METEOR	ROUGE-L	F1
S2SA	3.373	10.162	19.219	22.123
Per-S2SA	3.266	10.292	18.962	21.911
TransferTransfo	2.054*	7.672*	-	-
MI-GPT	3.151*	8.112*	-	-
TransferGPT2	3.597	11.379	19.054	18.899
GPT2-MAF	4.147*	8.988*	19.428	22.619
HIPPL$_{lstm}$	4.236	12.483	**21.494**	**24.429**
HIPPL$_{albert}$	**4.298**	**12.532**	21.275	24.244

* The results of the models are reported in [14]. Underlined results are the best among baselines but inferior to HIPPL.

F1 [18]. F1 score computes the harmonic mean of the unigram precision and recall in natural language generation [63].

Embedding Metrics. According to [64], embedding average (Avg.), embedding greedy (Grd.), and embedding extrema (Ext.) are employed as evaluation metrics to measure the quality of generated dialogues. The scores of these metrics depend on word embeddings. They are the measurement of the relevance between a generated response and a target response. GloVe [65] 100D word vectors are used in this experiment.

Human Evaluation. We invited three English-speaking participants to conduct human evaluation. Judges were employed to score 150 dialogue turns that were randomly sampled from the generated responses from three different aspects.:

Contextual coherence. A response is contextual-coherent if it is systematically or logically connected with the dialogue context. The range of the contextual coherence score is $\{1, 2, 3\}$. A higher score means higher contextual coherence.

Inverse duplicate. Inverse duplicate measures whether the generated response is duplicated with the dialogue history or not. The corresponding score is 0 (duplicated) and 1 (not duplicated), respectively.

Persona consistency. This metric measures whether the response contains persona information that should be consistent with the pre-defined persona. The generated response is scored 1 if meeting the condition, and 0 otherwise.

V. Result

A. Automatic Evaluation for Dialogue Generation

Tables II and III show the results of HIPPL and baseline models in n-gram-based metrics and embedding metrics, respectively. The last two rows are the results of HIPPL. HIPPL$_{albert}$ refers to HIPPL in this paper; HIPPL$_{lstm}$ is a variant where the ALBERT is replaced with a BiLSTM in the binary intent classifier. As shown in Table II, our lightweight model HIPPL$_{lstm}$ outperforms all the baselines in n-gram automatic metrics. Compared with the strongest baselines, HIPPL$_{lstm}$ yields 0.089 BLEU gains over GPT2-MAF, 1.104 METEOR gains over TransferGPT2, 2.066 ROUGE-L gains over GPT2-MAF, and 1.800 F1 gains over GPT2-MAF. Noticeably, most of these baselines are developed upon PLMs, i.e., GPT and GPT2. Whereas HIPPL$_{lstm}$ is based on the GRU and the BiERU. Compared with non-pre-trained language model-based methods, e.g., S2SA, HIPPL$_{lstm}$ demonstrated larger gains in BLEU, METEOR, ROUGE-L, and F1, achieving improvements of 0.863, 2.321, 2.275, and 2.306, respectively. When we employed a lightweight PLM, ALBERT, instead of a BiLSTM for the utterance embedding in the intent detector module (IntentDt), further improvements in BLEU (+0.062)

[1] [Online]. Available: https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/mteval-v14.pl (Accessed on Dec. 2021)

TABLE III Evaluation results on embedding metrics.

MODEL	GRD.	AVG.	EXT.
DeepCopy[a]	43.2	62.1	45.1
GPMN[a]	45.7	65.3	43.2
REGS[a]	44.2	64.3	44.8
S2SA	65.18	67.99	51.23
Per-S2SA	65.12	67.79	51.54
Transformer[a]	43.9	63.4	43.6
RCDG$_{bert}$[a]	47.2	66.9	46.8
TransferGPT2	64.79	66.34	50.78
GPT2-MAF	65.15	68.47	51.26
HIPPL$_{lstm}$	**66.01**	67.48	**52.39**
HIPPL$_{albert}$	65.94	67.73	52.35

[a] The results are reported in [1]. Underlined results are the best among baselines but inferior to HIPPL (except for Avg.).

TABLE IV Human evaluation results.

MODEL	C-CHR.	IN-DPL.	P-CNS.
S2SA	1.9667	0.8767	0.6067
Per-S2SA	2.0467	0.8533	0.6933
TransferGPT2	2.2600	0.8433	0.6700
GPT2-MAF	2.1167	0.8633	0.6367
HIPPL	**2.3567**	**0.8967**	**0.7867**

Note: c-chr., in-dpl., p-cns. are the abbreviation for contextual coherence, inverse duplicate score, and persona consistency, respectively.
Underlined results are the best among baselines but inferior to HIPPL.

TABLE V IntentDt results in interlocutor intent inference task.

MODEL	NUM. OF PARAM.	ACCURACY
Binary intent classifier (BiLSTM-based)	1.16M	0.7314
Binary intent classifier (ALBERT-based)	12.86M	0.8353
Multi-task intent retriever	2.32M	0.5156

and METEOR (+0.051) can be observed in the comparison between HIPPL$_{lstm}$ and HIPPL$_{albert}$.

