Received xxx; | Revised xx; | Accepted xx

DOI: xxx/xxxx

ORIGINAL ARTICLE

An effective context-focused hierarchical mechanism for task-oriented dialogue response generation†

Meng Zhao[1] | Zejun Jiang[1] | Lifang Wang*[1] | Ronghan Li[1] | Xinyu Lu[1] | Zhongtian Hu[1] | Daqing Chen[2]

[1]School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, PR China
[2]Division of Computer Science and Informatics, School of Engineering, London South Bank University, London SE1 0AA, UK

Correspondence
*Lifang Wang, School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, PR China. Email: wanglf@nwpu.edu.cn

Present Address
This is sample for present address text this is sample for present address text

Abstract

Task-oriented dialogue system (TOD) is one kind of application of artificial intelligence (AI). The response generation module is a key component of TOD for replying to user's questions and concerns in sequential natural words. In the past few years, the works on response generation have attracted increasing research attention and have seen much progress. However, existing works ignore the fact that not each turn of dialogue history contributes to the dialogue response generation and give little consideration to the different weights of utterances in a dialogue history. In this paper, we propose a hierarchical memory network mechanism with two steps to filter out unnecessary information of dialogue history. First, an utterance-level memory network distributes various weights to each utterance (coarse-grained). Second, a token-level memory network assigns higher weights to keywords based on the former's output (fine-grained). Furthermore, the output of the token-level memory network will be employed to query the knowledge base (KB) to capture the dialogue-related information. In the decoding stage, we take a gated-mechanism to generate response word by word from dialogue history, vocabulary, or KB. Experiments show that the proposed model achieves superior results compared with state-of-the-art models on several public datasets. Further analysis demonstrates the effectiveness of the proposed method and the robustness of the model in the case of an incomplete training set.

KEYWORDS:
task-oriented dialogue systems, memory networks, natural language gereration, natural language processing (NLP), deep learning

1 | INTRODUCTION

Developing a dialogue system is a big challenge in the area of Artificial Intelligence (AI), and has recently attracted a lot of attention in both in academia and industry due to the development of deep learning technologies. Task-oriented dialogue system (TOD) is an important paradigm of dialogue system, and is used to accomplish user-specific goals via natural dialogue with limited turns, such as in restaurant reservation and tickets booking. Compared with open-domain dialogue systems, a TOD

†This is an example for title footnote.

[0]**Abbreviations:** ANA, anti-nuclear antibodies; APC, antigen-presenting cells; IRF, interferon regulatory factor

R_name	madrid_expensive_british_6stars	madrid_expensive_british_8stars	madrid_expensive_british_1stars
R_rating	6	8	1
R_price	expensive	expensive	expensive
R_number	six	six	six
R_address	madrid_expensive_6stars_address	madrid_expensive_8stars_address	madrid_expensive_1stars_address
R_location	madrid	madrid	madrid
R_cuision	british	british	british
R_phone	madrid_expensive_6stars_phone	madrid_expensive_8stars_phone	madrid_expensive_1stars_phone

```
$u :  good morning
$s :  hello what can i help you with today
$u :  can you book a table in a cheap price range for two
$s :  i'm on it
$u :  <SILENCE>
$s :  any preference on a type of cuisine
$u :  with british food
$s :  where should it be
$u :  madrid please
$s :  ok let me look into some options for you
$u :  <SILENCE>
$s :  api_call british madrid two cheap
$u :  instead could it be for six people
$s :  sure is there anything else to update
$u :  actually i would prefer in a expensive price range
$s :  sure is there anything else to update
$u :  no
$s :  ok let me look into some options for you
$u :  <SILENCE>
$s :  api_call british madrid six expensive
$u :  <SILENCE>
$s :  what do you think of this option: madrid_expensive_british_8stars
$u :  that looks great
$s :  great let me do the reservation
$u :  can you provide the address
$s :  here it is madrid_expensive_8stars_address
$u :  may i have the phone number of the restaurant
$s :  here it is madrid_expensive_8stars_phone
 ...
```

FIGURE 1 Example dialogue session of TOD. The upper table shows several n-tuples sampled from knowledge base. Lower table shows multi-turn dialogues. In the lower table, the black sentences are spoken by the user, the orange sentences are spoken by the dialogue system. The special token "<SILENCE>" is used to fill in for the missing user utterance.

has a fixed task domain and a specific dialogue goal which determines a TOD should complete a dialogue quickly through a knowledge base (KB) within the specific task domain. A TOD usually consists of four functional components, natural language understanding (NLU), dialogue state tracking (DST), dialogue policy (DP) learning, and natural language generation (NLG). The DST and DP are also combined as dialogue management (DM) module[1]. The entire procedurefor a TOD to complete a dialogue can be described as follows.

The NLU module deals with a user's inputs to classify a user's intent and also is responsible for slot filling task, and then the DST module updates the dialogue state and makes an API call to require the relevant information from KB that mathes the user's goal. Furthermore, the DP module decides which dialogue act (including dialogue act and slot type) to choose for the next turn. Finally, the NLG module maps the system's act to a natural dialogue response. An example of the entire TOD process is given in Fig. 1.

There are two strategies to build a task-oriented dialogue system, namely, pipeline solutions and end-to-end solutions. Traditional pipeline solutions[2,3] build a TOD into four independent modules, and each of them is designed and trained separately. The main drawback of this strategy is that it involves significant manual annotation work, and therefore, is time-consuming and much more expensive. Even worse, any error in an intermediate module could propagate to the following modules, leading to error accumulation in the system. To reduce the human workload on feature extraction, end-to-end dialogue systems have been proposed[4,5,6], which directly take a natural dialogue as input and generate a corresponding dialogue response as output. Works in this area using recurrent neural networks (RNN) have proven promising.

Recently, with the release of multi-domain task-oriented dialogue datasets, such as MultiWOZ 2.0[7] , MultiWOZ 2.1[8], and SGD[9], a new variant of end-to-end framework, called end-to-end pipelined model, has been explored[10,11,12]. In the new paradigm, the overall training process is in an end-to-end manner that incorporates intermediate supervision in order to address new challenges, for instance, cross-domain slot filling and temporal planning. To tackle the error propagation problem, the training process employs multi-task joint learning with separate subtasks like DST and DP. Meanwhile, large-scale pre-trained language models, e.g., GPT[13], GPT2[14] and GPT3,[15] present a wide use in open-domain dialogue systems. Some works further leverage pre-trained language models to explore multi-domain TOD in an end-to-end pipelined way[16,17,18,19]. On the other hand, KB information is a key part of TOD, which determines the quality of responses. Recent research has exploited the memory network (MN)[20,21] to encode KB information and has achieved promising results[22,23,24].

Although the aforementioned studies have proven the power of end-to-end training manners and MN in TOD, they suffer from ineffectively using dialogue history. The main reason for that is they have ignored the fact that not each turn of dialogue history is useful for the dialogue response generation. Different from chitchat scenario, TOD's core task is to complete user's goal step by step. With the dialogue session proceeding, some dialogue history may not be valuable for the following turns. For example, in Fig. 1, when the current turn dialogue updates the request during the restaurant reservation, information of the previous turn's dialogue may be obsolete for the following turn. However, existing approaches[23,25,26] give little consideration to this issue. They tend to treat each utterance equally in the whole dialogue history, which suppresses the useful information to some extent.

Therefore, this paper proposes a hierarchical MN mechanism to focus on critical content in dialogue history. The mechanism is composed of a two-step key point filtering strategy as detailed below.

First, RNN are utilized to encode each utterance in the dialogue history and an utterance-level MN (coarse-grained filter) with the encoded utterance representations is constructed. In addition, an utterance-level attention distribution between a query and utterance-level MN is considered. The utterance-level attention distribution can be seen as the coarse-grained distribution to measure the different weights of each utterance in the dialogue history.

Second, a token-level MN (fine-grained filter) is constructed to represent the dialogue history based on previous studies[23,24]. Then a fine-grained output of the token-level MN can be achieved with the utterance-level attention. The output is used as an updated query to filter the KB memory network. In the decoding stage, a shared MN is used, combining token-level memory network and KB memory network, to generate the response word by word.

The contributions of this presented research are mainly four-fold:

- A novel hierarchical mechanism is proposed, including an utterance-level and a token-level MN, to focus on key content in a dialogue history. The utterance-level memory network is employed to select important utterances, and then the token-level memory network is used to grasp the keywords based on the result of the utterance-level memory network.
- Three separate memories are designed to model the utterance-level and token-level dialogue history and KB entries, respectively. The iterative interactions are performed by employing different granularity semantic representations (utterance-level and token-level dialogue history) and different representations for distinctive format data (dialogue history and KB) to produce coherent and human-like dialogue responses.
- To strengthen the model's ability of dealing with out-of-vocabulary (OOV) problem, an auxiliary task is introduced to provide an additional loss to spotlight the words both in dialogue history and responses.
- An experimental demonstration is given showing the proposed approach can achieve superior performance compared with other existing methods, and ablation study shows the importance of the auxiliary task.

The remainder of this paper is organized as follows: in Section 2, the previous works performed on TOD response generation, including pipeline manner and end-to-end manner, are examined and discussed. In Section 3, the MN architecture that we have utilized is briefly presented, and then the proposed hierarchical memory network end-to-end model including Encoder and Decoder is explained. Then, in Section 4, the datasets, experiments settings, results analysis and baseline comparison of our work are described. Finally, conclusions and potential research directions are presented in Section 5.

2 | RELATED WORK

With the development of deep learning in natural language processing (NLP) research, especially in machine translation, the end-to-end modeling framework has achieved promising performance. Inspired by this methodology, many studies have introduced this data-driven method to build a task-oriented dialogue system. Rojas-Barahona et al. proposed a framework combining the pipeline manner and end-to-end trainable manner[3]. The model was end-to-end trainable using two supervision tasks and a modest corpus of training data. This was the first neural network-based end-to-end model that can conduct meaningful dialogues in a task-oriented application, however, it needs to create and execute well-formatted API calls to KB. Following this work, more robust end-to-end trainable task-oriented dialogue systems were developed, which directly input dialogue history and output dialogue responses without intermediate supervisions[4,6,20]. With this training strategy, human effort on state tracking and dialog policy learning can be reduced.

In this section, the related works are introduced from two aspects. One is MN-based methods that mainly use a memory network to model the dialogue history and KB. The memory network is also the backbone of our model. The other is other methods, which are end-to-end trainable but do not use memory networks.

