IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 28, 2018, accepted January 6, 2019, date of publication January 14, 2019, date of current version February 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2892730

Hierarchical Attention and Knowledge Matching Networks With Information Enhancement for End-to-End Task-Oriented Dialog Systems

JUNQING HE[1,2], BING WANG[1,2], MINGMING FU[1,2], TIANQI YANG[1], AND XUEMIN ZHAO[1]
[1]Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 101408, China

Corresponding author: Junqing He (hejunqing@hccl.ioa.ac.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 11590770-4, Grant 61650202, Grant 11722437, Grant U1536117, Grant 61671442, Grant 11674352, Grant 11504406, and Grant 61601453, in part by the National Key Research and Development Program under Grant 2016YFB0801203, Grant 2016YFC0800503, and Grant 2017YFB1002803, and in part by the Foundation of Science and Technology on Information Assurance Laboratory under Grant KJ-17-102.

ABSTRACT Nowadays, most end-to-end task-oriented dialog systems are based on sequence-to-sequence (Seq2seq), which is an encoder-decoder framework. These systems sometimes produce grammatically correct, but logically incorrect responses. This phenomenon is usually due to information mismatching. To solve this problem, we introduce hierarchical attention and knowledge matching networks with information enhancement (HAMI) for task-oriented dialog systems. It contains a hierarchical attention dialog encoder (HADE) that models dialogs at the word and sentence level separately. HADE can focus on important words in a dialog history and generate context-aware representations. HAMI also contains a unique knowledge matching module to detect entities and interact with a knowledge base (KB). Then, dialog history and KB result representations serve as guidance for system response generation. Finally, HAMI's loss function is designed with an information regularization term to emphasize the importance of entities. The experimental results show that HAMI improves 9.8% in entity F1 compared with vanilla Seq2seq. HAMI also outperforms state-of-the-art models in both the entity F1 and dialog accuracy metrics.

INDEX TERMS Dialog systems, end-to-end response generation, context modeling, hierarchical attention, knowledge base interaction, neural networks, information enhancement.

I. INTRODUCTION

Task-oriented dialog systems (DS) like custom service agents and personal assistants are dialog systems that are designed for specific tasks such as restaurant reservation, weather inquiry, and ticket booking. They complete the tasks by interacting with users in natural language [1], [2]. There are two major approaches in developing dialog systems: pipeline-based method and end-to-end method [2]. The first approach builds a pipeline of modules to form a dialog system, including language understanding, dialog state tracking, policy management and natural language generation [3]–[5]. Its performance is relatively steady at the high cost of requiring labels for each module. There are joint learning and hybrid models to reduce the number of labels [2], [6], [7]. However, handcrafted templates or rules are necessary for these methods.

The second approach builds dialog systems in an end-to-end style that produces a sequence of output tokens directly without the modules above. It only needs text transcripts of dialogs without intermediate labels or handcrafted rules and can be adapted to different domains more easily than the first approach. Among all the end-to-end models, Seq2seq shows its potential in dialog systems in recent research [8], [9]. Seq2seq is an encoder-decoder model which is based on recurrent neural networks (RNN) and was invented for machine translation (MT) [10]. It has been widely used in open domain conversations [11]–[15] and end-to-end task-oriented DS [8], [9], [16], [17].

However, Seq2seq usually focuses on the coherence of word sequences and suffers from information deficiency. For example, the phrase ''address of restaurant_A'' can be

The associate editor coordinating the review of this manuscript and approving it for publication was Canbing Li.

TABLE 1. An example of Seq2seq output. Slot values are in italic bold.

User:	im looking for an ***expensive*** restaurant in the ***east*** part of town
Seq2seq:	sitar_tandoori is a nice place in the ***east*** of town serving tasty ***indian*** food
Gold Answer:	sitar_tandoori is a nice place in the ***east*** of town and the prices are ***expensive***

followed by a list of addresses, among which only one is correct. However, Seq2seq considers all of them as reasonable answers. Different from open domain conversations, it is fatal to task-completion DS since the provided information must be accurate. A similar example generated by Seq2seq is listed in Table 1.

The lack of incorporating in knowledge from dialog histories and KBs is the main reason for such a problem, since the slot values (pre-defined possible values of the slots in a task) are either mentioned in dialog histories or the provided KBs. Therefore, models that learn to carry information from dialog context were invented: attention-augmented Seq2seq [18], copy mechanism that helps to duplicate words from input [17], Transformer with self-attention [19] and memory-to-sequence (Mem2seq) that reasons over the dialog history and KB [20]. However, they encode dialog histories and KBs in a plain style. Besides, they match an encoded dialog and KB in the procedure of decoding each token. Namely, a dialog history and KB are searched for multiple times when producing a response, which is not in line with real scenarios. In reality, agents usually search the KB once. Then they determine what to say based on KB results and the dialog history before producing any single word. Based on this consideration, we summarize a dialog history and KB results separately before decoding a response.

In this paper, we aim to improve the information correctness of Seq2seq in various aspects. We propose Hierarchical Attention and knowledge Matching networks with Information enhancement (HAMI). HAMI models dialog histories using a hierarchical attention dialog encoder (HADE). It allows essential tokens to contribute more to the aggregated dialog representation. HAMI is able to search KBs using a novel and intuitive interaction method based on semantic matching. To emphasize the importance of information in system response, a penalty for predicting wrong entities is used in HAMI. We test our proposed model on two benchmark task-oriented datasets. Experimental results show that our approach achieves competitive performance, compared with a number of state-of-the-art methods. Our contributions are summarized as follows:

1) We propose HADE to distill essential information in dialog histories without any intermediate labels.
2) We design a novel and simple knowledge matching module. It first detects entities in a dialog history then match and merge the searching results into a comprehensive representation.
3) We add a penalty term in loss function to focus on the correctness of entities in generated results, while the ordinary cross-entropy loss only measures the correctness from the aspect of language quality.

The reminder of this paper is constructed as follows. Section II introduces related work. Section III presents preliminary of basic models. Section IV describes each module in HAMI in detail. The experimental results are given in section V. We have a thorough discussion about HAMI in Section VI. It is followed by our conclusions (section VII).

II. RELATED WORK

A. END-TO-END TASK-ORIENTED DIALOG SYSTEMS

Pipeline-based methods have been successfully applied to task-completion dialog systems [3]. However, they require labels for each module. Wen *et al*. [16] first proposed the concept of end-to-end task-oriented DS with traditional modules built by advanced neural networks based on Seq2seq framework. However, this system has a delexicalisation[1] step and utilizes a database operator, which are based on rules and are not trainable. Later on, hybrid code networks (HCN) was proposed to make modular outputs more controllable and reduce the need for training data. However, human-designed domain-specific rules and handcrafted templates are still incorporated in HCN [2]. In addition, it limits generated responses to specific templates. Lei *et al*. [9] designed a framework called Sequicity that simplifies task-oriented dialog systems into a two-stage Seq2seq model. It first generates dialog states then produces system responses using copy mechanism [21]. However, for some domains and conversation data, dialog states are hard to define and tag.

To solve this problem, producing dialog responses without intermediate labels becomes another trend for dialog systems. Memory Networks (MemNN) [22] was applied to DS as a mapping function from dialog histories to system responses [23]. Later on, gated Memory Networks (GMemNN) [24] and Query Reduction Networks (QRN) [25] were proposed based on MemNN to improve its reasoning ability. However, they cast the response generation task into an answer retrieval task. It is impossible for the models to generate new answers. Equipped with copy-augmented networks to duplicate proper words from user utterance, the Seq2seq framework without traditional modules was applied to task-completion dialog systems for the first time by Eric and Manning [17]. The same authors also proposed Key-Value retrieval networks based on Seq2seq to interact with KB by dynamically adding KB results to the vocabulary when decoding [26]. It matches the triples in a KB by soft attention between the hidden states of the decoder and the embeddings of subjects and relations. Then Mem2seq was proposed to improve the correctness of information retrieval in KB using memory network as dialog and KB encoder [20]. Besides database interaction, the generation procedure was also

[1]In this form of sentences, concrete slots and values are replaced with a specified mark.

considered by researchers. The task-oriented dialog systems were equipped with chatting capability by inserting chats in original data [8]. However, the representations of dialog history are quite straightforward and rarely discussed in the models above.