In embedding metrics, HIPPL$_{lstm}$ achieves better results than GPT2-MAF in greedy and extrema evaluation dimensions, yielding an average gain of 1.00. In embedding average, HIPPL$_{lstm}$ is slightly lower than GPT2-MAF. Compared with GPT2-MAF, the average gain of HIPPL$_{lstm}$ is 0.33in the three embedding metrics. Employing ALBERT (HIPPL$_{albert}$) delivers slight extra gains in embedding average.

In addition, our model is more parameter-efficient than the PLMs. The numbers of parameters of TransferGPT2 and GPT2-MAF are 124M and 327M, respectively. While our model HIPPL$_{albert}$ has 75M parameters (59.94M in generation + 15.19M in intent inference); HIPPL$_{lstm}$ has 63M parameters (59.94M in generation + 3.49M in intent inference). Thus, the parameter size of HIPPL$_{lstm}$ is approximately only 20% of that of GPT2-MAF (the strongest baseline), while HIPPL$_{lstm}$ performs better in diverse automatic evaluation metrics.

B. Human Evaluation for Dialogue Generation

Considering the code availability of each method, we selected S2SA, per-S2SA, TransferGPT2, and GPT2-MAF as baseline models. The human evaluation results are shown in Table IV. The Fleissâ Kappa [66] was calculated to measure the inter-rater consistency. The Fleissâ Kappa for contextual coherence, inverse duplicate, and persona consistency are 0.4266, 0.8620, and 0.4579, indicating *moderate agreement*, *almost perfect*, and *moderate agreement*, respectively. This is because contextual coherence and persona consistency metrics are more subjective than inverse duplicate. It is inherently harder to obtain perfect agreements in the two aforementioned metrics compared to inverse duplicate.

It is found that our model outperforms the baseline models in c-chr., in-dpl., and p-cns. scores. This means the responses from HIPPL are more contextual-coherent and less duplicated with the dialogue history, and they contain more persona information that is consistent with the pre-defined persona. HIPPL outperforms TransferGPT2 in c-chr. (+0.0967), surpasses S2SA in in-dpl (+0.0200), and exceeds Per-S2SA in p-cns. (+0.0934). TransferGPT2 tends to generate contents that are duplicated with the dialogue history. It diminishes the coherence of the responses, resulting in monotonous human-computer interactions. S2SA avoids the duplication through trivial responses, e.g., *i am not sure* or *i do not have a lot of time*. However, they are usually not appropriate for the current context and do not contain persona information, which may make the interlocutor lose interest or feel offended. Per-S2SA is better at generating persona-consistent responses, but it suffers from duplicated or incoherent contents.

C. Interlocutor Intent Inference

The performance of our IntentDt was reported in Table V for the III task. The accuracy for the binary intent classifier measures the closeness of the predicted

TABLE VI Ablation study of HIPPL.

MODEL	GLOBAL ENC.	PRD. INT.	P. CMP.	FULL P.	RND. DSG.	ORACLE	BLEU
M_1	0	1	1	0	1	0	3.437
M_2	1	0	0	1	1	0	3.099
M_3	1	1	0	0	1	0	3.36
M_4	1	1	1	0	0	0	3.82
M_5	1	1	0	0	0	1	4.17
M_6	1	1	1	0	0	1	3.52
HIPPL	1	1	1	0	1	0	4.185

Note: global enc., prd. int., p. cmp., full p., and rnd. dsg. are abbreviations for the global encoder, the predicted intent, persona complements, full persona descriptions, and random disguising.