2.1 | Works based on memory network

In addition to training strategies, a significant and commom problem that must be considered in TOD is how to deal with KB information, since it wouldn't be possible for a TOD to achieve a user's goal without accurate results from KB. To tackle unstructured dialogue history and manipulate structured KB information, Bordes et al. introduced MN to replace RNN to encode the dialogue history and KB, and achieved a promising result.[20] In their work, each dialogue utterance was represented as a bag-of-words. And in memory network, it was represented as a vector using a trainable embedding matrix. To store KB entries, the KB information was represented as a triplet, and every triplet was mapped to a vector[27]. However, due to the complexity of generating responses using data in different formats (text and KB), conventional sequence-to-sequence (Seq2Seq) model with MN failed to effectively produce correct words for a response. Eric et al. proposed a copy-augmented mechanism to alleviate this problem[6]. Further, Madotto et al. proposed a Mem2Seq model utilizing the copy-augmented mechanism to strengthen the ability to copy words from dialogue history or KB[23].

Different from Mem2Seq which combined dialogue history memory network and KB memory network, Lin et al. constructed a heterogeneous memory network to query history and KB in turn during the encoding and decoding stage[24]. Notably, a triple soft-gate, including vocabulary, dialogue history, and KB, was built in this work to better generate a response step by step. Meanwhile, some works believed that encoder module doesn't need to consider KB information. Thus, Chen et al. only considered the dialogue history memory network in the encoder and then incorporated the KB memory network in the decoder[25]. Following Chen's work, Reddy et al. proposed a multi-level KB memory network to improve the model's ability to select the right KB entries[28]. In their model, the first level of KB memory network focused on the KB's query attention, the second level of KB memory network paid attention to the KB results, and the last level of KB memory network captured the KB attribute attention of corresponding results. Recently, Wang et al.[29] introduced the idea of dual learning and leveraged MN to encourage the model to generate response effectively.

Although the above works have achieved promising results, they mostly treated the dialogue history and KB information as token-level aggregating that ignores the the sentence structure information, which results in poor performance when the amount of new unseen information in a KB increases. Therefore, Raghu et al. explored the cascade memory network model to represent the context and KB to improve the ability to generate a proper response[30]. Further, Wu et al. wrote the hidden states of dialogue history into an external KB memory network providing contextualized information and used a global-to-local policy to further improve the ability to copy words[26].

2.2 | Works based on other methods

In addition to MN-based methods, some studies are working with other methods. Banerjee et al.[31] firstly introduced graph convolutional networks (GCN) into TOD-related research. They proposed a memory-augmented GCN that leveraged entity relation graphs in a knowledge base and the dependency graph associated with an utterance to compute richer representations for response generation. Unlike the previous literature, He et al.[32] presented a "Flow-to-Graph" framework, which utilized RGCN[33] to represent the relationship between KB entities and dialogue tokens. Balakrishnan[34] proposed using tree-structured semantic representations for the model's input and output. This approach is similar to traditional rule-based NLG systems but uses the Seq2Seq framework. Also, this method requires a lot of data pre-processing work but is more efficient for response generation.

To improve the capability of TOD transferring to a new domain, Henderson et al.[35] introduced a two-stage framework called pretain then finetune to enable response selection. Shalyminov et al.[36] also followed this two-stage framework. However, their method generated responses word by word rather than extractive response generation. Besides, Qin et al.[37] introduced a shared-private network to enhance the model transferring to a new domain. The framework learned shared information from all domains and specific knowledge from every domain. In order to effectively utilize the dialogue history and KB, He et al.[38] proposed a "Two-Teacher One-Student" learning framework for TOD. Moreover, they adopted a generative adversarial network (GAN) to transfer knowledge from two teachers (dialogue history and KB) to the student (generator), which relaxed the rigid coupling between the student and teachers.

The weakness of the works mentioned above is the lack of consideration for online interactions. Liu et al.[39] and Wang et al.[40] addressed the problem of data ever-changing (including the user demands and KB, etc.) with the help of reinforcement learning and incremental learning methods, respectively.

With no surprise, all these mentioned methods perform well compared with the traditional models. Nevertheless, existing approaches still ignore the fact that not each turn of dialogue history is helpful for the dialogue response generation, which makes it struggle to perform well in long-turn interactions. Different from the aforementioned methods, we propose a hierarchical MN with an utterance-level MN and a token-level MN to capture the critical information.

3 | PROPOSED FRAMEWORK

To focus on critical contexts of dialogue history, an encoder-decoder neural conversation model augmented with a hierarchical MN mechanism is introduced in this paper, as shown in Fig. 2 and Fig. 3. The encoder module includes three steps, which are described below.

Step 1, each utterance of dialogue history is encoded using RNN to build utterance-level MN (coarse-grained filter). The utterance of the user's current turn is also embedded using RNN to get query vector q. Note that all the RNNs in the encoder module share parameters. The vector q is then used to execute multiple hops on the utterance-level memory network to output attention distribution p^k, which are coarse-grained weights of each utterance.

Step 2, using the tokenized dialogue history constructs the token-level MN (fine-grained filter), which is then multiplied by p^k for weighting. Further, token-level attention distribution, i.e., fine-grained weights, is computed by q interacting with weighted token-level MN. Through the above steps, irrelevant information can be filtered out.

Step 3, using the output q^k of weighted token-level MN query the KB memory network (dialogue-related KB information filter) and get output q^t. Note that, q^t is the initialized hidden state in the decoding stage.

In the decoder module, a gated-mechanism and RNN are used to generate natural response word by word from the vocabulary, token-level dialogue history, or KB, respectively.

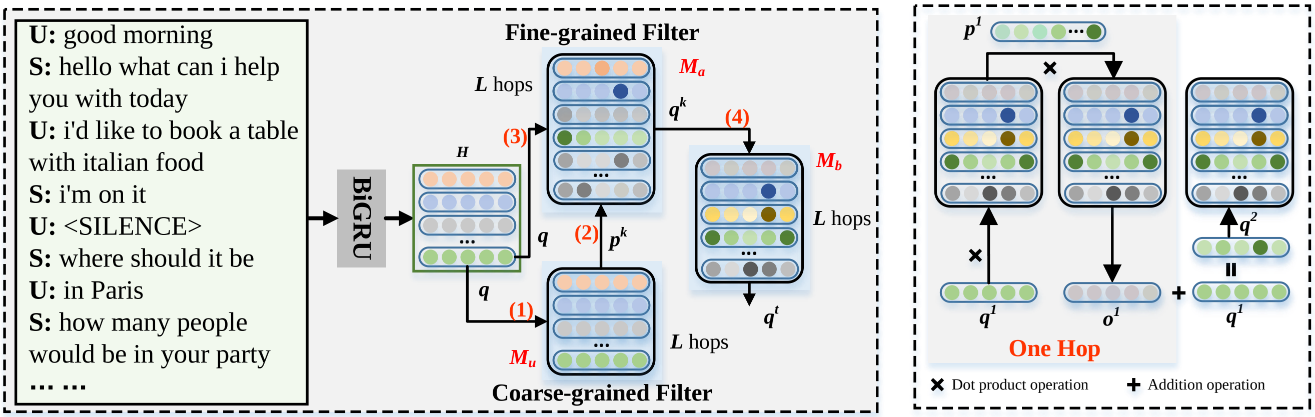
3.1 | Task Definition

Given a multi-turn dialogue history between a user and a system, a n-turned dialogue utterances is represented as $X = \{(u_1, s_1), (u_2, s_2), \dots, (u_n,)\}$, where u denotes the utterance from USER and s denotes the utterance from SYSTEM. Each utterance of X can be denoted word by word as $\tilde{\mathbf{x}} = (x_1, x_2, \dots, x_m)$, $\tilde{\mathbf{x}} \in \{u, s\}$, where m is the length of tokens in each utterance. The dialogue-related KB information is represented as $B = (b_1, b_2, \dots, b_l)$, where l is the length of KB.

The goal of the task-oriented dialogue response generation is defined as to generate response s_n (i.e., y), formulated as $y = (y_1, y_2, \dots, y_j)$, word by word given the dialogue history X and related KB B. Formally, the probability of a response is defined as

$$p(y|X, B) = \prod_{t=1}^{j} p(y_t|y_1, \dots, y_{t-1}, X, B)$$

where y_t represents an output token, j denotes the length of response.


U: good morning
S: hello what can i help you with today
U: i'd like to book a table with italian food
S: i'm on it
U: <SILENCE>
S: where should it be
U: in Paris
S: how many people would be in your party
… …

(a)

(b)

FIGURE 2 The left figure (i.e. (a)) is the encoder module of the proposed end-to-end framework, and the right figure (i.e. (b)) is a brief sketch of multi-hop operations of a memory network. Each utterance of dialogue history is encoded into hidden states, and then the representations are used to construct an utterance-level MN (i.e., M_u, the coarse-grained filter). A token-level MN (i.e., M_a, the fine-grained filter) is built using bag-of-words on dialogue history. Red numbers in Fig. (a) denote the procedure orders in the encoder module. In step (1), a query vector q interacts with M_u to output the coarse-grained weights (p^k) of each utterance. Then in step (2), using p^k times M_a to acquire a weighted $M_a{}'$, followed by step (3), using the query vector q to interact with $M_a{}'$ and output q^k. In step (4), the output q^k of the weighted token-level memory network is employed as the updated query vector to read KB memory network. Note that all memory networks (i.e., M_u, $M_a{}'$, M_b) execute L hops.

3.2 | Memory network

In this Section, the MN structure and its basic operations are introduced, including read and write MN, and multi-hop MN update, for the convenience of understanding the following sections. A brief diagram is depicted in Fig. 2 (b).

Given any sequence of n-length tokens $S = \{t_1, t_2, \dots, t_n\}$ to be embedded into memory network M^k, where $M^k = (m_1, m_2, \dots, m_n)$ donates the k-th MN representation, and m_i is the i-th memory item of MN. m_i can be represented as a vector using training embedding matrix A, i.e., $m_i = A\varphi(t_j)$, where $\varphi(\cdot)$ is a embedding function which can use GloVe[41] or bag-of-words, etc. To read the MN, a query vector is needed to interact with every memory item. The match between the query vector and memory item is computed by inner product. Moreover, it can loop over L hops and compute the attention weights at each hop k using

$$p^k = \mathrm{Softmax}(m_i^k \cdot (q^k)^{\mathrm{T}}) \tag{1}$$

where q^k is the query vector, T denotes transpose operation, and $m_i{}^k$ is memory item. Here, p^k is a soft memory selector that decides the memory relevance with respect to the query vector. Note that during the encoding phases, the softmax function is replaced by Sigmoid nonlinear function. Then, the readout vector o^k is the sum of memory matrix m^{k+1} with the corresponding attention weights p^k

$$o^k = \sum_i p_i^k \cdot m_i^{k+1} \tag{2}$$

To update the next hop, the readout vector and query vector of the k-th hop are summed to get query vector for the $(k+1)$-th hop. Therefore, the memory can be iteratively reread to look for additional pertinent information using the updated query vector q^{k+1} as

$$q^{k+1} = q^k + o^k \tag{3}$$

Note that the memory layers can be extended and stacked for L hop operations.