B. DIALOG HISTORY MODELING

Since response generation relies on contextual information, dialog history representation is of great significance. There are various approaches in modeling dialog histories for task-oriented DS. To model contextual states along with dialogs, Hierarchical Recurrent Encoder-Decoder (HRED) that added a layer of RNN to capture contextual states was proposed [14]. HRED is the most related model to our work for the hierarchical representation of sentences. Later, the Variant Hierarchical Recurrent Encoder-decoder that improve the diversity of generated sentences was developed based on HRED [15]. Certain improvements in dialog generation quality have been observed by adding stochastic noises. Also based on HRED, Multi-resolution Recurrent Neural Networks (MrRNN) was designed by the same team [27]. It uses nouns and entities as coarse representations of sentences in the encoder. However, the models above use the final state of RNN to summarize the whole dialog history and fail to concentrate on essential words. In the Sequicity framework that incorporates distinct dialog states tracking into a delexicalized sequence generation, a dialog history is represented as the final RNN state, given the combination of a dialog state and the current user utterance [9]. However, dialog state labels are unavailable in some data. In Mem2seq, previous turns of utterances are mapped to vectors called memory and a query vector is used to operate multi-hop attention with the memory [20]. The final operation result is the representation of dialog and used as the initial state of the decoder. It creatively stores utterance as embeddings to reduce the time for plain dialog encoding using RNN. However, positional information is not considered in Mem2seq.

In this paper, considering both different contributions and positional information of words in a dialog history, we propose HADE for dialog representation.

III. PRELIMINARY

Assume that X refers to a dialog input, $X = \{x_1, x_2, ..., x_T\}$ (x_i is a sentence, T is the number of sentences), which is a dialog history including current user utterance. Given X and probably a KB G containing triples in the form of (subject,relation,object), the task is to predict a correct output sequence of tokens $Y = \{y_1, y_2, ..., y_{T'}\}$ (y_j is a token, T' is the number of predicted tokens).

A. Seq2seq

In Seq2seq, all the utterances $x_1, x_2, ..., x_T$ in a dialog history X are concatenated sequentially. Namely, a dialog is encoded at the token level: $X = \{w_1, ..., w_n\}$ (w_i is the i token, n is the total number of tokens) using RNN as follows:

$$h_i = \text{RNN}(\phi_{emb}(w_i), h_{i-1}) \tag{1}$$

where ϕ_{emb} is a trainable word embedding matrix to map each token to a fixed-length vector. As for RNN, we can use bidirectional Long-short Term Memory [28] or GRU [29], as in previous literature [17], [18]. Then a RNN-based decoder is used to predict output tokens one by one.

$$h'_t = \text{RNN}(h_{t-1}, \phi'_{emb}(y_{t-1})) \tag{2}$$
$$P_t = \text{Softmax}(W_k h'_t) \tag{3}$$
$$\tilde{y}_t = \text{argmax}(P_t) \tag{4}$$

where ϕ'_{emb}, W_k are trainable parameters and ϕ'_{emb} is the embedding matrix of output tokens. $\tilde{y}_t$ is the predicted token at time step t. The initial state of the decoder is initialized by the final state of the encoder: $h'_0 = h_n$.

B. HRED

Different from Seq2seq, HRED treats a dialog as a hierarchical structure and encodes each sentence separately. A sentence x_i is denoted as $x_i = \{w_{i1}, w_{i2}, ..., w_{iL}\}$, L is the length of x_i. A word w_{ij} is encoded by RNN just like in Equation (1):

$$h_{ij} = \text{RNN}(\phi_{emb}(w_{ij}), h_{ij-1}) \tag{5}$$

Then each sentence x_i is represented by the final state h_{iL} of the RNN since it accumulates the information in the whole sentence. Then another RNN called context RNN, encodes the dialog at the sentence level. We write the hidden states of the context RNN as $s_1, s_2, ..s_T$ and the computation of s_i is formulated given h_{iL}:

$$s_i = \text{RNN}(s_{i-1}, h_{iL}) \tag{6}$$

Then the initial state of the decoder is equal to the final state s_T of the context RNN: $h'_0 = s_T$. It is the main difference between HRED and Seq2seq. In the decoding procedure, HRED shares the same process with Seq2seq as in Equation (2), (3), (4).

The HRED is expected to be superior to the vanilla Seq2seq since the context RNN can represent a form of common ground between speakers using a distributed vector, including topics and concepts [14].

IV. OUR SYSTEM

Considering that entities in system responses are determined by the dialog histories, KB results and the decoder, HAMI contains three modules: HADE for dialog history representation, a knowledge matching module for KB interaction and a decoder to generate system responses. The whole system is end-to-end trainable. The framework of our system is illustrated in Figure 1.

A. HADE

To enrich the dialog representation, we propose HADE in light of hierarchical attention networks (HAN) for document

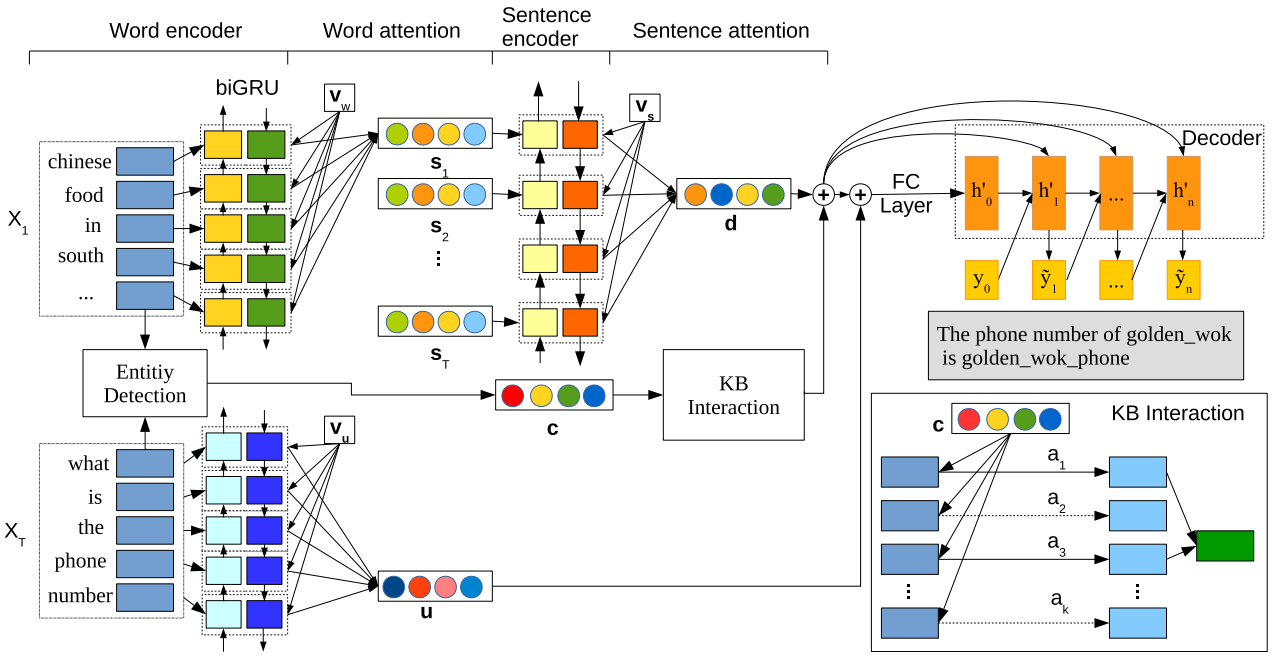


FIGURE 1. The architecture of HAMI. biGRU is an abbreviation for the bidirectional GRU mentioned in Section IV-A. FC Layer is short for fully-connected layer.

classification [30]. Our model strengthens current user utterance by encoding it twice. The whole encoder is composed of two levels of processing: the word level and the sentence level.