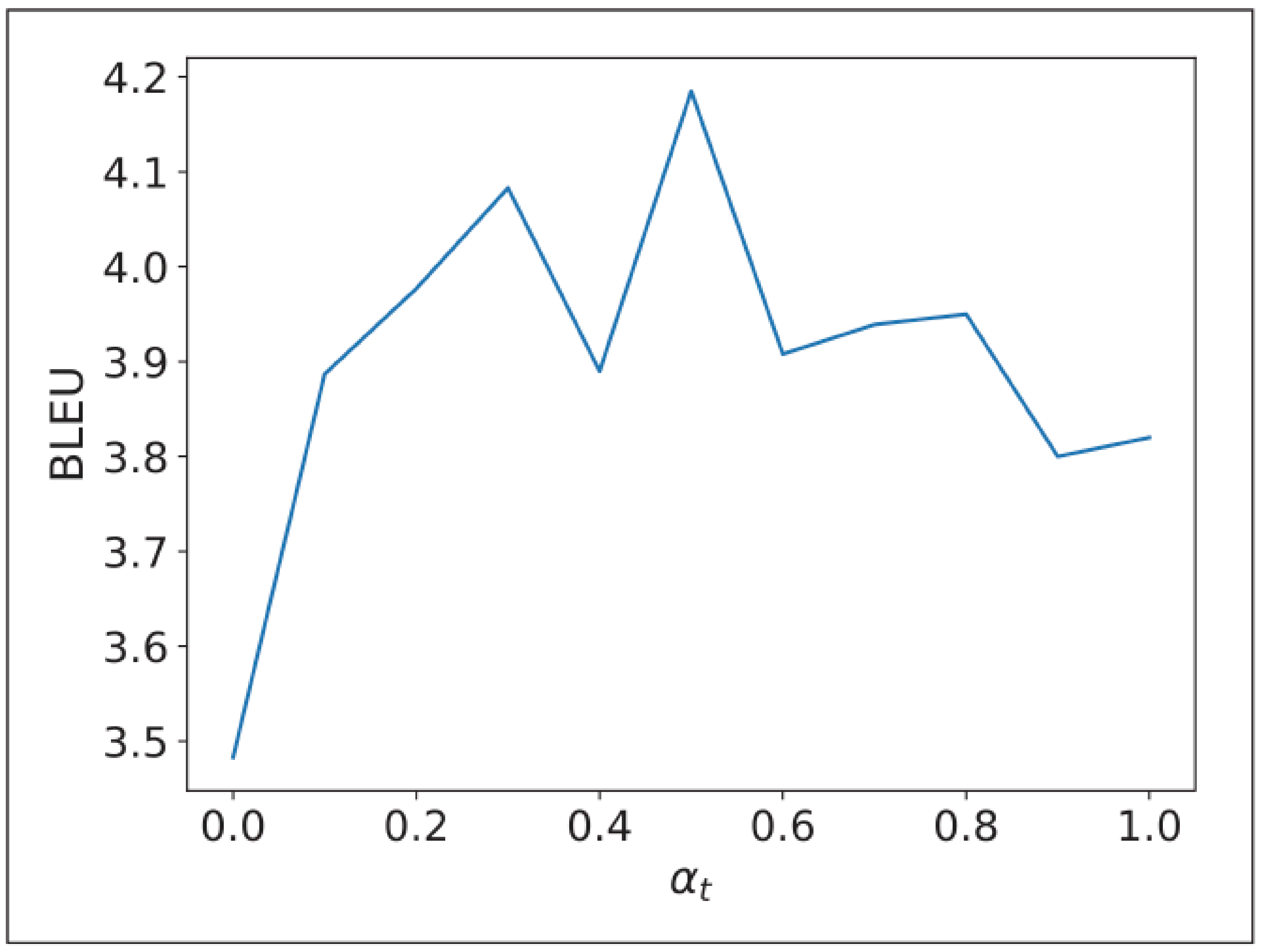


FIGURE 4 Parameter analysis of α_t.

intent $\tilde{Y}_b^P$ to the pseudo label Y_b^P; the accuracy for the multi-task intent retriever measures how close the predicted intent $\tilde{Y}_m^P$ is to the pseudo label Y_m^P.

In Section III-D, the ALBERT-based binary intent classifier was introduced. The BiLSTM-based binary intent classifier differs from the ALBERT-based classifier by replacing the ALBERT with a BiLSTM for generating the utterance-level embedding. This is also the only difference between HIPPL$_{lstm}$ and HIPPL$_{albert}$ in this paper. The ALBERT-based binary intent classifier surpasses the BiLSTM-based classifier by 0.1039in accuracy with more parameters (+11.70M).

The multi-task intent retriever trained via learning to rank is designed to select the persona intent from persona descriptions U^P. As mentioned in Section III-D, the retriever generates a multi-class label $\tilde{Y}_m^P$ to achieve this. Thus, the accuracy of 0.5156 suggests that the multi-task intent retriever performs well in the multi-class classification task.

D. Ablation Study

Table VI shows the ablation study results of HIPPL. The last row shows HIPPL performance in BLEU with $\alpha_t = 0.5$, $\tau_b = 0.05$, $\tau_m = 0.3$ and other default settings. Models $M_1 \sim M_6$ are the variants of HIPPL with different architectures. Oracle denotes the model that uses RoBERTa-NLI generated pseudo labels Y^P for testing. It demonstrates the upper limits of the model capacity with the corresponding architecture.

Comparing M_1 with HIPPL, we found the global encoder improves the performance by extracting the global context from the dialogue history, providing a long-term memory. The results of M_2 and M_3 indicate the positive influence of III. The difference between M_2 and HIPPL lies in their use of attention on the predicted interlocutor intent and persona complements either separately or jointly. The results show that HIPPL surpasses M_2 as it can independently focus on the useful information from the predicted interlocutor intent and persona complements. The result of HIPPL is better than that of M_3, since the attention mechanism on persona complements in MSPG provides extra knowledge.

Given the absence of ground-truth labels for intent inference training, two methods were employed to mitigate the impact of incorrect pseudo labels: 1) a soft gating mechanism on persona complements in MSPG, and 2) random disguising. Comparing M_4 and HIPPL, we observed that random disguising mitigates the errors from pseudo labeling, yielding a higher BLEU in the generation task. The gap between M_5 and HIPPL is smaller than that between M_4 and HIPPL, because M_5 is an oracle model utilizing pseudo labels. M_5 does not use persona complements, consequently exhibiting an excessive dependence on the intent given by pseudo labeling. The superior performance of HIPPL over M_5 suggests that persona complements and random disguising mitigate the impact of pseudo labeling errors. There is a significant improvement in BLEU for HIPPL, compared with M_6, which can be attributed to the use of random disguising for training the soft gating mechanism on persona complements.