To drop the parameters of the multi-layer MN, the two training strategies[42] have been adopted. The first one is adjacent weight tying scheme, i.e., the output embedding for one layer is the input embedding for one above. The second one is layer-wise weight tying scheme, i.e., the input and output embeddings are the same across different layers.

3.3 | Encoder

The encoder encodes the dialogue history and KB information into fixed dimension vectors. Dialogue semantic representations at different granularities are needed to acquire (utterance-level and token-level). RNN is used to encode each utterance in the dialogue history into hidden states to obtain utterance-level semantic representations, which are then used to construct the utterance-level MN. On the other hand, as the existing research shows[23,24,25,26], MN is a proper encoder to map the dialogue history and KB into continuous low-dimensional representations. Here MN is adopted as our encoder to gain token-level semantic representations and KB representations.

Coarse-grained Filter: Given n-turned dialogue history $X = \{(u_1, s_1), (u_2, s_2), \dots, (u_n,)\}$, each single utterance $\tilde{\mathbf{x}}$ is encoded into the hidden states h^e repeatedly applying BiGRU[43]. For the i-th word in utterance $\tilde{\mathbf{x}}$,

$$\hat{h}^e_i = \text{BiGRU}(\varphi^{emb}(x_i), \hat{h}^e_{i-1}) \tag{4}$$

where $\varphi^{emb}(\cdot)$ is a embedding function which maps token x_i to a vector. The last hidden states $\hat{h}^e$ of each utterance $\tilde{\mathbf{x}}$ is represented as h^e. Therefore, the hidden states of the whole dialogue history are represented as $H = (h^e_1, h^e_2, \dots, h^e_{2n-1})$.

The utterance-level memory network M_u is constructed using the encoded representation H. The last hidden state of the latest utterance by a user is used as our query vector q (i.e. h^e_{2n-1}) to interact with the utterance-level MN and the token-level MN. The query vector q is used to compute the utterance-level attention distribution p^k with Eqs. (1)-(3). In particular, p^k measures the different level of importance of each utterance, which is the coarse-grained filtering operation. Note that a 3-hop is adopted in our model.

As discussed in Section 2.1, a fatal shortcoming of MN is the lack of sequential information, so the utterance-level MN discussed here has two benefits. One is to focus on an important utterance, and the other is to incorporate the sequential information into the dialogue history representation.

Fine-grained Filter: As the existing works have shown[25,26], the words in the dialogue history are treated in a triplet format, i.e., (*Subject, Relation, Object*). *Subject* represents the role of a speaker, *Relation* denotes the number of which dialogue turn belongs to, and *Object* stores the dialogue context. For example, *"can i help you"* can be denoted as {(*$bot, turn1, can*), (*$bot, turn1, i*), (*$bot, turn1, help*), (*$bot, turn1, you*)}. A bag-of-words representation is utilized to embed them and get the token-level memory network M_a. During the decoding, when a position of token-level memory network is pointed to, *Object* is directly copied as the output.

Since not every utterance of dialogue history contributes to the generation of the current response, the computed utterance-level attention distribution p^k is used to product the token-level MN, and a weighted token distribution $M_a{}'$ is obtained,

$$M_a{}' = p^k \cdot M_a \tag{5}$$

Note that utterance-level attention distribution p^k has a different number of dimension from M_a. To perform the product operation between p^k and M_a, every attention in p^k is scaled up by the number of tokens in the corresponding utterances. Then the query vector q (i.e., h^e_{2n-1}) is used to compute the token-level attention distribution, and obtain the output vector q^k with Eqs. (1)-(3) after L hops. The token-level attention distribution gauge the various weights of each token, which is the fine-grained filtering opreation.

Dialogue-related KB Information Filter: KB memory network M_b is established and unrelated KB information is filtered out in this subsection. To represent KB entries, a triplet format is adopted. For example, *"willows_market distance 4_miles"*, it means the distance between the user's location and *"willows_market"* is four miles, and this KB entry can be represented as {(*willows_market, distance, 4_miles*)}. A bag-of-words representation is also used to embed them into fixed dimension vectors. Subsequently, using Eqs. (1)-(3) and query vector q^k, the output vector q^t of KB memory network M_b can be obtained after L hops.

Through the two-stage filtering operation and interaction with the KB memory network, the output q^t carries the contextual and KB information required for current response generation. Note that q^t will be the initialized input of succeeding decoder module.

3.4 | Decoder

The decoder adopts GRU and memory network to generate a response word by word. As shown in Fig. 3, the memory network M_c is a combination of token-level memory network M_a and KB memory network M_b. The reason for not using utterance-level

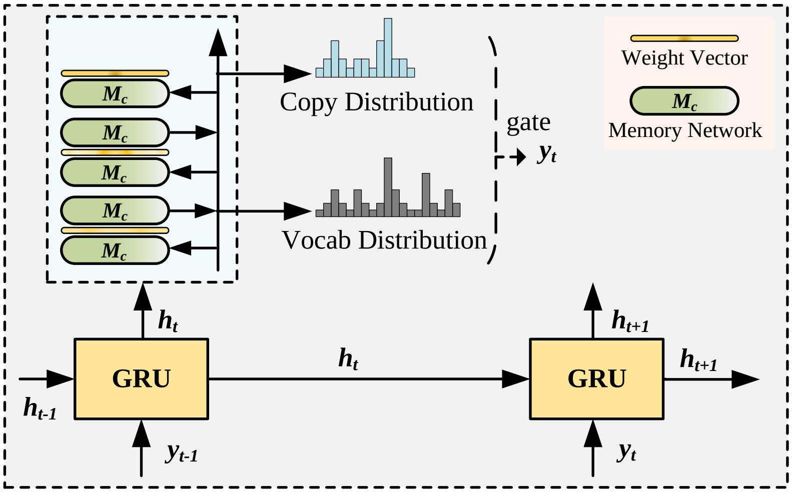


FIGURE 3 Decoder module of the proposed end-to-end framework. The vocabulary distribution and copy distribution are computed at the end of the first hop and last hop based on the combination (i.e., M_c) of the token-level memory network and KB memory network. To illustrate the decoding process more clearly and to save space, we represent the memory network as a rounded box.

memory network is that generator should copy words from history and KB, however, the utterance-level MN doesn't contain token information.

To generate dialogue response sequentially, the generator either uses a GRU to produce a response token from vocabulary or copies a specified entity from combined memory network M_c. Specifically, at decoding step t, the GRU uses the previously generated word y_{t-1} and hidden state h_{t-1} to generate the current hidden state h_t, as described as

$$h_t = \text{GRU}\left(M\left(y_{t-1}\right), h_{t-1}\right) \tag{6}$$

In particular, h_0 is q^t, which is the output vector of KB memory network.

At each decoding step, h_t queries the shared memory network M_c to generate the vocab distribution at the first hop using Eqs. (1)-(3), and the vocab distribution is computed by concatenating the query vector and readout vector of the first hop as

$$P_{vocab} = Softmax(W[h_t, o^1]) \tag{7}$$

where W is a trainable weight matrix. The readout o^1 of the first hop pluses the current hidden state h_t as the query vector to perform the next hop. The copy distribution P_{copy} is the attention output of last hop. The t-th generated token y_t is chosen using the gated-mechanism. Following the work on Mem2Seq[23] does, a special token is set at the last position of the memory network M_c as a sentinel. If the max pointer in the copy distribution points to the last position, i.e., the sentinel, y_t is from vocab distribution; Otherwise, y_t is from the copy distribution as expressed as

$$y_t = \begin{cases} \text{argmax}(P_{vocab}) & \text{if pointing to sentinel,} \\ \text{argmax}(P_{copy}) & \text{others .} \end{cases} \tag{8}$$

To learn the distribution of vocabulary P_{vocab} and P_{copy} in each time step, the loss in the t-th time step is the standard cross-entropy loss.

$$Loss_1 = -\frac{1}{T}\sum_{t=0}^{t=T}\sum_{i}\left(\log p_{ti}\right) \tag{9}$$

where p_{ti} represents the probability of the t-th word in $i \in \{P_{vocab}, P_{copy}\}$.

To strengthen the important words in the dialogue history and improve the ability of dealing with OOV problem, an auxiliary task is added in the decoding stage. In the auxiliary task, the label l^i is defined by checking whether the pointed words in the memory exists in the golden response. If so, $l_k^i = 1$, otherwise, $l_k^i = 0$. The auxiliary task is trained using a binary cross-entropy loss which is defined as

$$Loss_2 = -\sum_{k=1}^{n}\left[l_k^i \times \log l_k + \left(1 - l_k^i\right) \times \log\left(1 - l_k\right)\right] \tag{10}$$

Algorithm 1 Training Algorithm of the Proposed Model

```
Input: corpus $C$, model parameters $\theta$, initialize epochs $T$,batch number $N$, learning rate $lr$, hop number $L$, and $\alpha$.
Output: parameters of the proposed model
for each epoch in $[0, 1, 2, \dots, T]$ do
    shuffle training data $C$;
    for each mini-batch in $[0, 1, 2, \dots, N]$ do
        get mini-batch $\hat{C}$ from $C$;
        compute each dialog hidden state $H$ and corresponding query vector $q$ by Eq. 4;
        construct utterance-level MN $M_u$, token-level MN $M_a$, and KB MN $M_b$;
        for each hop in $[0, 1, 2, \dots, L]$ do
            use query vector and Eq. 1-3 to interact with $M_u$;    // for first hop, the query vector is q
        end for
        return $p_k$
        $M'_a = p^k \cdot M_a$;    // Eq. 5
        for  each hop in $[0, 1, 2, \dots, L]$ do
            use query vector and Eq. 1-3 to interact with $M'_a$;    // for first hop, the query vector is q
        end for
        return $q_k$
        for  each hop in $[0, 1, 2, \dots, L]$ do
            use query vector and Eq. 1-3 to interact with $M_b$;    // for first hop, the query vector is $q^k$
        end for
        return $p_t$
        use $q^t$ to initialize the decoding stage hidden state $h_0$;
        use Eq. 6, 7, 8 to generate every $y_t$;
        compute $Loss1$ using Eq. 9, and $Loss2$ using Eq. 10;
        use $\alpha$ to compute the Loss as Eq. 11;
        compute the gradient of $\nabla Loss$, update $\theta$ with Adam optimizer;
    end for
end for
return $\theta$
```

The parameters are jointly learned during the training stage by minimizing the weighted sum of two losses (α is a hyper-parameter). The entire training procedure of the proposed model is presented in Algorithm 1.

$$Loss = Loss_1 + \alpha Loss_2 \tag{11}$$

4 | EXPERIMENTS

4.1 | Datasets

To better evaluate the performance of the proposed model, several popular benchmark datasets have been used to conduct the experiments. They are : bAbI dialog dataset[20], standard multi-domain dialogue (SMD)[27], the second dialog state tracking challenge (DSTC2)[44], CamRest[45], and MultiWOZ 2.1[8]. A brief introduction to the datasets is given below.