1) WORD ENCODER AND WORD ATTENTION

Given a dialog history $X = \{x_1, x_2, ..., x_T\}$ without KB items, a word encoder first process each sentence $x_i, i \in [0, T]$ and produce a representation for each word $w_{ij}, j \in [0, L]$. We first map the words to vectors through a trainable word embedding matrix ϕ_{emb}. Then we use a bidirectional GRU [29] to obtain positional information and incorporate dialog context from both directions. The hidden states of a bidirectional GRU is formulated as follows:

$$\overrightarrow{h}_{ij} = \overrightarrow{\text{GRU}}(\phi_{emb}(w_{ij}), \overrightarrow{h}_{ij-1}) \tag{7}$$

$$\overleftarrow{h}_{ij} = \overleftarrow{\text{GRU}}(\phi_{emb}(w_{ij}), \overleftarrow{h}_{ij+1}) \tag{8}$$

We obtain a representation of word w_{ij} by concatenating the forward and backward hidden states, i.e., $h_{ij} = [\overrightarrow{h}_{ij}, \overleftarrow{h}_{ij}]$. Since not all the words contribute equally to the meaning of the sentence, we introduce attention mechanism to focus on important words and aggregate the representation of those informative words to form a sentence vector s_i. Specifically,

$$u_{ij} = \tanh(W_w h_{ij} + b_w) \tag{9}$$

$$\alpha_{ij} = \text{Softmax}(u_{ij}^{\top} v_w) \tag{10}$$

$$s_i = \sum_{t=1}^{L} \alpha_{ij} h_{ij} \tag{11}$$

where W_w, v_w, b_w are trainable parameters that jointly learned during the training process. That is, we first feed the word representation h_{ij} through a fully-connected layer to get u_{ij} as hidden representation. Then we measure the importance of the word by the similarity between u_{ij} and a word level context vector v_w. A normalized weight α_{ij} is computed through a softmax function and the sentence representation s_i is the weighted sum of the word representations. The context vector v_w can be seen as a query "what is the important word" over the words [30].

Considering that the last utterance x_T, namely the current user input, is of particular importance, we re-process it separately. We use another bidirectional GRU and attention parameters to encode x_T and get an additional representation **u** for it.

2) SENTENCE ENCODER AND SENTENCE ATTENTION

Given the sentence vectors $\{s_1, s_2, ..., s_T\}$, we can get a dialog vector in a similar way. We use a bidirectional GRU to encode sentence s_i:

$$\overrightarrow{h}_i = \overrightarrow{\text{GRU}}(s_i, \overrightarrow{h}_{i-1}) \tag{12}$$

$$\overleftarrow{h}_i = \overleftarrow{\text{GRU}}(s_i, \overleftarrow{h}_{i+1}) \tag{13}$$

We concatenate $\overrightarrow{h}_i, \overleftarrow{h}_i$ to get an annotation of sentence i, i.e., $h_i = [\overrightarrow{h}_i, \overleftarrow{h}_i]$. h_i summarizes the neighbor sentences and centers at sentence i.

To reward sentences that are clues for the subsequent decoding procedure, we again use attention mechanism and introduce a sentence level context vector v_s. This vector is used to measure the importance of the sentences. This can be formulated as:

$$u_i = \tanh(W_s h_i + b_s) \tag{14}$$

$$\alpha_i = \text{Softmax}(u_i^\top v_s) \tag{15}$$

$$\mathbf{d} = \sum_{i=1}^{T} \alpha_i h_i \tag{16}$$

where $\mathbf{d}$ is the dialog vector that summarizes all the information of sentences. Similarly, W_s, v_s, b_s are randomly initialized and learned during the training process.

B. KNOWLEDGE MATCHING

Besides dialog history, KB results are also important contents that should be contained in some system responses. Therefore we design a knowledge matching module for this situation. The whole process is similar to human-performed KB searching. First, we filter out the searching constraints in the dialog history, given a pre-defined entity list $E = \{e_1, e_2, ..., e_M\}$, M is the number of entities. Then we use the representation of the latest entities in the dialog to search for results in a KB $G = \{(sub_1, rel_1, obj_1), ...(sub_K, rel_K, obj_K)\}$, K is the number of items. Each item is a triple of (subject, relation, object).

1) ENTITY DETECTION

The entity detection task aims to find the last k entities mentioned in a dialog history. It can be formulated as a ranking problem based on semantic matching given a pre-defined entity list E. If a word is an entity, it must have high cosine similarity (close to 1.0) between itself and one of the entities in E since they are identical. Otherwise, the similarity score is low. Based on this principle, we design this module, and the details are as follows.

For each word w_i, $i \in n$ in dialog history X, we compute the similarity between it and each entity e_j in E by computing the cosine distance between the word embeddings of w_i and e_j. Then we select the most similar one as w_i's candidate. The scores are filtered by a threshold to ensure the confidence of the results. The last k candidates among the whole dialog history are selected, and their scores are fed to the softmax function for normalization. The normalized scores are used as weights, to sum up the word embeddings of their corresponding candidates:

$$m_{ij} = \cos(\phi_{emb}(w_i), \phi_{emb}(e_j)) \tag{17}$$

$$m_i = \text{Max}(m_{i1}, ..., m_{iM}) \tag{18}$$

$$\mathbf{m} = \text{f}(\{m_i, ..., m_n\}, tr_1) \tag{19}$$

$$\mathbf{m^k} = \text{Last}(\mathbf{m}, k) \tag{20}$$

$$\alpha = \text{Softmax}(\mathbf{m}) \tag{21}$$

$$\mathbf{c} = \sum_{i=1}^{k} \alpha_i \phi_{emb}(w_{\alpha_i}) \tag{22}$$

where w_{α_i} is the corresponding word that with score α_i. Note that $\text{f}(x, tr)$ is a threshold function that $x = x$, if $x > tr$, else $x = 0$. $\mathbf{m}$ is the filtered results after the threshold function, which has fewer elements than the original sequence $m_i, ..., m_n$. $\text{Last}(\cdot, k)$ is a function that chooses the last k values of the input sequence. Generally, k is set to the number of informable slots for a task. This allows HAMI to capture the latest constraints even if the user changes his mind during the dialog. $\mathbf{c}$ is the aggregated representation of detected entities in the dialog history. It will be used as constraint to search KB later.

2) KB INTERACTION

Given the constraint vector $\mathbf{c}$, HAMI matches the related items in KB likewise. For item $(sub_i, rel_i, obj_i), i \in K$, we first sum the word embeddings of obj_i and rel_i as key representation key_i. Then we compute the cosine distance as the similarity between $\mathbf{c}$ and key_i. The main idea is that if key_i is selected, the corresponding sub_i will be returned. The word embeddings of the top k matched subjects are weighted summed using their normalized corresponding similarity scores:

$$m'_i = \cos(\phi_{emb}(obj_i) + \phi_{emb}(rel_i), c) \tag{23}$$

$$\mathbf{m}' = \text{Top}(\{m'_i, ..., m'_K\}, k) \tag{24}$$

$$\mathbf{m}'' = \text{f}(\mathbf{m}', tr_2) \tag{25}$$

$$\alpha' = \text{Softmax}(\mathbf{m}'') \tag{26}$$

$$\mathbf{r} = \sum_{i=1}^{k} \alpha'_i \phi_{emb}(sub_{\alpha'_i}) \tag{27}$$

where $sub_{\alpha'_i}$ is the subject of the corresponding item with score α'_i. $\text{Top}(\cdot, k)$ is a function that chooses the top k values of the input sequence. The summarized representation $\mathbf{r}$ of KB results is computed.

