A Modular Task-oriented Dialogue System Using a Neural Mixture-of-Experts

Pei, J.; Ren, P.; de Rijke, M.

Publication date
2019

Document Version
Submitted manuscript

Published in
1st Workshop on Conversational Interaction Systems (WCIS)

A Modular Task-oriented Dialogue System Using a Neural Mixture-of-Experts

Jiahuan Pei
University of Amsterdam
Amsterdam, The Netherlands
j.pei@uva.nl

Pengjie Ren
University of Amsterdam
Amsterdam, The Netherlands
p.ren@uva.nl

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

End-to-end Task-oriented Dialogue Systems (TDSs) have attracted a lot of attention for their superiority (e.g., in terms of global optimization) over pipeline modularized TDSs. Previous studies on end-to-end TDSs use a single-module model to generate responses for complex dialogue contexts. However, no model consistently outperforms the others in all cases.

We propose a neural Modular Task-oriented Dialogue System (MTDS) framework, in which a few expert bots are combined to generate the response for a given dialogue context. MTDS consists of a chair bot and several expert bots. Each expert bot is specialized for a particular situation, e.g., one domain, one type of action of a system, etc. The chair bot coordinates multiple expert bots and adaptively selects an expert bot to generate the appropriate response. We further propose a Token-level Mixture-of-Expert (TokenMoE) model to implement MTDS, where the expert bots predict multiple tokens at each timestamp and the chair bot determines the final generated token by fully taking into consideration the outputs of all expert bots. Both the chair bot and the expert bots are jointly trained in an end-to-end fashion.

To verify the effectiveness of TokenMoE, we carry out extensive experiments on a benchmark dataset. Compared with the baseline using a single-module model, our TokenMoE improves the performance by 8.1% of *inform rate* and 0.8% of *success rate*.

CCS CONCEPTS

• **Computing methodologies** → **Discourse, dialogue and pragmatics**; • **Information systems** → *Search interfaces.*

KEYWORDS

Task-oriented dialogue systems, Mixture of experts, Neural networks

ACM Reference Format:
Jiahuan Pei, Pengjie Ren, and Maarten de Rijke. 2019. A Modular Task-oriented Dialogue System Using a Neural Mixture-of-Experts. In *Proceedings of WCIS '19: 1st Workshop on Conversational Interaction Systems (WCIS '19).* ACM, New York, NY, USA, 7 pages. https://doi.org/xx.xxx/xxx_x

WCIS '19, July 25, 2019, Paris, France

ACM ISBN 978-1-4503-9999-9/18/06...$15.00
https://doi.org/xx.xxx/xxx_x

1 INTRODUCTION

As an important branch of spoken dialogue systems, Task-oriented Dialogue Systems (TDSs) have raised considerable interest due to their broad applicability, e.g., for booking flight tickets or scheduling meetings [30, 33]. Unlike open-ended dialogue systems [23], TDSs aim to assist users to achieve specific goals.

Existing TDS methods can be divided into two broad categories: modularized pipeline TDSs [3, 6, 33] and end-to-end single-module TDSs [12, 28]. The former decomposes the task-oriented dialogue task into modularized pipelines that are addressed by separate models while the latter proposes to use an end-to-end model to solve the task. End-to-end single-module TDSs have many attractive characteristics, e.g., global optimization and easier adaptation to new domains [6]. However, existing studies on end-to-end single-module TDSs mostly generates a response token by token, where each token is drawn from only one distribution over output vocabulary. We think this is unreasonable because the distribution differs a lot among different intents. Actually, more and more empirical studies from different machine learning applications suggest that no model consistently outperforms all others in all cases [9, 19].

Inspired by this intuition, we propose a new *Modular Task-oriented Dialogue System* (MTDS) framework, as shown in Fig. 1. MTDS consists of a *chair bot* and several *expert bots*. Each expert

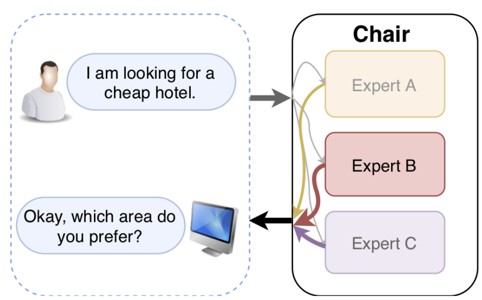

Figure 1: *Modular Task-oriented Dialogue System* (MTDS) framework.