The bAbI dataset consists of five tasks about restaurant reservations. Tasks 1 to 4 are API calls, refining API calls, recommending operations, and providing additional information (e.g., phone number or address, etc.), respectively. Task 5 is the union of Tasks 1-4. There are two test sets for each task, one follows the same distribution as the training set, and the other has OOV words.

The SMD dataset is a multi-domain dialogue dataset with three domains: calendar scheduling, weather information retrieval, and point-of-interest navigation. Compared with the bAbI dataset, this dataset has shorter conversation turns, but the user and

TABLE 1 Dataset statistics for bAbI, DSTC2, SMD, CamRest, and MultiWOZ 2.1

Task	**1**	**2**	**3**	**4**	**5**	**DSTC2**	**SMD**	**CamRest**	**MultiWOZ 2.1**
Avg. turns per dialog	6	9.5	9.9	3.5	18.4	9.3	2.6	5.1	5.6
Avg. KB triples	0	0	24	7	23.7	39.5	66.1	22.5	54.4
Avg. Sys words	6.3	6.2	7.2	5.7	6.5	10.2	8.6	10.8	15
Max. Sys words	9	9	9	8	9	29	87	39	48
Vocabulary	3747					1229	1601	902	3449
Train dialogues	1000					1618	2425	406	1839
Dev. dialogues	1000					500	302	305	117
Test dialogues	1000+1000 OOV					1117	304	135	141

TABLE 2 Hyper-parameters we use in the experiments.

	bAbI					**SMD**	**DSTC2**	**CamRest**	**MultiWOZ 2.1**
	T1	T2	T3	T4	T5				
Hidden Size	64	64	64	64	128	128	128	128	128
Dropout Ratio	0.3	0.3	0.3	0.7	0.1	0.2	0.2	0.2	0.1
Batch Size	8	8	8	8	8	8	8	8	16
Teacher Forcing Ratio	0.9								

system behaviors are more diverse. In addition, the system responses are variant, and the KB information is much complicated. Hence, this dataset requires stronger ability to interact with KBs, rather than dialog state tracking.

The DSTC2 dataset consists of real human-bot dialogues extracted from the Second Dialog State Tracking Challenge. A refined version of the data is used here which doesn't consider the dialogue state labels. Each dialogue is composed of user's and system's utterances, and API calls to the domain-specific KB for the user's queries.

The CamRest dataset contains human-to-human dialogues of the restaurant reservation.

MultiWOZ 2.1 is one of the most challenging datasets given its multi-domain setting, complex ontology, and diverse language styles. It is the corrected version of MultiWOZ 2.0[7], which contains 7 task domains, i.e., *Attraction, Hospital, Police, Hotel, Restaurant, Taxi, Train*. Follow Qin's[37] setting on MultiWOZ 2.1 dataset[1], we only use the single-domain dialogues belonging to *Attraction, Hotel, or Restaurant.*

Statistics of these five datasets are shown in Table 1.

4.2 | Training Details

All the training of the models has been implemented using PyTorch[2], and Adam optimizer[46] was used. The learning rate annealing started from 1e-3 to 1e-4 with a decay rate of 0.5. The embedding dimension was equivalent to the GRU hidden size that has been selected using grid-search over the development set between [64, 128]. The dropout ratio was set between [0.1, 0.7]. The multi-hop L was 3 with the adjacent weight sharing scheme[20], as the existing works[23,26,30] has proven that the model would perform poorly with too many hops(e.g., 6) or too small hops (e.g., 1). α was set equal to 1, selected from [0.4, 0.6, 0.8, 1.0, 1.1, 1.2] over the development set. A simple greedy strategy has been used as our decoding strategy. During the training stage, teacher-forcing scheme was used. Note that a small number of input source tokens have been randomly masked into UNK to simulate the OOV issue. For the bAbI dataset, the model with the best per-response accuracy over the development set in 200 epochs' training has been selected for testing. For the other datasets, the model selected for test was the one with the best BLEU score over the development set using an early stop training strategy. The average performance of each metric (see Section 4.3) over five runs are presented as final results. Details about hyper-parameters can be found in Table 2.

[1] The extended version of MultiWOZ 2.1 dataset can be found here: https://github.com/LooperXX/DF-Net

[2] https://pytorch.org/

4.3 | Evaluation Metrics

For a fair comparison, a set of commonly used metrics is employed for assessing specific aspects of the proposed model:

- **Per-Response Accuracy and Per-Dialogue Accuracy**: The per-response accuracy is the percentage of generated responses that exactly match their respective gold response. A prediction is correct only if each token output by the model matches the corresponding token in the gold response. Per-dialogue accuracy is the percentage of dialogs with all correctly generated responses. These two accuracy metrics show if a model is able to learn the distribution of reproducing factual details.

- **Bilingual Evaluation Understudy (BLEU)**: BLEU calculates the n-gram precision, which is a fraction of n-grams in a candidate text presented in any reference texts. BLEU metric is commonly employed in evaluating machine translation systems[47], and has also been used in the literature for evaluating dialogue systems of chat-bot and task-oriented variety[20,48]. The study by Sharma et al. shows that this metric has a strong correlation with human assessments on task-oriented datasets[49]. Therefore, average BLEU score is calculated over all the responses generated by the system. Note that the Moses *multi-bleu.perl* script was adopted to calculate the BLEU score in evaluation.

- **Entity F1**: Each system response in the test data defines a gold set of entities. To compute entity F1, the entire set of system dialogue responses is micro-average and the entities in plain text are compared. The entities in each gold system response are selected by a predefined entity list. This metric evaluates a model's ability to generate relevant entities from a underlying knowledge base and to capture the semantics of a user-initiated dialogue flow. Note that in SMD and MultiWOZ 2.1 datasets, the test set contains dialogues from all three domains, thus a per-domain entity F1 as well as an aggregated dataset entity F1 is computed.

4.4 | Baseline Models

To better show our model's ability, some models from existing works have been selected as baseline models. These models are Seq2Seq+Atten[50], Mem2Seq[23], Heterogeneous Memory Network (HMNs)[24], Global-to-Local Memory Pointer networks (GLMP)[26], BoSsNet[30], Dual Dynamic Memory Network (DDMN)[29], FG2Seq[32], Dynamic Fusion Network (DF-Net)[37], Template-guided Hybrid Pointer Network (THPN)[51], and GPT2[14] . Below is a brief description of the models.

- **Seq2Seq +Atten**: A model with simple attention over the input context at each time step during decoding.

- **Mem2Seq**: The model uses a memory network based approach for attending over dialog history and KB triples. During decoding, at each time step, the hidden state of the decoder is used to perform multiple hops over a single memory which contains both dialog history and the KB triples, to obtain the pointer distribution used for generating the response.

- **HMNs**: The model encodes the dialog history and KB entries into different memory networks and adopts the query vector to interact with dialogue memory network and KB memory network in turn. The generated words are chosen by vocabulary, history, or KB during the decoding phase.

- **BoSsNet**: The model adopts an encoder-decoder architecture with a novel bag-of-sequence memory, including one higher-level flat memory and one lower-level discrete memory. The architecture facilitates the disentangled learning of the response's language model and its knowledge incorporation.

- **GLMP**: The model adopts a global-to-local pointer mechanism to query dialogue history and knowledge base. This model uses the global pointer to acquire the related information in the encoding stage and computes the local pointer for coping entities based on the global pointer during the decoding stage.

- **DDMN**: The model constructs a dual dynamic memory network. It contains a dialogue memory manager and a KB memory manager to dynamically model long dialogue context and effectively incorporate KB information into dialogue generation.

TABLE 3 Results of per-response and per-dialogue accuracy (in the parentheses) on bAbI dataset with 3 hops. Per-dialog accuracy represents the rate of complete dialogues. The bold numbers in the table represent the best results in the corresponding task, and the underlined numbers represent the second-best results.

task	Seq2Seq+Atten	Mem2Seq	HMNs	BoSsNet	GLMP	THPN	GPT2	Ours
T1	100(100)	100(100)	-	100(100)	100(100)	100(-)	-	**100**(**100**)
T2	100(100)	100(100)	-	100(100)	100(100)	100(-)	-	**100**(**100**)
T3	74.8(0)	94.7(62.1)	93.6(56.1)	95.2(63.8)	96.3(75.6)	95.8(-)	-	**97.8**(**77.0**)
T4	57.2(0)	100(100)	100(100)	100(100)	100(100)	100(-)	-	**100**(**100**)
T5	98.4(87.3)	97.9(69.6)	98.0(69.0)	97.3(65.6)	99.2(88.5)	**99.6**(-)	91.5(32.9)	99.4(**89.2**)
T1-OOV	81.7(0)	94.0(62.2)	-	100(100)	100(100)	-	-	**100**(**100**)
T2-OOV	78.9(0)	86.5(12.4)	-	100(100)	100(100)	-	-	**100**(**100**)
T3-OOV	75.3(0)	90.3(38.7)	92.5(48.2)	95.7(66.6)	95.5(65.7)	-	-	**96.2**(**68.3**)
T4-OOV	57.0(0)	100(100)	100(100)	100(100)	100(100)	-	-	**100**(**100**)
T5-OOV	65.7(0)	84.5(2.3)	84.1(2.6)	91.7(18.5)	92.0(**21.7**)	-	73.5(3.0)	**92.8**(21.4)

- **FG2Seq**: This work proposes a Flow-to-Graph framework that utilizes RGCN to represent the relationship between KB entity and dialogue tokens. The framework encodes knowledge by considering inherent structural information of the knowledge graph and latent semantic information from dialog history.

- **DF-Net**: This work presents a shared-private network to enhance the model transferring to a new domain. The framework learns shared information from all domains and specific knowledge from each domain. In addition, a novel dynamic fusion network is proposed to automatically exploit the relevance between the target domain and each domain.

- **THPN**: The model proposes a template-guided hybrid pointer network, which retrieves several potentially relevant answers from a pre-constructed domain-specific conversational repository as guidance answers, and incorporates the guidance answers into both the encoding and decoding processes.

- **GPT2**: The model is a large autoregressive pre-trained language model. The small size GPT2 was used with no more than 1024 knowledge base tokens to fine-tune the model on the datasets.