C. RESPONSE DECODER

Given the dialog history representation $\mathbf{d}$, current user utterance vector $\mathbf{u}$ and the KB result $\mathbf{r}$, a concrete response Y' is predicted token by token in the context of $\mathbf{d}$, $\mathbf{r}$ as follows:

$$h'_t = \text{GRU}(h'_{t-1}, \phi'_{emb}(y_{t-1}), [\mathbf{d}, \mathbf{r}]) \tag{28}$$

$$Q_t = \text{Softmax}(W_o h'_t + b) \tag{29}$$

$$\tilde{y_t} = \text{argmax}(Q_t) \tag{30}$$

where h'_t is the hidden state of GRU, y'_t is the predicted token given Q_t, the probability of tokens over the output vocabulary when predicting the t^{th} token. W_o, b are trainable parameters. [,] denotes the concatenation of vectors. Here, ϕ'_{emb} is the embedding function of the decoder over the output vocabulary. Unlike in Seq2seq and HRED, the initial state h'_0 is computed using $\mathbf{d}$, $\mathbf{u}$ and $\mathbf{r}$. It is the semantic aggregation of dialog history, user request and KB results:

$$h'_0 = \text{ReLU}(W_h[\mathbf{d}, \mathbf{u}, \mathbf{r}]) + b_h \tag{31}$$

where W_h, b_h are trainable parameters. ReLU is the rectified activation function [31]. As we can see, the content to generate is determined before the decoding procedure by setting up the initial state h'_0. It serves as an information constraint that decides what to say in the decoding process. It is similar to the procedure of human speaking where the topic and information are usually planned based on dialog history, current user input and KB results (if necessary).

D. ENTITY REGULARIZATION

Generally, generation models learn to minimize the cross-entropy loss between the predicted and the ground truth probability distributions over the output vocabulary. However, as in human communication, informative words are more important and deserve more attention. Here, we design a penalty term L_{ent} for entities in responses. We first compute a representation y'_{att} of the generated response Y' as in HADE's word attention above. Then we generate a bag-of-words approximation b_e of y'_{att} using a fully-connected layer:

$$y'_{att} = \text{Attention}(\phi'_{emb}(Y'), v_y) \tag{32}$$

$$b_e = \text{sigmoid}(W_e y'_{att}) \tag{33}$$

where v_y, W_e are trainable parameters and v_y is the context vector for attention on generated responses. In this way, we expect the model to pay attention to the entities in the predicted response and to learn to decode the aggregated representation y'_{att} as a bag-of-words vector using the fully-connected layer.

We denote y_e as the sequence of entities in the ground truth response and build a one-hot vector of y_e using bag-of-words representation. The penalty term is formulated as:

$$\mathcal{L}_{ent} = |b_e - \text{BOW}(y_e)| \tag{34}$$

where BOW$(\cdot)$ is a function that outputs the bag-of-words vector for an input sequence. By minimizing the discrepancy between the predicted and ground truth bag-of-words vectors, we force the model to learn the generation of entities.

E. OPTIMIZATION

Finally, the sequential cross-entropy loss $\mathcal{L}_r$ between the predicted and the ground truth response is computed. Then we combine $\mathcal{L}_r$ and the entity penalty term $\mathcal{L}_{ent}$. The final loss function $\mathcal{L}$ of HAMI is written as:

$$\mathcal{L}_r = -\frac{1}{T'}\sum_{t=1}^{T'}\sum_{i=1}^{N} p_r(i_t)\log q_r(i_t) \tag{35}$$

$$\mathcal{L} = \mathcal{L}_{ent} + \mathcal{L}_r \tag{36}$$

where T' is the length of generated response, N is the size of the output vocabulary. $p_r(i_t)$ is the probability of the i^{th} token in the vocabulary at t^{th} step in the final response while $q_r(i_t)$ is predicted probability.

To illustrate how HAMI works, we give an example of the whole procedure here. Given a dialog history X listed in 2, HADE first computes the dialog vector $\mathbf{d}$ and user vector $\mathbf{u}$ to represent the context from x_1 to x_5 and user utterance x_5 respectively. Then the similarities between each word and the pre-defined entities are computed, and we obtain the matched entities "chinese", "south" and "expensive" in the entity detection part. The aggregation of their embeddings is computed as $\mathbf{c}$. It is used to interact with a KB based on semantic similarity subsequently. Then items "peking_restaurant r_cuisine chinese", "peking_restaurant r_location south" and "peking_restaurant r_price expensive" are top-ranked by the model. The subjects "peking_restaurant" of the items will be returned, and their embeddings are weighted summed to be $\mathbf{r}$. Given $\mathbf{d}$, $\mathbf{u}$ and $\mathbf{r}$, the decoder generates the response "peking_restaurant is a great restaurant serving expensive chinese food in the south of town" token by token.

TABLE 2. An example dialog history for illustration.

X	Role	Sentence
x_1	User	chinese food in the south part of town
x_2	System	what pricerange would you like
x_3	User	expensive
x_4	System	api_call chinese south expensive
x_5	User	<silence>

V. EXPERIMENTS

In this section, we present the experimental results of our system on two benchmark dialog generation tasks and comparisons them to existing approaches.

A. EXPERIMENTAL SETUP

1) DATASET

We use two public multi-turn task-oriented dialog datasets to evaluate our model: DSTC2 [32] and bAbI dialog task5 [23]. These two datasets are the largest and most favorite benchmark task-oriented dialog datasets used by previous studies [2], [17], [20], [23]–[25]. We use the refined version [23] of DSTC2. It contains human-computer dialog transcripts data without dialog state annotations. The database calls in DSTC are transformed into "api_call" and inserted into the dialog transcripts [2]. We directly use this dataset for our experiments. Some entities are normalized for consistency, as listed in Appendix A. Example dialogs are provided in Appendix B. The dataset contains a training set, a development set, and a held-out test set. It is a human-machine dialog dataset from a real restaurant reservation system, which provides the expression diversity in the real world.

The bAbI dialog dataset consists of five end-to-end dialog learning subtasks in the restaurant domain, which are simulated data. Task 1 to 4 are dialog segments about API calls, refining API calls, recommending options, and providing additional information respectively. Task 5 focuses on the whole procedure. It contains complete dialogs and is the most laborious task of the five. There are two test sets for each task: an ordinary one and the other one that has out-of-vocabulary (OOV) entities unseen in the training set. We only conduct experiments on the ordinary test set because our system is not targeted on the OOV problem. The statistics of the data are given in Table 3.

2) TRAINING DETAILS

The dimension of word embeddings, which is equivalent to the size of attention vectors, is selected to be 300. Numbers of hidden units in HADE are set to 150. The size of hidden units in the decoder is 300 as in previous work based on Seq2seq [17]. The learning rate is controlled by

TABLE 3. Statistics of DSTC2 and bAbi dialog task 5 data.