bot is specialized for a particular situation, e.g., one domain, one type of action of a system, etc. The chair bot coordinates multiple expert bots and adaptively selects an expert bot to generate the final response. Compared with existing end-to-end single-module TDSs, the advantages of MTDSs are two-fold. First, the specialization of different expert bots and the use of a dynamic chair bot for combining the outputs breaks the bottleneck of a single model. Second, it is more easily *traceable*: we can analyze who is to blame when the model makes a mistake. Under this framework, we further propose a neural *Mixture-of-Expert* (MOE) model, namely *Token-level Mixture-of-Expert* (TokenMoE), where the expert bots predict

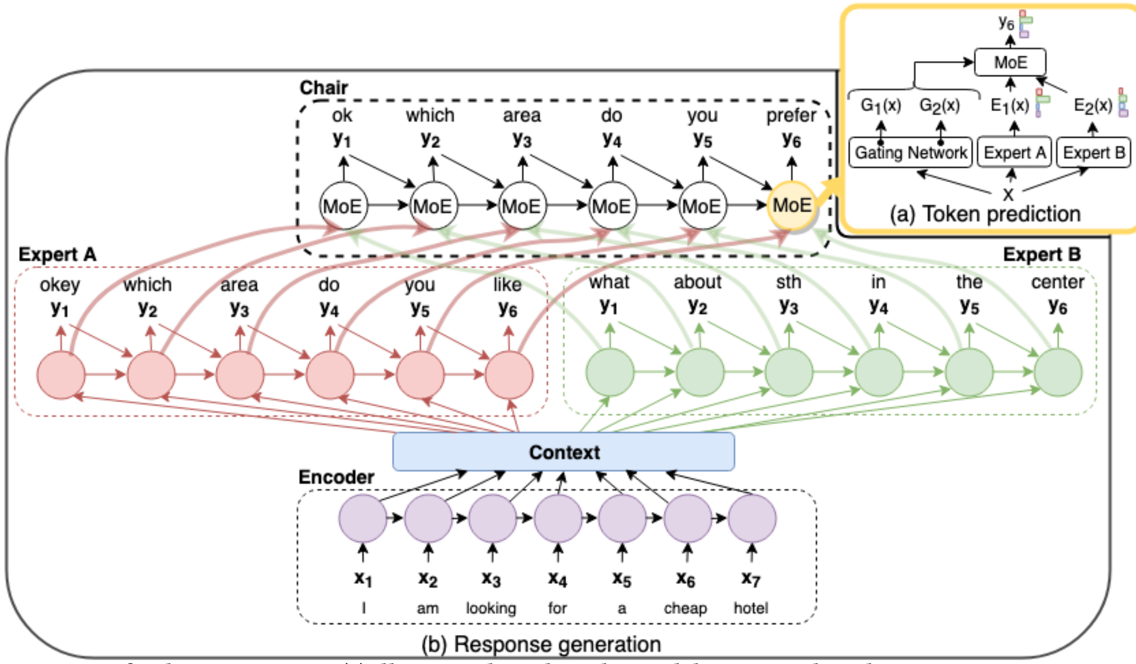

Figure 2: Overview of TokenMoE. Figure (a) illustrates how does the model generate the token y_6 given sequence X as an input. Figure (b) shows how does the model generate the whole sequence Y as a dialogue response.

multiple tokens at each timestamp and the chair bot determines the final generated token by taking into account the outputs of all expert bots. We devise a global-and-local learning scheme to train TokenMoE. We design a *localized expert loss* to force each expert to specialize on a particular task. We also design a *global chair loss* to differentiate the loss incurred from different experts [13].

To verify the effectiveness of TokenMoE, we carry out extensive experiments on the MultiWOZ benchmark dataset. The results show that, compared with the baseline using a single-module model, our TokenMoE improves the performance by 8.1% of *inform rate*, and 0.8% of *success rate*.

The contributions of this paper can be summarized as follows.

- We propose the MTDS framework, which breaks the bottleneck of a single-module model and provides better traceability of mistakes.
- We present the TokenMoE model to implement MTDS at the token-level.
- We devise a global-and-local learning scheme to effectively train TokenMoE.

2 METHODOLOGY

Let $\mathcal{D} = \{(X_p, Y_p)\}_{p=1}^{|\mathcal{D}|}$ denote a dataset with $|\mathcal{D}|$ independent random samples of (X, Y), where $X = (x_1, \dots, x_m)$ is a sequence of dialogue context with m words and $Y = (y_1, \dots, y_n)$ is a sequence of system response with n words. The model aims to optimize the generation probability of Y conditioned on X, i.e., $p(Y|X)$.