E. Parameter Analysis

Hyper-parameter analysis. Figure 4 shows the performance of HIPPL in BLEU with different values of α_t. Here, $\tau_b = 0.05$, $\tau_m = 0.3$, and others were set as default. It is found that when $\alpha_t = 0$, the model obtains the lowest BLEU. This means the auxiliary task, i.e., the training of IntentTr, improves the performance of HIPPL. As α_t changes from 0 to 0.5, there is a

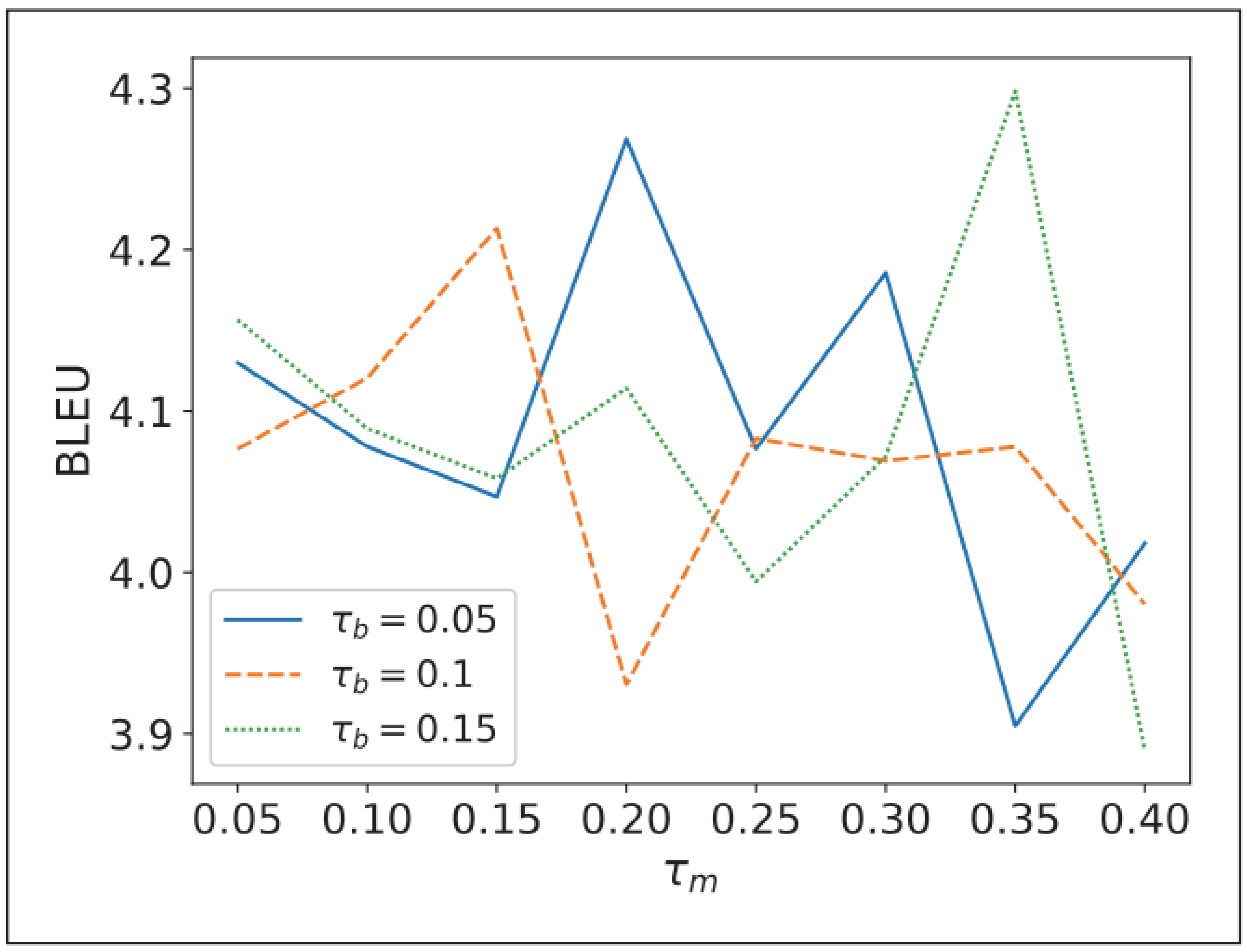
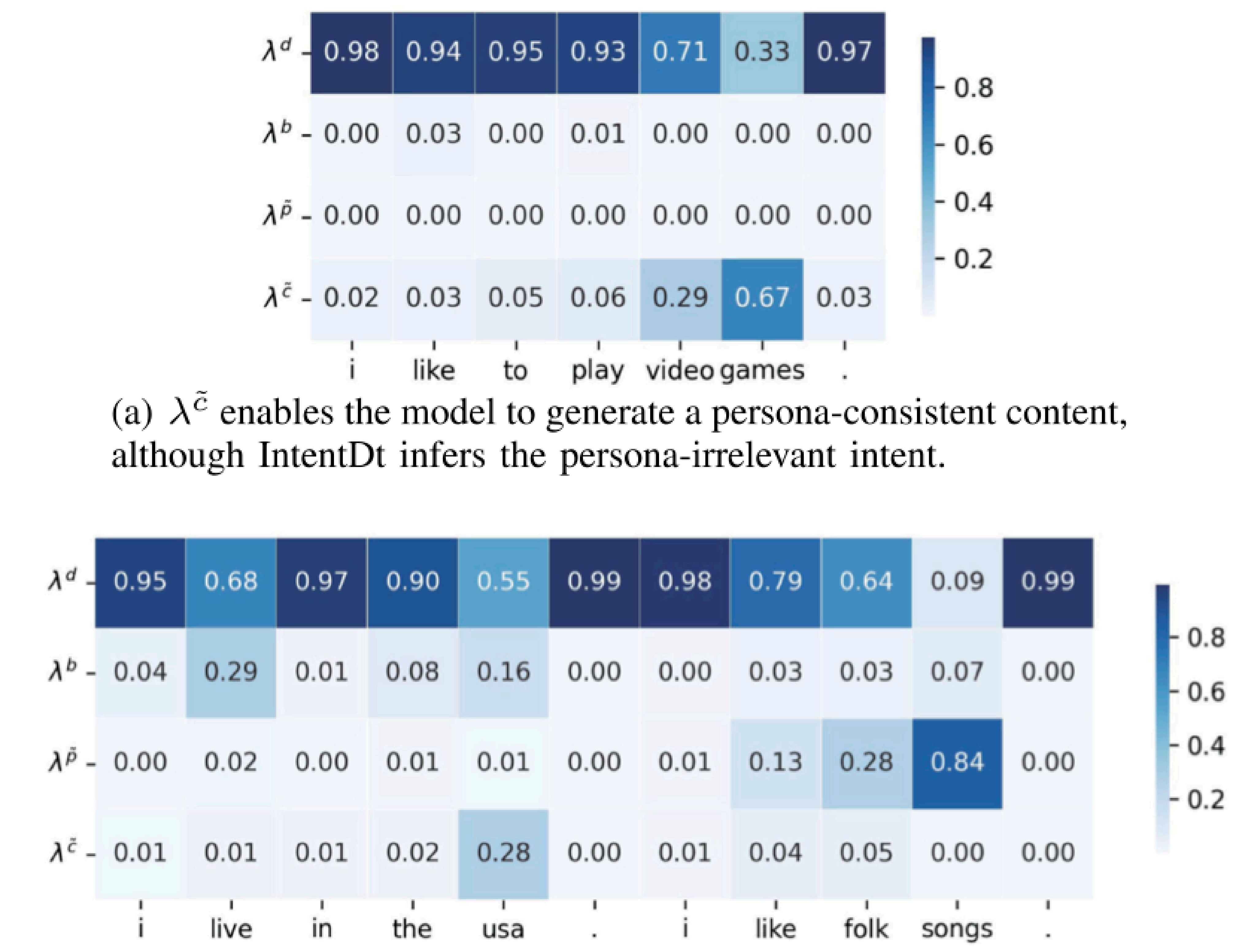