Dataset	DSTC2	bAbI Task 5
Avg. # of Utterance Per Dialogue	16	31.3
Input Vocabulary Size (EOS included)	1202	2168
Output Vocabulary Size (EOS included)	604	2000
# of Slots	7	10
# of Entities	491	3642
# of Training Dialogues	1618	1000
# of Training Turns	14404	18340
# of Validation Dialogues	500	1000
# of Validation Turns	4159	18457
# of Test Dialogues	1117	1000
# of Test Turns	11237	18398
Max. # of Tokens Per Sentence	23	17
Max. # of Tokens Per Response	23	8

Adam [33] optimizer with the initial learning rate set to 0.001. The learning rate is halved, when the performance stops improving. We apply gradient clipping with a clip-value of 10. Dropout is applied to all the attention outputs with the dropout rate set between [0.1,0.3]. We train models for 50 epochs and select the best one on the development set based on turn accuracy. Early stop is applied when the performance keeps declining for five epochs. For KB items, we reverse the objects and subjects when the relation is one of ''r_address'',''r_phone'',''r_post_code''. It is for the cases that users ask for the objects given the restaurant name. Thresholds for entity detection and KB interaction are set to 0.8 and 0.1 respectively.

3) METRICS

a: TURN/DIALOG ACCURACY

Following the previous work [2], [17], [23]–[25], we report average turn accuracy and dialog accuracy. A generated response is considered to be correct only if all the predicted tokens are identical to the ground truth. Likewise, a dialog is correct only if every generated response is correct. Note that retrieval-based methods only select system responses from candidates [23]–[25], and template-based methods choose a template and fill in slot values. Our model generates tokens one by one, which is more challenging.

b: BLEU

We also use BLEU [34], to measure the correlation between generated and the ground truth responses [35]. It is widely used for MT [34], dialog systems [8], [17] and chat-bots [13].

c: ENTITY F1

Additionally, we report micro entity F1 that measures the correctness of information in generated responses. In this metric, the ground truth entities of each response are extracted according to the ontology set and the underlying KB via simple string matching. This metric evaluates the ability to generate relevant entities from the provided dialog history and KBs [17], [26]. Since the datasets are end-to-end dialogs without dialog state tracking labels, we report entity F1 rather than the joint accuracy of dialog states as in [8], [9], and [16].

TABLE 4. Performance of models on DSTC2. Results are in percentage. Dashes indicate unavailable values. Results with $*$ are not directly comparable. The results of the best performer and our proposed HAMI are in bold.

Models	Ent.F1	Turn Acc	Dial. Acc	BLEU
Rules	-	33.3	0.0	-
MemNN	-	41.1	0.0	-
GMemNN	-	**47.4**	1.4	-
QRN	-	43.8	-	-
HCN*	-	55.6	1.3	-
Transformer	69.4	38.3	0.0	52.2
Seq2seq	69.7	46.0	**1.5**	55.0
+Attn	67.1	46.0	1.4	**56.6**
+Copy	71.6	47.3	1.3	55.4
Mem2seq	**75.3**	45.0	0.5	55.3
HRED	74.2	41.3	1.2	54.1
HAMI	**79.5**	**45.9**	**1.7**	**54.9**

B. EXPERIMENTAL RESULTS

1) DSTC2

We compare proposed HAMI with state-of-the-art end-to-end approaches including MemNN [22], [23], GMemNN [24], QRN [25], HCN [2], Seq2seq+Copy [17], Mem2seq [20]. We also implement two baseline models: HRED [14] and Transformer [19]. We are very interested in the Transformer's performance in this task since it achieves state-of-the-art results in MT with less training time. Among all the approaches above, the first four are retrieval-based models that choose an answer from candidates. We emphasize that HCN is a combination of hand-coded templates and learned models while the rest are pure data-driven. Note that the results for QRN are the refined results reported by [20]. We run our model 5 times with different seeds and report the average results. Results are shown in Table 4.

a: OUR MODEL ACHIEVES THE HIGHEST ENTITY F1 AND DIALOG ACCURACY

As shown in Table 4, our model achieves the highest 79.5% in entity F1 score and 1.7% for dialog accuracy. The high entity F1 score indicates that HAMI performs well in predicting entities by jointly using HADE, knowledge matching and entity regularization. Compared with Mem2seq which features entity retrieving and previous state-of-the-art results, our model surpasses it by 4.2% in entity F1 score. It is plausible that Mem2seq only considers KB interaction while HAMI models dialog representation and uses entity regularization in addition. Namely, HAMI can predict entities that from both dialog histories and KB results but Mem2seq can only carry the entities from KB results. Our model also creates a record of 1.7% in dialog accuracy, winning the previous best result of Seq2seq by 0.2%. Though HAMI and Seq2seq have similar turn accuracy and BLEU, there is an essential difference between them. Most of the correct samples produced by Seq2seq are sentences that do not contain entities like greeting and asking if need any help. Compared to Seqseq, HAMI generates more correct replies in providing results including restaurants, addresses and postcodes.

TABLE 5. Performance of models on bAbi dialog task 5 in percentage. Dashes indicate unavailable values. Results with * are not directly comparable. Best records and results of HAMI are in bold.

Models	Turn Acc	Dial. Acc
Rules	100	100
MemNN	96.1	49.4
GMemNN	96.3	52.5
QRN	**99.6**	-
HCN*	100	100
Transformer	95.5	59.8
Seq2seq*	98.8	81.5
+Attn*	98.4	**87.3**
+Copy	-	-
Mem2seq	98.2	72.9
HRED	97.4	62.7
HAMI	**99.2**	**88.3**

Therefore, HAMI is capable of finishing more complicated dialogs and capturing more information than existing models, which results in higher dialog accuracy.

b: OTHER FINDINGS

There are other interesting results in Table 4. As we can see, Transformer performs even worse than Seq2seq. It may not be suitable for this task since it seems to converge to a local optimum and learns some unique features in the training set rather than universal characteristics for the task. It achieves high turn accuracy (90% or so) on training set quickly but reports low performance on the development set. The gap between these two results is more substantial than other models. It may be attributed to its complicated multiple levels of attention operations. Another finding is that HRED achieves higher entity F1 score than Seq2seq+Copy. That means HRED can improve entity F1 of Seq2seq by hierarchical dialog history modeling. Compared with Seq2seq+Copy that concatenate a dialog into a long sentence, HRED reduces time-consuming since it encodes multiple sentences in parallel.

2) bAbi DIALOG

We also report results on bAbI dialog task 5. We follow Bordes and Weston [23] to compare the performance based on average turn accuracy and dialog accuracy. This task is easier than DSTC2 since it is a simulated dataset that can be traversed by handcrafted rules. Note that the results reported by HCN are not comparable to ours as HCN casts the task into a template selection task and incorporates handcrafted rules. Results are listed in Table 5.

a: OUR MODEL ACHIEVES COMPETITIVE TURN ACCURACY AND THE HIGHEST DIALOG ACCURACY

Among all the generation models, our system achieves the best performance even though results of Seq2seq and Seq2seq+Attn are augmented with multiple features.[2] It shows the effectiveness of HADE for dialog modeling and knowledge matching in HAMI. Our system ranks second in turn accuracy, only losing 0.4% compared with the best result of QRN. Our system outperforms all the retrieval-based and generation-based methods in dialog accuracy. It shows the superiority of HAMI in conducting successful dialogs over all the existing template-free models.

[2]The results of these two models are from [20], where roles and turn numbers are used as features. For example, for each word w_i in an user utterance, it is denoted as ($user, turn1, w_i).

TABLE 6. Performance of models on DSTC2. Results are in percentage. EntReg is short for entity regularization. KM indicates the knowledge matching module.