2.1 MTDS framework

We propose the MTDS framework, which consists of two types of modules as shown in Figure 1:

- k **expert** bots, each of which is specialized for a particular situation, namely *intent* (e.g., one domain, one type of action of a system, etc.). Those intents partition dataset $\mathcal{D}$ into k pieces $\mathcal{S} = \{\mathcal{S}_l\}_{l=1}^{k}$, where $\mathcal{S}_l \triangleq \{(X^l, Y^l)\}$. Each expert is trained to predict $\mathbf{p}^l(Y^l|X^l)$. We expect the l-th expert generally performs better than the others on $\mathcal{S}_l$.
- a **chair** bot, which learns to coordinate a group of expert bots to make an optimal decision. The chair bot is trained to predict $\mathbf{p}(Y|X)$, where (X, Y) is any sample pair from $\mathcal{D}$.

2.2 TokenMoE model

In this section, we introduce TokenMoE, a token-level implementation of the MTDS framework. As shown in Figure 2, TokenMoE consists of three types of components, i.e., a shared encoder, k expert decoders, and a chair decoder.

Shared context encoder. The role of shared context encoder is to read the dialogue context sequence and construct their representations at each timestamp. Here we follow Budzianowski et al. [5] and employ a Long Short-Term Memory (LSTM) [14] to map the input sequence X to hidden vectors $\{\mathbf{h}_1, \dots, \mathbf{h}_m\}$. The hidden vector $\mathbf{h}_i$ at timestep i-th can be represented as:

$$\mathbf{h}_i, \mathbf{s}_i = \text{LSTM}(emb(x_i), \mathbf{h}_{i-1}, \mathbf{s}_{i-1}), \tag{1}$$

where $emb(x_i)$ is the embedding of the token x_i at i. The initial state of LSTM $\mathbf{s}_0$ is set to 0.

Expert decoder. The l-th expert outputs the probability $\mathbf{p}_j^l$ over the vocabulary set $\mathcal{V}$ at j-th step by:

$$\begin{aligned} \mathbf{p}_j^l &= \text{softmax}(\mathbf{U}^T \mathbf{o}_j^l + \mathbf{a}), \\ \mathbf{o}_j^l, \mathbf{s}_j^l &= \text{LSTM}(y_{j-1}^l \oplus \mathbf{c}_j^l, \mathbf{o}_{j-1}^l, \mathbf{s}_{j-1}^l), \end{aligned} \tag{2}$$

where $\mathbf{U}, \mathbf{a}$ are learnable matrices. $\mathbf{s}_j^l$ is the state vector which is initialized by the last state of the shared context encoder. y_{j-1}^l

is the generated token at timestamp $j-1$ by expert l. $\mathbf{c}_j^l$ is the context vector which is calculated with a concatenation attention mechanism [1, 18] over the hidden representations from shared context encoder.

$$\begin{aligned} \mathbf{c}_j^l &= \sum_{i=1}^{m} \alpha_{ji}^l \mathbf{h}_i, \\ \alpha_{ji}^l &= \frac{\exp(w_{ji}^l)}{\sum_{i=1}^{m} \exp(w_{ji}^l)}, \\ w_{ji}^l &= \mathbf{v}^T \tanh(\mathbf{W}^T(\mathbf{h}_i \oplus \mathbf{s}_{j-1}^l) + \mathbf{b}), \end{aligned} \tag{3}$$

where α is the attention weights; $\oplus$ is the concatenation operation. $\mathbf{W}$, $\mathbf{b}$, $\mathbf{v}$ are learnable parameters, which are not shared by different experts in our experiments.

Chair decoder. The chair decoder estimates the final token prediction distribution $\mathbf{p}_j$ by combining prediction distribution of all experts (including chair itself) with a proposed token-level Mixture-of-Expert (MOE) scheme. As shown in Fig. 2(b), following the typical neural MOE architecture [22, 24], $\mathbf{p}_j$ is computed based on the state $\mathbf{s}_j^l$ and token prediction distribution $\mathbf{p}_j^l$ from all experts (including chair itself) at j as follows.