FIGURE 5 Parameter analysis of τ_b, τ_m.



(a) $\lambda^{\tilde{c}}$ enables the model to generate a persona-consistent content, although IntentDt infers the persona-irrelevant intent.

(b) λ^d, $\lambda^{\tilde{p}}$ enriches the generated response; λ^b enables the model to generate a contextual coherent response although IntentDt infers a wrong intent.

FIGURE 6 Visualization of λ.

general uptrend in BLEU. It reaches the highest score when $\alpha_t = 0.5$. As α_t changes from 0.5 to 1.0, there is a general downtrend in BLEU. Figure 5 illustrates the BLEU changes of HIPPL with different values of τ_b and τ_m, where others were set as default. τ_b and τ_m control the probability of a pseudo intent label Y^P being disguised. It is found that the volatility of $\tau_b = 0.05$ and $\tau_b = 0.15$ is much higher than that of $\tau_b = 0.1$ in BLEU. Comparing the results across different lines at each τ_m, we observed that the BLEU varies significantly when τ_b changes with $\tau_m = 0.15$, 0.2, or 0.35. This fluctuation shows a general uptrend as τ_m increases. In addition, the best performance is achieved at $\tau_b = 0.15$ and $\tau_m = 0.35$.

λ visualization. Given the absence of ground-truth labels to train a robust IntentDt for the III task and the difference between the pseudo labels and the predicted intent labels by IntentDt, pseudo labeling and random disguising were utilized to train a noise-resistant MSPG with a trade-off in the accuracy of fed pseudo labels. Specifically, λ in MSPG (see Eq. 23) was trained to mitigate the accuracy loss in IntentDt. The motivation for introducing λ is that λ can complement the response generation from multiple input sources, even if the IntentDt delivered wrong actions and intents.

Figure 6 illustrates the visualization of computed weights λ for different source inputs in two different cases. The detailed information of the two cases can be viewed in Table VII. In Exp. (a) of Table VII, the true intent U^p is related to *playing video games* while IntentDt predicts a persona-irrelevant intent. However, $\lambda^{\tilde{c}}$ enables the model to generate persona-consistent responses by copying the relevant content from persona complements $U^{\tilde{c}}$. In Exp. (b) of Table VII, the true intent U^p should be irrelevant to given persona descriptions, whereas IntentDt predicts an incorrect persona intent (*i like to sing folk songs*). Nevertheless, λ^b assists the model in focusing on the local topic, i.e., living address, and λ^d allows the model to select *USA* from the vocabulary. In addition, $\lambda^{\tilde{p}}$ enriches the response by utilizing the wrong persona intent (*folk songs*).