Models	Ent.F1	Turn Acc	Dial. Acc	BLEU
HADE+EntReg+KM	**79.5**	**45.9**	**1.7**	54.9
HADE+EntReg	78.3	44.3	1.5	54.2
HADE	76.7	44.9	1.3	**55.1**
HRED	74.2	41.3	1.2	54.1

b: GENERATIVE METHODS CAN ACHIEVE GOOD PERFORMANCE

As can be seen, though retrieval-based methods (MemNN, GMemNN, and QRN) treat the dialog learning task as a selection task that is easier than a generation task, generation models can still overpass retrieval-based methods. Besides, one can find that Seq2Seq and Seq2seq+Attn are also strong baselines using role and turn features, which further supports that generation methods can also achieve good performance in task-oriented dialog systems. Compared with Seq2seq, HRED model does not perform well in this dataset. It is plausible that though HRED can capture informative words in dialog histories, it fails to search the KB for correct KB results. Therefore, it fails to improve turn accuracy that requires each token to be correct. Transformer performs the worst as it suffers from converging to local optima.

VI. DISCUSSION

A. ABLATION STUDY

In this part, we conduct experiments on DSTC2 to investigate the influence of HADE, knowledge matching and entity regularization in our system by removing them separately.

By removing the KB interaction module, we can see the results in all metrics decrease. It indicates that by providing KB results, KB interaction module benefits the dialog generation procedure. It also reveals that some sentences generated by HADE+REG are almost correct, only missing the entities from KB results. After filling in the information from KB results, these sentences become complete and correct. Comparing HADE+EntReg to HADE, one can observe an increase in entity F1 and dialog accuracy but a decrease in turn accuracy and BLEU. The 1.6% improvement in entity F1 shows the effectiveness of the entity regularization. But it probably deteriorate the fluency of generated sentences, resulting in turn accuracy and BLEU dropping. The results of HADE and HRED shows that by using attention mechanism and adding the encoded current user utterance, HADE improves the performance in all aspects. It also reveals the importance of the informative words in dialog histories.

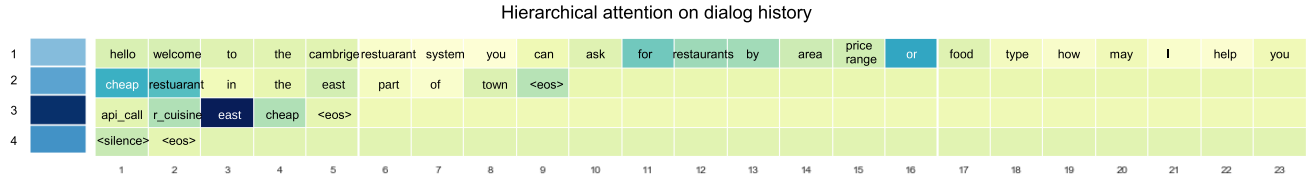


FIGURE 2. The hierarchical attention visualization in the restaurant domain. The blue bars on the left are attention scores of the sentences. Dark color indicates high energy. The turn indexes of sentences are on the left.

TABLE 7. The gain from our system over Mem2seq based on pairwise human judgments.

	HAMI-win	HAMI-lose	Tie
Ratio(%)	61.5	7.5	31.0

B. HUMAN EVALUATION

In this part, we report human evaluation regarding 200 random sampled generated responses. Results of our system and the state-of-the-art model Mem2seq are compared. Different from the protocols that ask experts to score the results on a scale from 1 to 5 [16], we follow Li *et al*. [36], [37] to invite ten judges to choose the right one between two given answers. Ties are permitted. If both answers are correct or wrong, they are marked. Each given sample is presented with its multi-turn dialog history and KB items if any. Each dialog consists of up to 6 turns. Note that we consider an answer to be correct only if it is accurate in both grammar and entities (even KB results). Results are presented in Table 7. We observe a significant quality improvement from our model. It supports that our system generates more informative and effective responses. Among the tied samples that are summed up to 31%, 15% are responses that are correctly generated by both systems and the other 16% are wrong. That is, HAMI achieves 76.5% accuracy actually. The 76.5% turn accuracy is higher than the 44.5% turn accuracy in DSTC2, which implies that people are satisfied with most of HAMI results and the golden answers are not unique in reality.

C. VISUALIZATION

To display what HADE captures from a dialog history, we plot the attention weights of each word and each sentence with different colors in Figure 2.

We first discuss the word level attention in HADE. Among the weights of words, we find that entities earn more attention than other words. For example, ''cheap'' and ''east'' are always in darker blue than the surrounding words. Second, functional words are attached more emphasis. In the first guiding sentence generated by the agent system, we can see the greeting words are basically ignored while the keywords that indicate the system function are focused. i.e.,''for restaurants by'' and ''or'' are paid more attention than ''hello''. In addition, the same word is given different scores in sundry contexts. Take ''restaurant'' as an example. It gains higher weight following ''cheap'' in the second sentence than following ''Cambridge'' in the first sentence. Based on these findings, we believe that HADE can capture entities and informative words and is context sensitive.

As for sentence level attention, the third utterance that contains all the entities and the system action draws the most attention. It indicates that HADE results are the aggregations of important contents in dialog histories. Therefore, it can produce better and more meaningful representation than Seq2seq and other methods that encode dialog histories in a plain way.

TABLE 8. The matching scores between the triples in the given KB and the constraint vector, a weighted sum of the word embeddings of (''cheap'',''east'',''east''). The top 3 scores are in bold.

Subjects	Relations	Objects	Scores
the_missing_sock_post_code	r_post_code	the_missing_sock	**0.124**
the_missing_sock	r_cuisine	international	0.029
the_missing_sock	r_location	east	**0.584**
the_missing_sock_phone	r_phone	the_missing_sock	0.043
the_missing_sock_address	r_address	the_missing_sock	0.119
the_missing_sock	r_price	cheap	**0.322**

To illustrate how the knowledge matching module in HAMI works, we visualize the matching scores of each word in an example of dialog history. It is displayed in Figure 3. We can see the entities are all detected and given high similarity scores. As in our algorithm, we choose the last k entities as the constraint to interact with KB. In DSTC2, k is set to 3. So we get ''cheap'',''east'',''east'' and their word embeddings are weighted summed as $\mathbf{c}$. In this way, HAMI uses the latest constraint to search KBs even users change their minds.

Next, $\mathbf{c}$ is used to interact with KB. Then cosine distances between the triples in the given part of KB and the constraint representation is listed in Table 8. As expected, the KB item with the ''east'' object obtains the highest score. The KB item with the ''cheap'' object follows. It is because a key presentation is a semantic fusion of the corresponding relation and object. A relative object or relation results in a high similarity between the key representation and the constraint vector. Then the corresponding subjects of the top ranking items will be returned and their embeddings will be weighted summed according to their matching scores. If multiple top entries point to the same subject, then the embeddings of the subject will be added multiple times. The resulted vector will be semantically close to the subject, which is exactly what we want. Therefore, the knowledge matching module is capable of searching KBs and representing reasonable results.

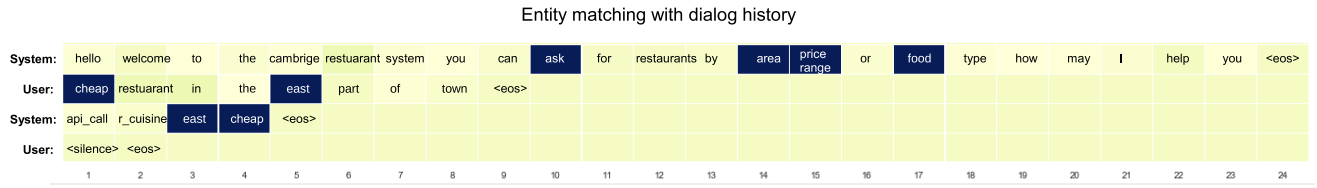


FIGURE 3. The entity detection visualization in the restaurant domain. Dark color indicates high energy.

TABLE 9. The statistics of the wrong samples generated by HAMI.