4.5 | Results and Analysis

The experimental results on each of the benchmark datasets are discussed below.

bAbI: The results on bAbI dataset are given in Table 3. Our model with 3 hops achieved the best performance in most of the tasks. In particular, proposed model achieved 1.5, 0.7, 0.2, and 0.8 points increase compared with the highest per-response accuracy of baseline models, i.e., GLMP, in T3, T3-OOV, T5, and T5-OOV, respectively. More importantly, the proposed method has demonstrated the same trend on the per-dialogue accuracy, as shown in the parentheses in Table 3, expect in the T5-OOV task. On the other hand, our model achieved a 0.2 drop on T5 in per-response accuracy compared with the THPN model. These results indicate our hierarchical MN mechanism can assist the model to capture more important information in the dialogue procedure and promote this information to retain in the following steps. However, it has also been observed that our model performed slightly better than the baseline models in T3 and T3-OOV. T3 is a recommended task that is closely related to KB. The slight improvement in T3 may be associated with the lack of fine-grained filtering operations on KB. It was found that all the models achieved a lower digit in T3 compared with the other tasks. Shown in Fig.1, when a dialog system performs recommended tasks, it should understand that different numbers of stars represent different levels of restaurants. But all the baseline systems, even ours, have failed to do this. This remains a challenging problem for further studies.

Furthermore, it has been found that the MN-based models, i.e., baseline models, except for Seq2Seq+Atten[50], and GPT2[14], demonstrated a better performance in the most tasks than the canonical sequence-to-sequence model Seq2Seq+Atten. This result may be attributed to the weak ability of the Seq2Seq architecture in processing formatted KB data. A more pronounced trend can be observed on F1 metric on SMD dataset because the SMD provides more KB entries. In further analysis of T5 performance, it was observed that Seq2Seq+Atten had a better performance than Mem2Seq, HMNs, and BoSsNet, whereas it

TABLE 4 Comparison of the proposed model with baseline models on SMD dataset.

	Seq2Seq+Atten	Mem2Seq	HMNs	BoSsNet	GLMP	DDMN	FG2Seq	THPN	GPT2	Ours
BLEU	9.3	12.6	14.5	8.3	13.9	15.8	**16.8**	12.8	16.5	14.9
Entity F1	19.9	33.4	43.1	35.9	60.7	60.7	61.1	37.8	57.6	**62.1**
Schedule F1	23.4	49.3	61.3	50.2	72.5	69.3	**73.3**	50.0	70.8	72.2
Weather F1	25.6	32.8	40.3	34.5	56.5	**64.7**	57.4	37.9	57.2	58.6
Navigation F1	10.8	20.0	32.3	21.6	54.6	53.2	56.1	27.5	48.3	**56.9**

TABLE 5 Results on DSTC2 and CamRest datasets.

		Seq2Seq+Atten	Mem2Seq	HMNs	BoSsNet	GLMP	DDMN	FG2Seq	THPN	GPT2	Ours
DSTC2	BLEU	56.6	55.3	56.4	-	58.1	-	-	59.8	**65.8**	61.9
	Entity F1	67.1	75.3	77.7	-	67.4	-	-	76.8	72.8	**78.2**
CamRest	BLEU	5.9	12.6	-	15.2	18.2	18.7	20.2	12.9	15.2	**20.7**
	Entity F1	21.4	33.4	-	43.1	52.7	59.1	62.1	30.9	51.0	**63.0**

TABLE 6 Results on MultiWOZ 2.1 dataset.

	Seq2Seq+Atten	Mem2Seq	BoSsNet	GLMP	DDMN	DF-Net	GPT2	Ours
BLEU	4.5	6.6	5.7	6.9	11.5	9.4	**14.7**	13.0
Entity F1	11.6	21.6	25.3	32.4	34.2	35.1	30.7	**35.5**
Restaurant F1	11.9	22.4	26.2	38.4	38.5	**40.9**	34.3	39.5
Hotel F1	11.1	21.0	23.4	28.1	31.1	30.6	27.9	**32.7**
Attraction F1	10.8	22.0	24.8	24.4	34.1	28.1	26.3	**34.8**

appeared an opposite situation in T5-OOV. This may indicate that the attention mechanism is efficient for generative issues, but it's also inadequate of improving the OOV problem.

The pre-trained language model GPT2 has also employed in T5 and T5 OOV tasks to explore the GPT2's power to handle the TOD issues. The GPT2 model achieved poorer per-response and per-dialogue accuracy in both tasks compared with other models. This may be because the KB data format during fine-tuning is different from the natural language used for pre-training the language model.

SMD: The results on SMD dataset is given in Table 4. and as shown, our model outperformed the best baseline model, i.e., FG2Seq, on Entity F1 and Navigation F1 metrics by 2.3% and 1.4%, respectively. In addition, the proposed model achieved the second-best score on Weather F1 and was on par with the FG2Seq model on Schedule F1. On the other hand, three baseline models produced a better performance than the proposed model on metric BLEU. This may be because the SMD dataset has a shorter dialogue turn, as shown in Table 1. This characteristic makes our proposed two-stage filtering strategy not work well. Although the three models showed a better performance on the BLEU metric, our model gave a better F1 score overall. This owes to the two-stage filtering operation providing fine-grained and precise information to assist the model in capturing the related KB entries. These observations prove that our model has a more solid ability to deal with complex KB information.

Besides, it can seen that the scores of all the MN-based models were closer on BLEU metric than on F1 metric. This indicates that generating dialogue responses with the correct entity information is more challenging than developing natural ones. In particular, compared with other baselines, the Seq2Seq+Atten framework had comparable results on BLEU metric but achieved the worst performance on F1 metric. This also demonstrates that developing a response with correct entities is challenging. Notably, the GPT2 model achieved a second-best BLEU score, but a shallow F1 score. This demonstrates that the pre-trained language models have a solid capability to generate natural responses, but have limitations in handling OOV tokens (i.e., knowledge base tokens).

TABLE 7 Performance of models on updated bAbI dataset and updated SMD dataset.

Model	**bAbI**		**SMD**	
	T5	T5-OOV	BLEU	Entity F1
BoSsNet	90.4(37.9)	83.7(7.0)	3.7	21.8
GLMP	87.2(12.7)	84.0(5.9)	4.7	21.1
Ours	**92.6(42.3)**	**86.0(12.3)**	**7.7**	**32.8**

TABLE 8 Results of ablation study on bAbI, SMD, CamRest, and MultiWOZ 2.1 datasets. (w/o) represents the results of the proposed model without corresponding component.

	bAbI		**SMD**		**CamRest**		**MultiWOZ 2.1**	
	T5	T5-OOV	BLEU	Entity F1	BLEU	Entity F1	BLEU	Entity F1
WHOLE	99.4(89.2)	92.8(21.4)	14.9	62.1	20.7	63.0	13.0	35.5
-w/o t	98.7(88.0)	90.6(18.7)	14.2	55.7	19.5	58.2	12.1	34.2
-w/o U	98.4(84.9)	85.2(15.3)	13.9	54.8	18.8	59.2	12.4	33.6
-w/o T	85.2(68.3)	79.5(7.8)	12.3	36.9	17.1	42.8	10.5	22.5

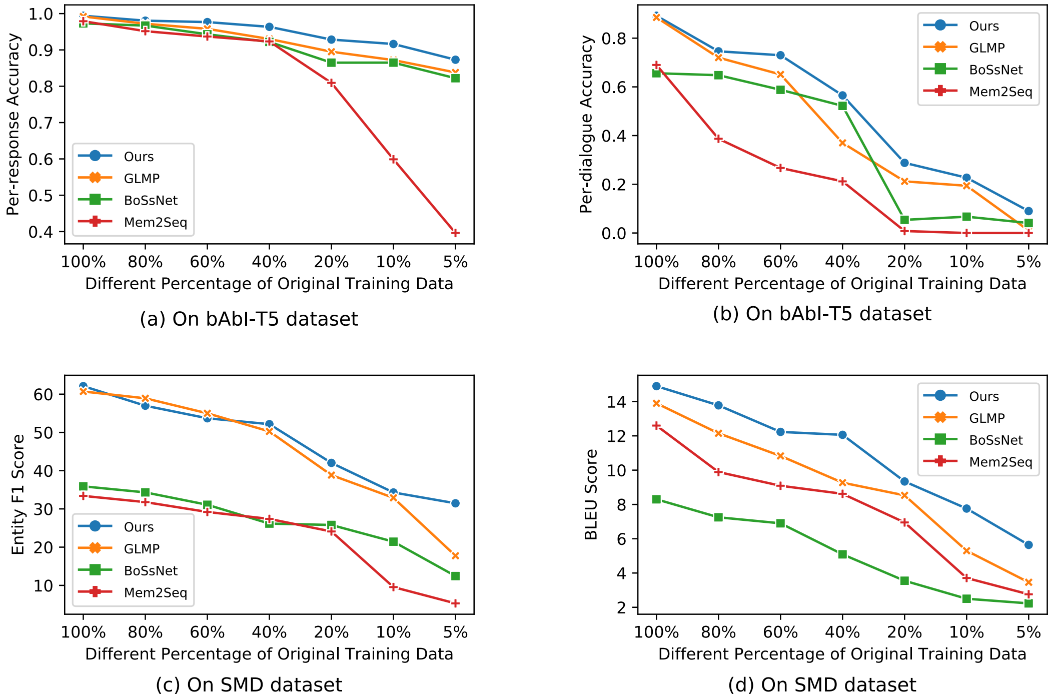
DSTC2 and CamRest: The results on DSTC2 and CamRest datasets are shown in Table 5. Note that our model obtained the best performance on entity F1 metric on both datasets and the highest score on BLEU on DSTC2 dataset. Moreover, our model showed the best score on BLEU compared with all baseline models except GPT2. This demonstrates that the proposed model can produce a more fluent response. Another noteworthy observation is that the GPT2 model on CamRest dataset performed worse on BLEU metric than on DSTC2 dataset. The reason for this is that CamRest has fewer training instances for fine-tuning than CamRest.

MultiWOZ 2.1: MultiWOZ 2.1 dataset has the longest system response, and therefore is a more challenging multi-domain dataset, as seen in Table 1. Both BLEU and F1 metrics give a lower score than another multi-domain dataset (i.e., SMD). This further proves that MultiWOZ 2.1 dataset is more challenging. As shown in Table 6, the proposed model had the best performance on Entity F1, Hotel F1, and Attraction F1 and the second-best result on BLEU and Restaurant F1, demonstrating that our model holds the solid ability to deal with the complex situation. Specifically, our BLEU score was 13.0 points, an increase of 13% compared to the best baseline model for this metric except GPT2, proving that the proposed two-stage filtering mechanism was effective.

Note that the GPT2 model achieved the best BLEU score, indicating that the large-scale pre-trained language model can generate natural responses without much effort. Meanwhile, it was observed that the F1 scores of the GPT2 model were lower by a big margin compared with the other baseline models on all datasets. This may be because of the structured KB data used in the fine-tuning phase that corrupts the original pre-training framework. As a revelation, we will also explore the capabilities of the proposed model using Transformer[52] in future work.