Visualization of examples in Figure 6 indicates the computed weights λ work as a soft gating mechanism, allowing the model to switch among multiple source inputs. It facilitates the model in attaining a judicious response, notwithstanding the potential transmission of errors from IntentDt to MSPG. It also improves the generalization capacity and enriches the response contents.

F. Case Study

Some examples generated by S2SA, Per-S2SA, TransferGPT2, GPT2-MAF, and HIPPL are presented in Table VIII and Table IX. In general, our model is more capable of generating context-coherent and persona-consistent responses, in comparison to baseline models. In

TABLE VII Examples of generated responses from MSPG with multi-source inputs.

Exp. (a)	$U^{(p)}$	i love playing video games.
	$U^{(\bar{p})}$	persona-irrelevant intent
	$U^{(\bar{c})}$	hey there my name is Jordan and i am a veterinarian. i am originally from california but i live in florida. i love playing *video games*. love to read drama books.
	$U^{(b)}_{T-1}$	what do you like to do in your free time ?
	Response	i like to play *video games*.
Exp. (b)	$U^{(p)}$	persona-irrelevant intent
	$U^{(\bar{p})}$	*i like to sing folk songs.*
	$U^{(\bar{c})}$	i adopted a stray cat recently. i like making my own clothing. my mother was an army nurse.
	$U^{(b)}_{T-1}$	hi my name is lauren, and i live in new jersey. where do you live?
	Response	i live in the USA. *i like to sing folk songs*.

Note: Underlined contents are context-coherent. Italic ones use pre-defined personas.

TABLE VIII Generated dialogue responses by S2SA, Per-S2SA, TransferGPT2, GPT2-MAF, and HIPPL (Case I).

ITEM	TEXT	C-CHR	P-CNS
Persona	*i like to eat muffins.* *i love film.* *i work in the healthcare industry.* i have a dog named pedro.		
Dialog history	... b: better now that the bug is gone and my bacon is done cooking agent: i do not like bugs either but bacon is good b: what are you up to tonight ?		
Human	muffins are my favorite food.	2	1
S2SA	i am going to be a doctor.	1	0
Per-S2SA	i am going to go to the gym. i love it.	3	1
TransferGPT2	i am *watching a movie*.	3	1
GPT2-MAF	i am *working in the healthcare industry* right now.	2	1
HIPPL	i am going to *eat muffins*.	3	1

Note: Underlined contents are contextually coherent and italic ones use pre-defined personas.

Table VIII, S2SA yields the lowest scores in c-chr and p-cns. Per-S2SA generates a response that is contextual-coherent. Although it is irrelevant to the pre-defined personas, they are not contradictory. The response from GPT2-MAF is persona-consistent. However, it is not coherent with the local topic. In Table IX, *i am sorry to hear that.* from Per-S2SA is not appropriate for the local topic and *what do you do for a living?* is consistent but irrelevant to the pre-defined personas. Responses from S2SA, TransferGPT2, and GPT2-MAF all contradict the pre-defined persona in Table IX. Moreover, the response from TransferGPT2 contradicts itself, achieving the lowest c-chr.

In contrast, our HIPPL model yields context-coherent and persona-consistent responses in the two dialogues. For example, given a question *what are you up to tonight?* from Speaker b in Table VIII, the HIPPL replies *i am going to eat muffins*, where *i like to eat muffins* is one of the given personas. Given *i am doing alright but i really wish i had kids* from the agent in Table IX, and *i do not have kids either, i have a diet company* from Speaker b, the HIPPL responses *i want to be a fashion designer*. This is very similar to the human response, i.e., *oh okay cool i want to be a fashion designer*. Moreover, HIPPL possesses the capacity to selectively determine a suitable persona element, e.g., *i want to be a fashion designer* from the persona description set, which is in accordance with the context of a career topic.

VI. Discussion

According to the experiments, IntentDt trained via pseudo labeling and random disguising contributed to the good performance of generated responses in terms of BLEU. However, there is still room for the improvement of predicted intents. First, IntentDt is a two-stage module consisting of a binary intent classifier and a multi-task retriever. Thus, an end-to-end intent detector module could be proposed to reduce the accumulated biases in two-stage settings. Second, the IntentDt predicted interlocutor intents based on the concatenated dialogue history without an in-depth analysis of the interactions between utterances. However, even the utterances at the very beginning of the conversation may also influence the topic or speaker's intent. Hence, a dependency-aware context representing module [67] would benefit intention detection.

Furthermore, RoBERTa-NLI might not be an ideal solution to pseudo labeling since its design was primarily tailored for document-level NLI. Thus, alternatives could be employing an NLI model that is specifically trained on persona-based chitchat data or integrating external knowledge graphs to improve the precision of pseudo labels.

TABLE IX Generated dialogue responses by S2SA, Per-S2SA, TransferGPT2, GPT2-MAF, and HIPPL (Case II).

ITEM	TEXT	C-CHR	P-CNS
Persona	*i want to be a fashion designer.* i was arrested for stealing. i want children badly but not right time in my life. my parents were always gone when i was growing up.		
Dialog history	b: hello, how are you ? my cats and dogs are fine. agent: i am doing alright but i really wish i had kids... b: i do not have kids either, i have a diet company.		
Human	oh okay cool i want to be a fashion designer.	3	1
S2SA	that is cool. i am a teacher.	3	0
Per-S2SA	i am sorry to hear that. what do you do for a living ?	1	1
TransferGPT2	i am a *fashion designer.* i want to be a fashion designer.	1	0
GPT2-MAF	that is cool. i am a *fashion designer.*	3	0
HIPPL	i want to be a *fashion designer.*	3	1

Note: Underlined contents are contextually coherent and italic ones use pre-defined personas.