$$\mathbf{p}_j = \sum_{l=1}^{k+1} \beta_j^l \cdot \mathbf{p}_j^l, \tag{4}$$

where $\mathbf{p}_j^{k+1}$ is the prediction of the chair, which employs the same architecture of the other experts but is trained on all data. β_j^l is the normalized importance scores that can be computed as:

$$\begin{aligned} \beta_j^l &= \frac{\exp(u_l^T u_{e,l})}{\sum_{b=1}^{k} \exp(u_b^T u_{e,l})}, \\ u_l &= \mathrm{MLP}(\mathbf{h}), \\ \mathbf{h} &= \mathbf{s}_j^1 \oplus \mathbf{p}_j^1 \oplus \cdots \oplus \mathbf{s}_j^k \oplus \mathbf{p}_j^k \oplus \mathbf{s}_j^{k+1} \oplus \mathbf{p}_j^{k+1}, \end{aligned} \tag{5}$$

where $u_{e,l}$ is an expert-specific, learnable vector that reflects which dimension of the projected hidden representation is highlighted for the expert.

Loss function. We devise a global-and-local learning scheme to train TokenMoE. Each expert l is optimized by a localized expert loss defined on $\mathcal{S}_l$, which forces each expert to specialize on one of the portions of data $\mathcal{S}_l$. We use cross-entropy loss for each expert and the joint loss for all experts are as follows.

$$\mathcal{L}_{experts} = \sum_{l=1}^{k+1} \sum_{(X^l, Y^l) \in \mathcal{S}_l} \sum_{j=1}^{n} \mu_k y_j^l \log \mathbf{p}_j^l, \tag{6}$$

where p_j^l is the token prediction by expert l (Eq. 2) computed on the r-th data sample; $y_j^{l,r}$ is a one-hot vector indicating the ground truth token at j; and μ_k is the weight of the k-th expert.

We also design the global chair loss to differentiate the loss incurred from different experts. The chair can attribute the source of errors to the expert in charge. For each data sample in $\mathcal{D}$, we follow the MOE architecture and calculate the combined taken prediction $\mathbf{p}_j$ (Eq. 4). Then the total loss incurred by MOE can be denoted as follows.

$$\mathcal{L}_{chair} = \sum_{r=1}^{|\mathcal{D}|} \sum_{j=1}^{n} y_j \log \mathbf{p}_j. \tag{7}$$

Our overall optimization follows the joint learning paradigm that is defined as a weighted combination of constituent loss.

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{experts} + (1-\lambda) \cdot \mathcal{L}_{chair}, \tag{8}$$

where λ is a hyper-parameter.

Table 1: Different settings of learning schemes.

Variants	$\mathbf{p}_j$ (Eq. 2)	μ_k (Eq. 6)	λ (Eq. 8)
S1	MOE	learnable	learnable
S2	MOE	–	0.0
S3	w/o MOE	$\frac{1}{k}$	0.5
S4	MOE	$\frac{1}{k}$	0.5

3 EXPERIMENTAL SETUP

3.1 Research questions

We seek to answer the following research questions.

(**RQ1**) Is there a single model that consistently outperforms the others on all domains? The point of this question is to verify the motivation behind MTDS and TokenMoE.

(**RQ2**) Does the TokenMoE model outperform the state-of-the-art end-to-end single-module TDS model? The point of this question is to determine the effectiveness of the proposed TokenMoE model.

(**RQ3**) How do the proposed token-level MOE scheme (Eq. 4 and Eq. 5) and the global-and-local learning scheme (Eq. 6 and Eq. 8) in the TokenMoE model affect the final performance? The point of this question is to do an ablation study on effective learning schemes.

3.2 Comparison methods

We use the dominant Sequence-to-Sequence (Seq2Seq) model in an encoder-decoder architecture [6] and reproduce the state-of-the-art single model baseline, namely Sequence-to-Sequence with Attention Using LSTM (S2SAttnLSTM) [4, 5], based on the source code provided by the authors.[1]

To answer **RQ1**, we investigate the performance of the following variants of S2SAttnLSTM on different domains.

- **V1.** This variant excludes the attention mechanism from the baseline model and keeps the other settings unchanged.
- **V2.** This variant changes the LSTM cell as GRU and keeps the other settings the same.
- **V3.** This variant reduces the number of hidden units to 100 and maintains the other settings.

To answer **RQ2**, we train TokenMoE based on the benchmark dataset and test how it performs compared to the single-module baseline.

[1]https://github.com/budzianowski/multiwoz. For fair comparison, we remove validation set from training set and report the reproduced results.