Categories	Themes	Ratio(%)	Sum(%)
Tracking Entities	provide restaurants	22.5	30.8
	api_call	8.3	
Searching KB	negative result	11.5	14.8
	provide addresses	2.0	
	provide phone numbers	1.0	
	provide postcodes	0.3	
Dialog Policy	require more	19.5	54.1
	confirm	12.6	
	repeat	8.8	
	ask for food	4.3	
	ask for area	3.8	
	ask for price range	2.4	
	help	2.7	
	greeting	0.2	

D. ADVANTAGE AND LIMITATION

We sample some generated responses from Mem2seq and our system to see their ability to predict entities. Samples are given in Figure 4. In sample 1, our system accurately captures the updated entities and searching conditions while the Mem2seq focuses on the out-dated ones. In sample 3, HAMI is able to offer results by matching entities from the dialog history and extracting KB results. As for Mem2seq, it asks the user for cuisine information again, which shows its limitation in context modeling. Sample 4 shows HAMI's capability of providing correct information to the user. Though all the answers generated by the Mem2seq are grammatically correct, they fail to provide expected information and the entities mentioned in the previous dialog. As illustrated, our model generates better replies with correct entities.

Though our system improves the information correctness effectively, it fails to report a high turn accuracy in DSTC2. To dig out the reasons, we counted the failure samples generated by HAMI for each category.[3] The statistics are listed in Table 9.

The distribution over different categories tells that most wrong responses are involved in dialog policy.[4] Among the themes in dialog policy, require more[5] counts for the largest proportion. These sentences are used to ask for additional constraints, sharing the same function as "ask for food/area/price range". They are designed to improve the user experience. However, HAMI always directly ask for this information. Confirm and repeat are also used to make responses more human-like or when there is automatic speech recognition failure. HAMI does not learn these dialog policies and fails at all these three themes.

	Dialog History	Generated Responses
1	System: i m sorry but there is no restaurant serving swiss food User: ***vietnamese*** System: could you please repeat that User: how about ***vietnamese*** food	Mem2seq: You are looking for is ***swiss*** restaurant right? HAMI: ***thanh_binh*** serves ***vietnamese*** food
KB	None	
2	User: looking for a restaurant in the ***south*** part of town System: there are restaurants in the south of town what type of food do you want? User: ***eritrean*** food System: you are looking for a restaurant serving ***any*** kind of food right? User: yes	Mem2seq: api_call ***eritrean south expensive*** HAMI: ***api_call r_cuisine south expensive***
KB	None	
3	User: i need a restaurant in the ***north*** part of town System: there are restaurants in the north of town what type of food do you want User: i ***dontcare*** System: There are restaurants in the north of town serving any kind of food. what pricerange would you like? User:***moderate*** System:api_call r_cuisine ***north moderate*** User:<silence>	Mem2seq: What kind of food would you like? HAMI: ***meghna*** is a nice place in the ***north*** of town and the prices are ***moderate***
KB	meghna R_post_code meghna_post_code meghna R_cuisine indian meghna R_location north meghna R_phone meghna_phone meghna R_address meghna_address meghna R_price moderate the_nirala R_post_code the_nirala_post_code the_nirala R_cuisine indian the_nirala R_location north the_nirala R_phone the_nirala_phone the_nirala R_address the_nirala_address the_nirala R_price moderate golden_wok R_post_code golden_wok_post_code golden_wok R_cuisine chinese golden_wok R_location north golden_wok R_phone golden_wok_phone golden_wok R_address golden_wok_address golden_wok R_price moderate	
4	System: could you please repeat that User: ***vietnamese*** food in any pricerange System: api_call vietnamese r_location r_price User: silence System: ***thanh_binh*** serves ***vietnamese*** food User: uh ***address***	Mem2seq: api_call ***vietnamese*** r_location ***expensive*** HAMI: sure, ***thanh_binh*** is on ***thanh_binh_addres***
KB	venue R_cuisine vietnamese venue R_address venue_address venue R_phone venue_phone venue R_post_code venue_post_code venue R_location west thanh_binh R_post_code thanh_binh_post_code thanh_binh R_cuisine vietnamese thanh_binh R_location west thanh_binh R_phone thanh_binh_phone thanh_binh R_address thanh_binh_address thanh_binh R_price cheap	

FIGURE 4. Examples of generated responses of Mem2seq and our system. `r_cuisine`, `r_location` and `r_price` are used for no special demand in cuisine, location and price respectively when issuing a call for database search.

[3]These categories are defined by us according to the content of the system replies.

[4]The dialog policy determines what to do with user input, namely controlling system actions including greeting, repeating, confirming, searching KB and farewell.

[5]It indicates the sentences that start with "There are restaurants with" and end with "what location/price/food do you like?".

One can also find that the ratio of providing restaurants is also high. We look up dialog histories of the samples and find two main reasons: first, the golden answers are always given the first restaurant if there are multiple suitable results; second, there are several expressions of providing results in the dataset. We find that most of the answers provided by HAMI are acceptable but not identical to the ground truth answers. As for "api_call", sometimes HAMI generates meaningless api_calls like "api_call r_cuisine r_location r_price" that without any constraint. Sometimes there are grammar mistakes or wrong entities in generated api_calls. After checking the frequency of these entities in the training set, we find that they are of low occurrence. These words are rarely learned by the decoder thus hard to predict due to their low values over the probability distribution of the whole vocabulary. Grammar errors may be attributed to the entity penalty that reduces the importance of fluency and language quality.

When it comes to negative results which state matching failure in the form of " I am sorry but there is no...", it always appears without any api_call. HAMI fails to predict all this kind of results since it predicts api_calls instead. In fact, it is unreasonable to provide a negative result without searching a KB. It may be a problem of the dataset. One can find that the ratios of providing addresses, phone numbers and postcodes are low. It shows the advantage of HAMI.

VII. CONCLUSION

In this paper, we have presented novel end-to-end neural networks to solve the vital information deficiency problem in task-oriented dialog generation without handcrafted rules or well-designed features. Our model features a hierarchical attention encoder for dialogs, a knowledge matching module that searches the KB based on semantic similarity, and an entity penalty term that emphasizes the entities in system responses. Experimental results on public benchmark datasets show that our model improves the correctness of informational contents in generated responses. Our model also outperforms state-of-the-art result by 4.2% in entity F1. The ablation study shows that all these three modules in our proposed model are effective in improving information correctness.

Besides task-oriented dialog generation, HADE and the idea of emphasizing keywords by a penalty term can also be applied to open domain conversations, with the dialog history information and keywords regularization instructing the response generation progress.

This paper also shows the potential of Seq2seq framework for task-oriented dialog systems. We hope this project may support a better understanding of Seq2seq for task-oriented dialog generation.

APPENDIX A
ENTITY NORMALIZATION

The original entities in the table are refined to the form on the right.

TABLE 10. Entity normalization. The strings on the left are transform to strings on the right.

Original	Normalized
addr	address
post code	postcode
dont care	dontcare
good bye	goodbye
center	centre
price range	pricerange

TABLE 11. An example dialog from the refined DSTC2 dataset. User utterances are italic.

1	Hello , welcome to the Cambridge restaurant system . You can ask for restaurants by area , price range or food type . How may I help you ?
2	*cheap restaurant south part of town*
3	`api_call R_cuisine south cheap [database result returned]`
4	the_lucky_star is a nice restaurant in the south of town in the cheap price range
5	*address*
6	Sure , the_lucky_star is on the_lucky_star_address
7	*phone number*
8	The phone number of the_lucky_star is the_lucky _star_phone
9	*thank you goodbye*
10	you are welcome

APPENDIX B
bAbI TASK 6 EXAMPLE DIALOG

An example dialog in refined DSTC2 dataset is given in Table 11.