Further Results: To verify the validity of the proposed model, further experiments were conducted on the updated bAbI and SMD datasets[53]. The user utterances in the original datasets are straight-forward and always stick to the user's goal without any diversity and novelty, which is unusual with our reality in natural language. The updated bAbI and SMD datasets introduce naturalistic variation dialogues through the Natural Conversation Framework (NCF)[54] to alleviate this problem. Therefore, the updated datasets are more challenging than the original datasets. The reported results on the updated datasets of GLMP and BoSsNet from Ganhotra et al.[53] are adopted.

The performance of the proposed model on the updated bAbI and SMD datasets are shown in Table 7. It was found that there was a significant drop in performance of all models on both datasets on all metrics. Specifically, per-response accuracy in Task 5 of bAbI had the smallest decline (7%-12%) compared with other metrics. There was a 60%-85% decrease in per-dialogue accuracy on bAbI, while a 40%-60% reduction on BLEU and Entity F1. Overall, the proposed model achieved the best performance on both updated datasets, which proves our model is effective.



(a) On bAbI-T5 dataset

(b) On bAbI-T5 dataset

(c) On SMD dataset

(d) On SMD dataset

FIGURE 4 The performance of the proposed model and baseline models on bAbI-T5 (a, b) and SMD (c, d) datasets with the decrease in training set size.

4.5.1 | Ablation Study

To test our proposed hypothesis, we assess the value of each model element on bAbI, SMD, CamRest, and MultiWOZ 2.1 datasets by removing it from our framework. Table 8 reports the metric scores for various configurations of our model. WHOLE represents the framework we proposed, w/o t represents our model removing the auxiliary task t, w/o U represents our model without using the utterance-level memory network, and w/o T means that our model does not use the token-level memory network.

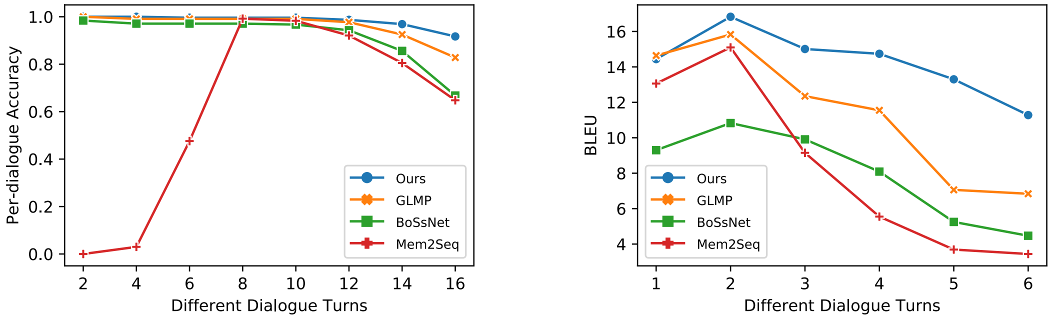
Take the SMD dataset as an example, by removing the auxiliary task, a 4.7% BLEU drop and a 10.3% entity F1 drop have been observed. This indicates that the auxiliary task plays an important role in improving the model's ability to copy KB entities from the KB results and dialogue history.

From the last two rows in Table 8, it can be observed that, when the utterance-level memory network module was removed, all metrics had a drop compared with the proposed model. This observation proves the utterance-level memory network filters out irrelevant utterances in the dialogue history. In addition, all metrics showed a more significant drop when removing the token-level memory network than the utterance-level memory network. This indicates that the token-level memory network can capture richer information than the utterance-level memory network. It is our view that the token-level Mn can embed more fine-grained information because each dialog token is represented as a tuple format , as described in the ENCODER section.

On the other hand, a trend on BLEU metric can be observed where BLEU does a slighter drop than the F1 metric. This proves that RNN as a decoder are strong and stable.

4.5.2 | Robustness for Incomplete Training Set

It is well known that deep learning methods require a large amount of training data to fit the parameterized functions. However, generating high-quality labeled data is time-consuming and labor-intensive. To verify the robustness of the proposed model in a small amount of training dataset, a group of experiments was performed on bAbI and SMD datasets by gradually decreasing the training set size. Fig. 4 illustrated the performance of baselines and proposed model on bAbI Task 5 and SMD datasets when the training dataset was reduced from 100% to 5%. The upper two figures (a) and (b) display the trend of per-response accuracy



(a) On bAbI-T5 dataset

(b) On SMD dataset

FIGURE 5 The performance of the proposed model and baseline models on bAbI-T5 (a) and SMD (b) datasets with the increasing of dialogue turns.

(a) and per-dialogue accuracy (b) of Task 5 with different proportions of training set on bAbI. The lower two figures (c) and (d) showed the Entity F1 (c) and BLEU (d) scores with different proportions of training set on SMD.

For Fig. 4 (a), it can be seen that the trend of baselines and the proposed model kept stable when the proportion exceeded 40%. Remarkably, the Mem2Seq model had a sharp drop when the training set size was reduced below 40%. For Fig. 4 (b), it was observed that all the models had an apparent reduction when declining the proportion of the training set. However, the proposed model consistently achieved the highest accuracy on the corresponding training set, indicating that our model is more substantial for dealing with inadequate training. For Fig. 4 (c) and (d), it was found that a more pronounced gap appeared on Entity F1 between GLMP and the other two baselines than the BLEU. Note that, this also coincides with the observation in Section 4.5, i.e., the scores of all the MN-based models are closer on BLEU metric than on F1 metric.

In general, the proposed model maintained the best performance on variant proportions of the training set of both datasets, which demonstrates our model possesses higher robustness. Furthermore, from the overall four figures, it can be observed that the performance would drop when the training set's proportion decreased below 40%. This indicates a parameterized model requires a certain number of training sets to fit the function if starting from scratch. From this perspective, the foundation models[55] may be a promising scenario.

4.5.3 | Ability to Deal with Increasing Dialogue Turns

As the dialogue proceeds, the longer the dialogue history is presented, the more useless information may carry. Therefore, the capacity of a dialogue system to deal with long and enriched contexts will be essential for developing responses. To verify our model's ability to manage different numbers of turns' dialogue, a series of experiments were conducted to this end. Fig. 5 illustrated the performance of increasing the dialogue's turn on bAbI Task 5 and SMD datasets. The left figure (a) presented the per-dialogue accuracy of Task 5 on bAbI dataset. The right figure (b) gave the BLEU scores on SMD dataset.

In Fig. 5 (a), it can be seen that the proposed model, GLMP, and BoSsNet had a stable trend when the dialogue's turn increased. However, after ten dialogue turns, the accuracy of GLMP and BoSsNet started to decrease, while the proposed model continued to maintain steady. In particular, the Mem2Seq achieved a low accuracy at the 2, 4, 6 turns' dialogue. After eight turns, it showed a similar trend to GLMP and BoSsNet. On the other hand, the SMD dataset has a shorter dialogue history than the bAbI dataset, as shown in Table 1. In Fig. 5 (b), it was observed that all the models demonstrated similar trends, but the proposed model revealed a relatively smooth accuracy curve compared to the baseline models. These observations demonstrate that the proposed model can utilize the dialogue history more effectively as the dialogue turns increases than the baseline models.

4.5.4 | Visualization of Hierarchical MN

To more visually demonstrate the proposed framework's performance, the attention weights of several components in the encoding and decoding stages are visualized. As shown in Fig. 6, the attention vectors in the last hop are presented for utterance-level memory network, token-level memory network, KB memory network in the first three columns and the attention vector of each decoding time step is displayed in the fourth column. Y-axis represents memory items being able to copy by generator, including the dialogue history and KB entries. As shown, the conversation was about gas stations, and our model needs to carry out

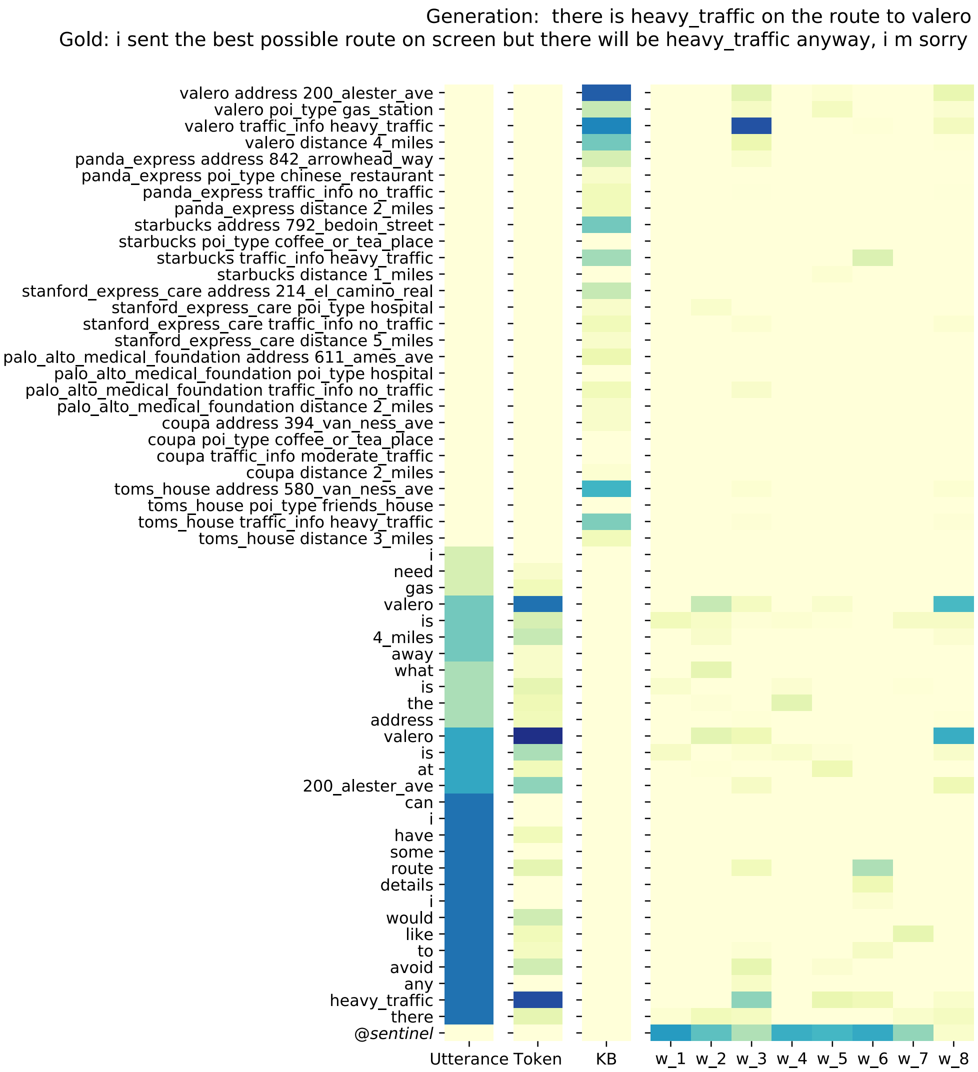
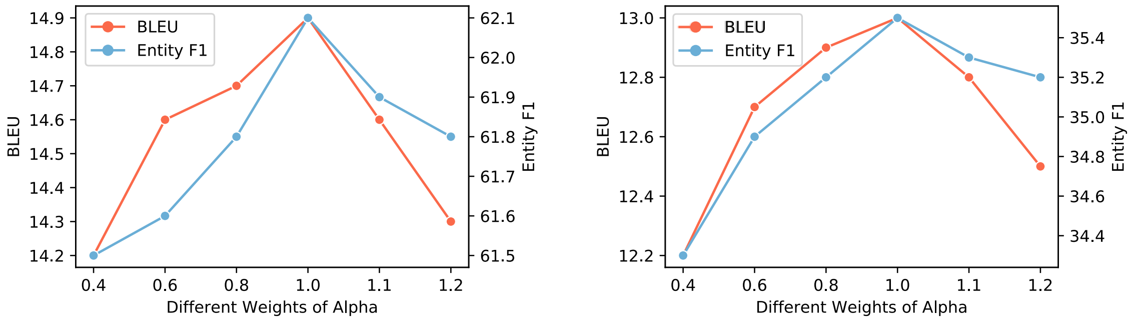