VII. Conclusion

Persona consistency and contextual coherence are two key challenges in persona-driven dialogue generation tasks. To this end, a lightweight hierarchical intent-inferring pointer network is put forth for multi-source multi-turn consistent persona-driven dialogue generation. The parameter size of our model is only approximately 20% of that of the state-of-the-art GPT-2-based model. Notably, our model demonstrated superior performance in diverse evaluation metrics. To assist the persona-driven dialogue agent, the model also involves training a classifier with pseudo labels for interlocutor intent inference.

Experimental results indicate that the hierarchical interlocutor encoder extracting global context improves model performance in BLEU. The interlocutor intent inference improves the quality of generated responses from the dialogue agent. The random disguising mechanism makes the model more resistant to noise. And the multi-source pointer-generator enhances the generalization capacity of the model by employing a soft gating mechanism and a multi-source attention among different source inputs.

Acknowledgment

This work was supported in part by the Agency for Science, Technology and Research (A*STAR) through AME Programmatic Funding Scheme under Project A18A2b0046, in part by the Ministry of Education, Singapore under its MOE Academic Research Fund Tier 2 (STEM RIE2025 Award MOE-T2EP20123-0005), and in part by the RIE2025 Industry Alignment Fund—Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore.

References

[1] H. Song, W.-N. Zhang, J. Hu, and T. Liu, "Generating persona consistent dialogues by exploiting natural language inference," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 05, pp. 8878–8885.

[2] G. Tu, J. Wen, C. Liu, D. Jiang, and E. Cambria, "Context-and sentiment-aware networks for emotion recognition in conversation," *IEEE Trans. Artif. Intell.*, vol. 3, no. 5, pp. 699–708, Oct. 2022.

[3] Y. Wang, P. Si, Z. Lei, and Y. Yang, "Topic enhanced controllable CVAE for dialogue generation (student abstract)," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 10, pp. 13955–13956.

[4] L. Li, C. Xu, W. Wu, Y. Zhao, X. Zhao, and C. Tao, "Zero-resource knowledge-grounded dialogue generation," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 8475–8485.

[5] H. H. Clark and D. Wilkes-Gibbs, "Referring as a collaborative process," *Cognition*, vol. 22, no. 1, pp. 1–39, 1986.

[6] M. J. Pickering and S. Garrod, "Toward a mechanistic psychology of dialogue," *Behav. Brain Sci.*, vol. 27, no. 2, pp. 169–190, 2004.

[7] Y. Ma, K. L. Nguyen, F. Z. Xing, and E. Cambria, "A survey on empathetic dialogue systems," *Inf. Fusion*, vol. 64, pp. 50–70, 2020.

[8] O. Vinyals and Q. Le, "A neural conversational model," in *Proc. Int. Conf. Mach. Learn. Deep Learn. Workshop*, 2015.

[9] J. Li, M. Galley, C. Brockett, G. Spithourakis, J. Gao, and W. B. Dolan, "A persona-based neural conversation model," in *Proc. Assoc. Comput. Linguistics*, 2016, pp. 994–1003.

[10] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston, "Personalizing dialogue agents : I have a dog, do you have pets too ?," in *Proc. Assoc. Comput. Linguistics*, 2018, pp. 2204–2213.

[11] S. Welleck, J. Weston, A. Szlam, and K. Cho, "Dialogue natural language inference," in *Proc. Assoc. Comput. Linguistics*, 2019, pp. 3731–3741.

[12] S. Louvan and B. Magnini, "Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 480–496.

[13] C. Wang et al., "Mell: Large-scale extensible user intent classification for dialogue systems with meta lifelong learning," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2021, pp. 3649–3659.

[14] Y. Cao, W. Bi, M. Fang, and D. Tao, "Pre-trained language models for dialogue generation with multiple input sources," in *Proc. Findings Assoc. Comput. Linguistics*, 2020, pp. 909–917.

[15] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.

[16] A. Lavie and A. Agarwal, "Meteor: An automatic metric for MT evaluation with high levels of correlation with human judgments," in *Proc. 2nd Workshop Stat. Mach. Transl.*, 2007, pp. 228–231.

[17] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. Text Summarization Branches Out*, 2004, pp. 74–81.

[18] C. J. Van Rijsbergen, "Foundation of evaluation," *J. Documentation*, vol. 30, no. 4, pp. 365–373, 1974.

[19] I. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 3776–3783.

[20] J. Li, M. Galley, C. Brockett, J. Gao, and W. B. Dolan, "A diversity-promoting objective function for neural conversation models," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 110–119.

[21] M. Li et al., "Don't say that! making inconsistent dialogue unlikely with unlikelihood training," in *Proc. Assoc. Comput. Linguistics*, 2020, pp. 4715–4728.

[22] T. Young, F. Xing, V. Pandelea, J. Ni, and E. Cambria, "Fusing task-oriented and open-domain dialogues in conversational agents," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 10, pp. 11622–11629.