ACKNOWLEDGMENT

This work is partially supported by the National Natural Science Foundation of China (Nos. 11590770-4, 61650202, 11722437, U1536117, 61671442, 11674352, 11504406, 61601453), the National Key Research and Development Program (Nos. 2016YFB0801203, 2016YFC0800503, 2017YFB1002803), Foundation of Science and Technology on Information Assurance Laboratory (No.KJ-17-102) and the Young Talents Program of the Institute of Acoustics, Chinese Academy of Sciences (No. QNYC201603).

REFERENCES

[1] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 2009.

[2] J. D. Williams, K. Asadi, and G. Zweig, "Hybrid code networks: Practical and efficient end-to-end dialog control with supervised and reinforcement learning," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*. Vancouver, BC, Canada: Association for Computational Linguistics, vol. 1, 2017, pp. 665–677. [Online]. Available: http://www.aclweb.org/anthology/P17-1062

[3] J. D. Williams, "The best of both worlds: Unifying conventional dialog systems and pomdps," in *Proc. 9th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Brisbane, Hall, Australia, Sep. 2008, pp. 1173–1176. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2008/i08_1173.html

[4] S. Young, M. Gašić, B. Thomson, and J. D. Williams, "POMDP-based statistical spoken dialog systems: A review," *Proc. IEEE*, vol. 101, no. 5, pp. 1160–1179, May 2013.

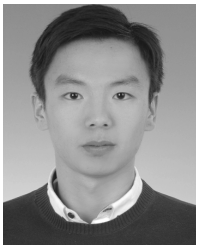
[5] M. S. Henderson, "Discriminative methods for statistical spoken dialogue systems," Ph.D. dissertation, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 2015.

[6] X. Yang *et al.* (2016). "End-to-end joint learning of natural language understanding and dialogue manager." [Online]. Available: https://arxiv.org/abs/1612.00913
[7] T.-H. Wen *et al.*, "Conditional generation and snapshot learning in neural dialogue systems," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Austin, TX, USA: Association for Computational Linguistics, Nov. 2016, pp. 2153–2162. [Online]. Available: https://aclweb.org/anthology/D16-1233
[8] T. Zhao, A. Lu, K. Lee, and M. Eskenazi, "Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability," in *Proc. 18th Annu. SIGdial Meeting Discourse Dialogue*, 2017, pp. 27–36.
[9] W. Lei, X. Jin, M.-Y. Kan, Z. Ren, X. He, and D. Yin, "Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1. Melbourne, VIC, Australia: Association for Computational Linguistics, 2018, pp. 1437–1447. [Online]. Available: http://aclweb.org/anthology/P18-1133
[10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
[11] O. Vinyals and Q. Le. (2015). "A neural conversational model." [Online]. Available: https://arxiv.org/abs/1506.05869
[12] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2015, pp. 1577–1586.
[13] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and B. Dolan. (2016). "A persona-based neural conversation model." [Online]. Available: https://arxiv.org/abs/1603.06155
[14] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proc. AAAI*, vol. 16, 2016, pp. 3776–3784.
[15] I. V. Serban *et al.*, "A hierarchical latent variable encoder-decoder model for generating dialogues," in *Proc. AAAI*, 2017, pp. 3295–3301.
[16] T.-H. Wen *et al.*, "A network-based end-to-end trainable task-oriented dialogue system," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 1. Valencia, Spain: Association for Computational Linguistics, 2017, pp. 438–449. [Online]. Available: http://aclweb.org/anthology/E17-1042
[17] M. Eric and C. Manning, "A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 2, 2017, pp. 468–473.
[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Representations*, 2015.
[19] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[20] A. Madotto, C.-S. Wu, and P. Fung. (2018). "Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems." [Online]. Available: https://arxiv.org/abs/1804.08217
[21] J. Gu, Z. Lu, H. Li, and V. O. K. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 1631–1640.
[22] S. Sukhbaatar J. Weston, and R. Fergus, "End-to-end memory networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2440–2448.
[23] A. Bordes, J.-L. Boureau, and J. Weston. (2016). "Learning end-to-end goal-oriented dialog." [Online]. Available: https://arxiv.org/abs/1605.07683
[24] F. Liu and J. Perez, "Gated end-to-end memory networks," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1–10.
[25] M. Seo, S. Min, A. Farhadi, and H. Hajishirzi. (2016). "Query-reduction networks for question answering." [Online]. Available: https://arxiv.org/abs/1606.04582
[26] M. Eric, L. Krishnan, F. Charette, and C. D. Manning, "Key-value retrieval networks for task-oriented dialogue," in *Proc. 18th Annu. SIGdial Meeting Discourse Dialogue*, 2017, pp. 37–49.
[27] I. V. Serban *et al.*, "Multiresolution recurrent neural networks: An application to dialogue response generation," in *Proc. AAAI*, 2017, pp. 3288–3294.
[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[29] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. (2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling." [Online]. Available: https://arxiv.org/abs/1412.3555
[30] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489.
[31] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, Apr. 2011, pp. 315–323. [Online]. Available: http://proceedings.mlr.press/v15/glorot11a.html
[32] M. Henderson, B. Thomson, and J. D. Williams, "The second dialog state tracking challenge," in *Proc. 15th Annu. Meeting Special Interest Group Discourse Dialogue (SIGDIAL)*, 2014, pp. 263–272.
[33] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980
[34] 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*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 311–318.
[35] A. Ritter, C. Cherry, and W. B. Dolan, "Data-driven response generation in social media," in *Proc. Conf. Empirical Methods Natural Lang. Process.*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 583–593.
[36] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao, "Deep reinforcement learning for dialogue generation," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2016, pp. 1192–1202.
[37] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2017, pp. 2157–2169.

JUNQING HE was born in Zhongshan, Guangdong, China, in 1991. She received the B.S. degree in digital media processing from the Communication University of China, Beijing, China, in 2014. She is currently pursuing the Ph.D. degree in natural language processing with the Institute of Acoustics, Chinese Academy of Sciences and the University of Chinese Academy of Sciences, Beijing.

She used to study cross-lingual semantic representation and conversational text classification. Her current research interest includes end-to-end dialog systems. She has been a member of the Association for Computational Linguistics, since 2017, and a member of the European Language Resource Association, since 2018.

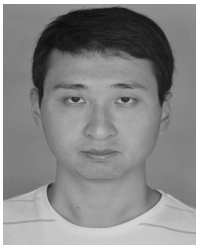
BING WANG received the Ph.D. degree in computer science from The University of New South Wales, Australia, in 2014. She is currently an Associate Professor with the Institute of Acoustics, Chinese Academy of Sciences, China. Her research interests include data mining, machine learning, and social computing.



MINGMING FU was born in Ji'an, Jiangxi, China, in 1993. He received the B.S. degree in electrical and information engineering from the Communication University of China, Beijing, in 2016. He is currently pursuing the degree with the University of Chinese Academy of Sciences. His research direction is natural language processing.

His research interests include spoken language understanding, named entity recognition and text classification.

TIANQI YANG was born in Cangzhou, Hebei, China, in 1993. She received the B.S. degree from the Department of Automation, Tsinghua University, China, Beijing, in 2015, and the M.S. degree in automation from Tsinghua University, China, Beijing, in 2018. She is currently a Research Assistant with the Institute of Acoustics, Chinese Academy of Sciences. Her research direction is natural language processing, including spoken language understanding, natural language generation, and dialog systems.



XUEMIN ZHAO received the B.S. degree in communication engineering from Nankai University, Tianjin, China, in 2007, and the Ph.D. degree in speech signal processing from the Institute of Acoustics, Chinese Academy of Sciences, Beijing, China, in 2012.

From 2012 to 2014, he was a Research Assistant with the Institute of Acoustics, Chinese Academy of Sciences, and became an Associate Researcher, since 2014. His research interests include natural language processing and digital audio watermark techniques, spoken language understanding, named entity recognition and text classification. He is in charge of the development of spoken language understanding engines in intelligent agent projects.