FIGURE 6 Memory attention visualization on SMD dataset. The first three columns are the attention vectors in the last hop for utterance-level MN, token-level MN, and knowledge base MN, respectively. The fourth column is the attention vectors of each decoding time step. The last row in the figure named *@sentinel* represents whether the generated token is from the vocabulary or copied from the KB MN. The darker color appears in the figure, the higher score achieves.

a natural and plausible response based on the question *"can I have some route details? I would like to avoid any heavy_traffic there"* asked by the driver in the last turn. The ground truth and our generated response are at the top of Fig. 6.

It can be seen that in the first two columns, the upper part is zero, and in the third column, the lower part is zero. This is because when the query vector interacted with utterance-level memory network and token-level memory network, there was no interaction between the query vector and KB memory network and vice versa. As observed, in the first column, each token belonged to the same utterance with the same attention vector. In the second column, the *valero* entity had a dark color, which means that *valero* plays a vital role in the context. Moreover, the latter had a darker color than the former, which holds the same trend as the utterance-level attention. On the other hand, in the fourth column, the KB entries related to the *valero* had a higher attention score than others. Note that the evidence (i.e., the triplet (*valero traffic_info heavy_traffic*)) to traffic conditions had the highest score, which proves that our model can look up the correct KB entries and filter out the irrelevant ones. Further, it has been found that the entity *heavy_traffic* and *valero* were successfully copied in Step 3 and Step 7, where the pointer didn't point to the *@sentinel* at the corresponding step. The visualized weights as shown in the last row in Fig. 6. This proves that the copy mechanism is essential in improving the performance of developing responses.



(a) On SMD dataset

(b) On MultiWOZ 2.1 dataset

FIGURE 7 The performance comparison of the proposed model when α takes different weights.

TABLE 9 Examples were generated by the proposed model and baseline models given the dialogue history and knowledge base.

Affairs	**party**	**date**	**time**	**agenda**	**room**
football_activity	tom	the_5th	9am	-	-
dentist_appointment	alex	sunday	2pm	-	-
conference	management	friday	10am	discuss_the_merger	conference_room_50
doctor_appointment	father	wednesday	7pm	-	-
swimming_activity	sister	the_13th	1pm	-	-
doctor_appointment	alex	the_5th	6pm	-	-
dentist_appointemnt	jon	thursday	2pm	-	-
meeting	management	the_3rd	2pm	discuss_the_company_picnic	100_conference_room

usr-1	what time is my doctor apartment and who is it with ?
sys-1	are you talking about the doctor_appointment on wednesday or the one on the_5th ?
usr-2	the one on wednesday
sys-2	your doctor_appointment on wednesday is at 7pm with your father
usr-3	**and how about the one on the_5th ?**
Gold	your doctor_appointemnet on the_5th is at 6pm with alex
Seq2Seq+Atten	the doctor_appointment is at the_5th
GLMP	your doctor_appointemnet is on the_5th at 2pm
HMN	your doctor_appointment on the_5th is at 6pm
BoSsNet	your doctor_appointment on the_5th is at 7pm
OURS	your doctor_appointment on the_5th is at 6pm with alex

4.5.5 | Effect of Different Weights of Alpha

To analyze the effect of α when it picks up different weights, a group of experiments was conducted to display the diverse performance on SMD and MultiWOZ 2.1 datasets, as shown in Fig. 7.

An apparent observation can be seen that the proposed model achieved the best performance when the α was equal to 1.0. This demonstrates the auxiliary task is pretty crucial for our model. It also has been found that the BLEU performance showed a sharper drop than Entity F1 on both datasets when the number chosen for α exceeds one. The reason for this is that an excessive weight assigned to the auxiliary task could impair the model's ability to produce fluent responses. These observations prove that the assignment of different tasks' weights is a critical operation for multi-task joint training that directly affects the final performance of the model.

4.5.6 | Error Analysis

We qualitatively compare the performance of the proposed model with other baseline models using examples. A generated example is given in Table 9. This example is randomly selected from 50 dialogue examples that are randomly sampled from the test set. Comparing the generated response by the gold sentence, it has been found that most of the models could perfectly copy the entity the_5th from the context. Nevertheless, the Seq2Seq+Atten model producing a duplicate sentence, even though it was natural and without grammatical errors. The GLMP model failed to copy the correct KB entry 6*pm* from the KB results, and so did the BoSsNet model. Likewise, the HMN model missed the entity *alex*, only our model carried out the completely accurate response. This demonstrates that our proposed hierarchical MN could filter out the unnecessary information and further filter out the unrelated KB entries.

In addition, from Fig.6, it can be observed that our generated response accurately answered the driver's question. Our generated response was *"there is heavy_traffic on the route to valero"*, which was a paraphrase of the gold response. Unfortunately, the gold response more natural and emotional than ours. There is no doubt that the gold expression is more frequently happened in our daily life. Also, it sounds more like a human-human conversation. Employing emotional information or personalized feature, e.g., user profiles, will be challenging and practical work. We will leave it as our future research direction.

5 | CONCLUSIONS

This work presents an end-to-end trainable model using hierarchical MN for a task-oriented dialogue system. The hierarchical memory network maintains an utterance-level memory network, a token-level memory network, and a KB memory network. Our model adopts the proposed hierarchical MN mechanism to focus on crucial information of dialogue history and KB, and employs a gated-mechanism to generate response word by word from vocabulary, dialogue history, or KB results during the decoding stage. The proposed model has achieved comparable performance on several open datasets. Moreover, ablation study and attention weights visualization have shown that our model can efficiently deal with dialogue history and filter out unnecessary information. Further analysis has demonstrated that the proposed model is more robust and efficient in dealing with long dialogues. This prove that the proposed framework is practical. Through the error analysis of dialogue examples produced by the proposed model, our future work will focus on filtering KB information to decrease the interference of useless information and how to enforce the emotional and personalized information into the conversation procedure.

ACKNOWLEDGMENTS

This research is supported by the Open Project Foundation of Henan key Laboratory for Big Data Processing & Analytics of Electronic Commerce, China[2022-KF-2].

Author contributions

This is an author contribution text.

Financial disclosure

None reported.

Conflict of interest

The authors declare no potential conflict of interests.

SUPPORTING INFORMATION

The following supporting information is available as part of the online article:

References

1. Zhang Zheng, Takanobu Ryuichi, Huang Minlie, Zhu Xiaoyan. Recent Advances and Challenges in Task-oriented Dialog System. *CoRR.* 2020;abs/2003.07490.

2. Yan Zhao, Duan Nan, Chen Peng, Zhou Ming, Zhou Jianshe, Li Zhoujun. Building Task-Oriented Dialogue Systems for Online Shopping. In: Singh Satinder P., Markovitch Shaul, eds. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, :4618–4626AAAI Press; 2017.

3. Wen Tsung-Hsien, Vandyke David, Mrksic Nikola, et al. A Network-based End-to-End Trainable Task-oriented Dialogue System. In: Lapata Mirella, Blunsom Phil, Koller Alexander, eds. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, :438–449Association for Computational Linguistics; 2017.

4. Zhao Tiancheng, Lu Allen, Lee Kyusong, Eskenazi Maxine. Generative Encoder-Decoder Models for Task-Oriented Spoken Dialog Systems with Chatting Capability. In: :27–36Association for Computational Linguistics; 2017; Saarbrücken, Germany.

5. Lei Wenqiang, Jin Xisen, Kan Min-Yen, Ren Zhaochun, He Xiangnan, Yin Dawei. Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-Sequence Architectures. In: :1437–1447Association for Computational Linguistics; 2018; Melbourne, Australia.

6. Eric Mihail, Manning Christopher. A Copy-Augmented Sequence-to-Sequence Architecture Gives Good Performance on Task-Oriented Dialogue. In: :468–473Association for Computational Linguistics; 2017; Valencia, Spain.

7. Budzianowski Paweł, Wen Tsung-Hsien, Tseng Bo-Hsiang, et al. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In: :5016–5026Association for Computational Linguistics; 2018; Brussels, Belgium.

8. Eric Mihail, Goel Rahul, Paul Shachi, et al. MultiWOZ 2.1: A Consolidated Multi-Domain Dialogue Dataset with State Corrections and State Tracking Baselines. In: :422–428European Language Resources Association; 2020; Marseille, France.

9. Rastogi Abhinav, Zang Xiaoxue, Sunkara Srinivas, Gupta Raghav, Khaitan Pranav. Towards Scalable Multi-Domain Conversational Agents: The Schema-Guided Dialogue Dataset. In: :8689–8696AAAI Press; 2020.

10. Chen Wenhu, Chen Jianshu, Qin Pengda, Yan Xifeng, Wang William Yang. Semantically Conditioned Dialog Response Generation via Hierarchical Disentangled Self-Attention. In: :3696–3709Association for Computational Linguistics; 2019; Florence, Italy.

11. Gao Silin, Zhang Yichi, Ou Zhijian, Yu Zhou. Paraphrase Augmented Task-Oriented Dialog Generation. In: :639–649Association for Computational Linguistics; 2020; Online.

12. Zhang Yichi, Ou Zhijian, Yu Zhou. Task-Oriented Dialog Systems That Consider Multiple Appropriate Responses under the Same Context. In: :9604–9611AAAI Press; 2020.

13. Radford Alec, Narasimhan Karthik, Salimans Tim, Sutskever Ilya. *Improving Language Understanding by Generative Pre-Training.* 2018.