[23] J. Ni, V. Pandelea, T. Young, H. Zhou, and E. Cambria, "HiTKG: Towards goal-oriented conversations via multi-hierarchy learning," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 10, pp. 11112–11120.

[24] O. Tikhomirov, "The psychological structure of the man-computer dialogue," *Sov. Psychol.*, vol. 23, no. 4, pp. 24–37, 1985.

[25] T. Bickmore and J. Cassell, "Relational agents: A model and implementation of building user trust," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2001, pp. 396–403.

[26] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2692–2700.

[27] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," in *Proc. Int. Conf. Learn. Representations*, 2018.

[28] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, "Gated self-matching networks for reading

comprehension and question answering," in *Proc. Assoc. Comput. Linguistics*, 2017, pp. 189–198.
[29] M. Eric and C. D. Manning, "A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 468–473.
[30] F. Sun, P. Jiang, H. Sun, C. Pei, W. Ou, and X. Wang, "Multi-source pointer network for product title summarization," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 7–16.
[31] S. Yavuz, A. Rastogi, G.-L. Chao, and D. Hakkani-Tur, "Deepcopy: Grounded response generation with hierarchical pointer networks," in *Proc. 20th Annu. Meeting Special Int. Group Discourse Dialogue*, 2019, pp. 122–132.
[32] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. Assoc. Comput. Linguistics*, 2017, pp. 1073–1083.
[33] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 21–29.
[34] H. Fan and J. Zhou, "Stacked latent attention for multimodal reasoning," in *Proc. Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1072–1080.
[35] R. Mao, C. Lin, and F. Guerin, "End-to-end sequential metaphor identification inspired by linguistic theories," in *Proc. Assoc. Comput. Linguistics*, 2019, pp. 3888–3898.
[36] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proc. Empirical Methods Natural Lang. Process.*, 2017, pp. 670–680.
[37] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 3982–3992.
[38] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial attacks on deep-learning models in natural language processing: A survey," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 3, pp. 1–41, 2020.
[39] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. NeurIPS Workshop Deep Learn.*, 2014.
[40] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *Proc. Int. Conf. Learn. Representations*, 2019.
[41] W. Li, L. Zhu, and E. Cambria, "Taylor's theorem: A new perspective for neural tensor networks," *Knowl.-Based Syst.*, vol. 228, 2021, Art. no. 107258.
[42] R. Mao and X. Li, "Bridging towers of multitask learning with a gating mechanism for aspect-based sentiment analysis and sequential metaphor identification," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 13534–13542.
[43] R. Mao, X. Li, M. Ge, and E. Cambria, "MetaPro: A computational metaphor processing model for text pre-processing," *Inf. Fusion*, vol. 86-87, pp. 30–43, 2022.
[44] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 6645–6649.
[45] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Proc. Assoc. Comput. Linguistics*, 2016, pp. 1631–1640.
[46] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.
[47] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 926–934.
[48] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, 2013, vol. 30, no. 1, p. 3.
[49] D.-H. Lee et al., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Workshop Challenges Representation Learn.*, vol. 3, no. 2, 2013, p. 896.
[50] X. Wang, C. Li, N. Golbandi, M. Bendersky, and M. Najork, "The lambdaloss framework for ranking metric optimization," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 1313–1322.
[51] S. Ma, X. Sun, Y. Wang, and J. Lin, "Bag-of-words as target for neural machine translation," in *Proc. Assoc. Comput. Linguistics*, 2018, pp. 332–338.
[52] M.-T. Luong and C. D. Manning, "Stanford neural machine translation systems for spoken language domains," in *Proc. 12th Int. Workshop Spoken Lang. Transl.: Eval. Campaign*, 2015, pp. 76–79.
[53] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[54] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," in *Proc. Empirical Methods Natural Lang. Process.*, 2017, pp. 2157–2169.
[55] T. Wolf, V. Sanh, J. Chaumond, and C. Delangue, "Transfertransfo: A transfer learning approach for neural network based conversational agents," 2019, *arXiv:1901.08149*.
[56] A. Radford et al., "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019, Art. no. 9.
[57] S. Golovanov, R. Kurbanov, S. Nikolenko, K. Truskovskyi, A. Tselousov, and T. Wolf, "Large-scale transfer learning for natural language generation," in *Proc. Assoc. Comput. Linguistics*, 2019, pp. 6053–6058.
[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2014.
[59] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, 2017.
[60] W. Li, W. Shao, S. Ji, and E. Cambria, "BiERU: Bidirectional emotional recurrent unit for conversational sentiment analysis," *Neurocomputing*, vol. 467, pp. 73–82, 2022.
[61] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.
[62] G. A. Miller, "WordNet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.
[63] L. Derczynski, "Complementarity, f-score, and NLP evaluation," in *Proc. 10th Int. Conf. Lang. Resour. Eval.*, 2016, pp. 261–266.
[64] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, "How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation," in *Proc. Empirical Methods Natural Lang. Process.*, 2016, pp. 2122–2132.
[65] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.
[66] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychol. Bull.*, vol. 76, no. 5, 1971, Art. no. 378.
[67] W. Li, L. Zhu, R. Mao, and E. Cambria, "SKIER: A symbolic knowledge integrated model for conversational emotion recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 13121–13129.