14. Radford Alec, Wu Jeffrey, Child Rewon, Luan David, Amodei Dario, Sutskever Ilya. *Language Models are Unsupervised Multitask Learners.* 2019.

15. Brown Tom, Mann Benjamin, Ryder Nick, et al. Language Models are Few-Shot Learners. In: Larochelle H., Ranzato M., Hadsell R., Balcan M. F., Lin H., eds. *Advances in Neural Information Processing Systems*, :1877–1901Curran Associates, Inc.; 2020.

16. Ham Donghoon, Lee Jeong-Gwan, Jang Youngsoo, Kim Kee-Eung. End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems using GPT-2. In: :583–592Association for Computational Linguistics; 2020; Online.

17. Peng Baolin, Zhu Chenguang, Li Chunyuan, et al. Few-shot Natural Language Generation for Task-Oriented Dialog. In: :172–182Association for Computational Linguistics; 2020; Online.

18. Hosseini-Asl Ehsan, McCann Bryan, Wu Chien-Sheng, Yavuz Semih, Socher Richard. A Simple Language Model for Task-Oriented Dialogue. In: Larochelle H., Ranzato M., Hadsell R., Balcan M. F., Lin H., eds. *Advances in Neural Information Processing Systems*, :20179–20191Curran Associates, Inc.; 2020.

19. Yang Yunyi, Li Yunhao, Quan Xiaojun. UBAR: Towards Fully End-to-End Task-Oriented Dialog System with GPT-2. In: :14230–14238AAAI Press; 2021.

20. Bordes Antoine, Boureau Y-Lan, Weston Jason. Learning End-to-End Goal-Oriented Dialog. In: OpenReview.net; 2017.

21. Weston Jason, Chopra Sumit, Bordes Antoine. Memory Networks. In: Bengio Yoshua, LeCun Yann, eds. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, ; 2015.

22. Wu Chien-Sheng, Madotto Andrea, Winata Genta Indra, Fung Pascale. End-to-End Dynamic Query Memory Network for Entity-Value Independent Task-Oriented Dialog. In: :6154-6158; 2018.

23. Madotto Andrea, Wu Chien-Sheng, Fung Pascale. Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems. In: :1468–1478Association for Computational Linguistics; 2018; Melbourne, Australia.

24. Lin Zehao, Huang Xinjing, Ji Feng, Chen Haiqing, Zhang Yin. Task-Oriented Conversation Generation Using Heterogeneous Memory Networks. In: :4558–4567Association for Computational Linguistics; 2019; Hong Kong, China.

25. Chen Xiuyi, Xu Jiaming, Xu Bo. A Working Memory Model for Task-oriented Dialog Response Generation. In: :2687–2693Association for Computational Linguistics; 2019; Florence, Italy.

26. Wu Chien-Sheng, Socher Richard, Xiong Caiming. Global-to-local Memory Pointer Networks for Task-Oriented Dialogue. In: OpenReview.net; 2019.

27. Eric Mihail, Krishnan Lakshmi, Charette Francois, Manning Christopher D.. Key-Value Retrieval Networks for Task-Oriented Dialogue. In: :37–49Association for Computational Linguistics; 2017; Saarbrücken, Germany.

28. Gangi Reddy Revanth, Contractor Danish, Raghu Dinesh, Joshi Sachindra. Multi-Level Memory for Task Oriented Dialogs. In: :3744–3754Association for Computational Linguistics; 2019; Minneapolis, Minnesota.

29. Wang Jian, Liu Junhao, Bi Wei, et al. Dual Dynamic Memory Network for End-to-End Multi-turn Task-oriented Dialog Systems. In: Scott Donia, Bel Núria, Zong Chengqing, eds. *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, :4100–4110International Committee on Computational Linguistics; 2020.

30. Raghu Dinesh, Gupta Nikhil, Mausam . Disentangling Language and Knowledge in Task-Oriented Dialogs. In: :1239–1255Association for Computational Linguistics; 2019; Minneapolis, Minnesota.

31. Banerjee Suman, Khapra Mitesh M.. Graph Convolutional Network with Sequential Attention for Goal-Oriented Dialogue Systems. *Trans. Assoc. Comput. Linguistics.* 2019;7:485–500.

32. He Zhenhao, He Yuhong, Wu Qingyao, Chen Jian. Fg2seq: Effectively Encoding Knowledge for End-To-End Task-Oriented Dialog. In: :8029–8033IEEE; 2020.

33. Schlichtkrull Michael Sejr, Kipf Thomas N., Bloem Peter, Berg Rianne, Titov Ivan, Welling Max. Modeling Relational Data with Graph Convolutional Networks. In: Gangemi Aldo, Navigli Roberto, Vidal Maria-Esther, et al. , eds. *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, Lecture Notes in Computer Science, vol. 10843: :593–607Springer; 2018.

34. Balakrishnan Anusha, Rao Jinfeng, Upasani Kartikeya, White Michael, Subba Rajen. Constrained Decoding for Neural NLG from Compositional Representations in Task-Oriented Dialogue. In: Korhonen Anna, Traum David R., Màrquez Lluís, eds. *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, :831–844Association for Computational Linguistics; 2019.

35. Henderson Matthew, Vulic Ivan, Gerz Daniela, et al. Training Neural Response Selection for Task-Oriented Dialogue Systems. In: Korhonen Anna, Traum David R., Màrquez Lluís, eds. *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, :5392–5404Association for Computational Linguistics; 2019.

36. Shalyminov Igor, Lee Sungjin, Eshghi Arash, Lemon Oliver. Data-Efficient Goal-Oriented Conversation with Dialogue Knowledge Transfer Networks. In: Inui Kentaro, Jiang Jing, Ng Vincent, Wan Xiaojun, eds. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, :1741–1751Association for Computational Linguistics; 2019.

37. Qin Libo, Xu Xiao, Che Wanxiang, Zhang Yue, Liu Ting. Dynamic Fusion Network for Multi-Domain End-to-end Task-Oriented Dialog. In: :6344–6354Association for Computational Linguistics; 2020; Online.

38. He Wanwei, Yang Min, Yan Rui, Li Chengming, Shen Ying, Xu Ruifeng. Amalgamating Knowledge from Two Teachers for Task-oriented Dialogue System with Adversarial Training. In: Webber Bonnie, Cohn Trevor, He Yulan, Liu Yang, eds. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, :3498–3507Association for Computational Linguistics; 2020.

39. Liu Bing, Tür Gökhan, Hakkani-Tür Dilek, Shah Pararth, Heck Larry P.. Dialogue Learning with Human Teaching and Feedback in End-to-End Trainable Task-Oriented Dialogue Systems. In: Walker Marilyn A., Ji Heng, Stent Amanda, eds. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, :2060–2069Association for Computational Linguistics; 2018.

40. Wang Weikang, Zhang Jiajun, Li Qian, Hwang Mei-Yuh, Zong Chengqing, Li Zhifei. Incremental Learning from Scratch for Task-Oriented Dialogue Systems. In: Korhonen Anna, Traum David R., Màrquez Lluís, eds. *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, :3710–3720Association for Computational Linguistics; 2019.

41. Pennington Jeffrey, Socher Richard, Manning Christopher D.. Glove: Global Vectors for Word Representation. In: Moschitti Alessandro, Pang Bo, Daelemans Walter, eds. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, :1532–1543ACL; 2014.

42. Sukhbaatar Sainbayar, szlam arthur, Weston Jason, Fergus Rob. End-To-End Memory Networks. In: Cortes C., Lawrence N., Lee D., Sugiyama M., Garnett R., eds. *Advances in Neural Information Processing Systems*, Curran Associates, Inc.; 2015.

43. Chung Junyoung, Gülçehre Çaglar, Cho KyungHyun, Bengio Yoshua. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR*. 2014;abs/1412.3555.

44. Henderson Matthew, Thomson Blaise, Williams Jason D.. The Second Dialog State Tracking Challenge. In: :263–272The Association for Computer Linguistics; 2014.

45. Wen Tsung-Hsien, Miao Yishu, Blunsom Phil, Young Steve J.. Latent Intention Dialogue Models. In: Precup Doina, Teh Yee Whye, eds. *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, Proceedings of Machine Learning Research, vol. 70: :3732–3741PMLR; 2017.

46. Kingma Diederik P., Ba Jimmy. Adam: A Method for Stochastic Optimization. In: Bengio Yoshua, LeCun Yann, eds. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, ; 2015.

47. Papineni Kishore, Roukos Salim, Ward Todd, Zhu Wei-Jing. Bleu: a Method for Automatic Evaluation of Machine Translation. In: :311–318Association for Computational Linguistics; 2002; Philadelphia, Pennsylvania, USA.

48. Li Jiwei, Galley Michel, Brockett Chris, Gao Jianfeng, Dolan Bill. A Diversity-Promoting Objective Function for Neural Conversation Models. In: :110–119Association for Computational Linguistics; 2016; San Diego, California.

49. Sharma Shikhar, Asri Layla El, Schulz Hannes, Zumer Jeremie. Relevance of Unsupervised Metrics in Task-Oriented Dialogue for Evaluating Natural Language Generation. *CoRR.* 2017;abs/1706.09799.

50. Luong Thang, Pham Hieu, Manning Christopher D.. Effective Approaches to Attention-based Neural Machine Translation. In: :1412–1421Association for Computational Linguistics; 2015; Lisbon, Portugal.

51. Wang Dingmin, Chen Ziyao, He Wanwei, Zhong Li, Tao Yunzhe, Yang Min. A Template-guided Hybrid Pointer Network for Knowledge-based Task-oriented Dialogue Systems. In: :18–28Association for Computational Linguistics; 2021; Online.

52. Vaswani Ashish, Shazeer Noam, Parmar Niki, et al. Attention is All you Need. In: Guyon Isabelle, Luxburg Ulrike, Bengio Samy, et al. , eds. *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, :5998–6008; 2017.

53. Ganhotra Jatin, Moore Robert, Joshi Sachindra, Wadhawan Kahini. Effects of Naturalistic Variation in Goal-Oriented Dialog. In: Cohn Trevor, He Yulan, Liu Yang, eds. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, Findings of ACL, vol. EMNLP 2020: :4013–4020Association for Computational Linguistics; 2020.

54. Moore Robert J., Arar Raphael. *Conversational UX Design: A Practitioner's Guide to the Natural Conversation Framework*. ACM; 2019.

55. Bommasani Rishi, Hudson Drew A., Adeli Ehsan, et al. On the Opportunities and Risks of Foundation Models. *CoRR.* 2021;abs/2108.07258.

How to cite this article: Williams K., B. Hoskins, R. Lee, G. Masato, and T. Woollings (2016), A regime analysis of Atlantic winter jet variability applied to evaluate HadGEM3-GC2, *Q.J.R. Meteorol. Soc.*, *2017;00:1–6*.