Table 2: Performance of the single-module baseline (S2SAttnLSTM) and its three variations (V1, V2 and V3) on different domains. Bold highlighted results indicate a statistically significant improvement of a metric over the strongest baseline on the same domain (paired t-test, $p < 0.01$). UNK denotes a unknown domain excluding the domains described in §3.4. Please note that the number of the evaluated dialogue turns varies among different domains.

	Inform (%)				Success (%)				BLEU (%)				Score				# of turns
	Baseline	/V1	/V2	/V3	Baseline	/V1	/V2	/V3	Baseline	/V1	/V2	/V3	Baseline	/V1	/V2	/V3	
Attraction	87.20	86.20	**91.80**	88.70	81.30	74.80	**83.70**	83.70	15.14	14.95	**16.08**	14.86	99.39	95.45	**103.83**	101.06	1042
Hotel	89.90	**93.90**	89.90	90.30	87.50	**91.70**	87.40	89.10	**16.60**	15.60	15.11	14.13	105.30	**108.40**	103.76	103.83	1068
Restaurant	89.20	**91.70**	86.40	86.10	85.80	**87.80**	84.00	83.40	17.07	**17.70**	16.07	17.34	104.57	**107.45**	101.27	102.09	1024
Taxi	100.00	100.00	100.00	100.00	**99.90**	99.80	**99.90**	99.80	17.33	19.18	**20.13**	18.32	117.28	119.08	**120.08**	118.22	395
Train	77.70	77.70	79.00	**81.60**	75.60	74.80	77.20	**79.60**	20.35	15.64	**22.81**	20.62	97.00	91.89	100.91	**101.22**	1702
Booking	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	22.05	21.61	21.96	**22.06**	122.05	121.61	121.96	**122.06**	1407
General	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	20.21	19.53	20.13	**20.80**	120.21	119.53	120.13	**120.80**	2596
UNK	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	12.40	11.75	**13.12**	11.80	112.40	111.75	**113.12**	111.80	81

To answer **RQ3**, we explore different settings of the learning schemes by considering alternative choices of $\mathbf{p}_j$ in Eq. 4, μ_k in Eq. 6 and λ in Eq. 8. We summarize different variants in Table 1.

In this work, we are focusing on context-to-act task [5], so natural language generation (NLG) baselines (e.g., SC-LSTM [27]) will not be taken into consideration.

3.3 Implementation details

The vocabulary size is the same as in the original paper that releases the dataset [5], which has 400 tokens. Out-of-vocabulary words are replaced with "<UNK>". We set the word embedding size to 50 and all LSTM hidden state sizes to 150. We use Adam [15] as our optimization algorithm with hyperparameters $\alpha = 0.005$, $\beta 1 = 0.9$, $\beta 2 = 0.999$ and $\epsilon = 10^{-8}$. We also apply gradient clipping [20] with range $[-5, 5]$ during training. We use $l2$ regularization to alleviate overfitting, the weight of which is set to 0.00001. We set mini-batch size to 64. We use greedy search to generate the response during testing. Extra techniques (e.g., beam search) are not incorporated, because our main concern is the modular model outperforms single-module model instead of the effectiveness of these popular techniques.

3.4 Dataset

Our experiments are conducted on the Multi-Domain Wizard-of-Oz (MultiWOZ) [5] dataset. This is the latest large-scale human-to-human TDS dataset with rich semantic labels (e.g., domains and dialogue actions) and benchmark results of response generation.[2] MultiWOZ consists of ~10k natural conversations between a tourist and a clerk. We consider 6 specific action-related domains (i.e., *Attraction*, *Hotel*, *Restaurant*, *Taxi*, *Train*, and *Booking*) and 1 universal domain (i.e., *General*). 67.37% of dialogues are cross-domain which covers 2–5 domains on average. The average number of turns per dialogue is 13.68 and a turn contains 13.18 tokens on average. To facilitate reproducibility of the results, the dataset is randomly split into into 8,438/1,000/1,000 dialogues for training, validation, and testing, respectively.

3.5 Evaluation metrics

We use three commonly used evaluation metrics [5]:

- *Inform.* The fraction of responses that provide a correct entity out of all responses.
- *Success.* The fraction of responses that answer all the requested attributes out of all responses.
- *BLEU.* This is a score for comparing a generated response to one or more reference responses.

Following Budzianowski et al. [4], we use *Score* = 0.5**Inform*+ 0.5**Success*+*BLEU* as the selection criterion to choose the best model on the validation set and report the performance of the model on the test set. We utilize a paired t-test to show statistical significance ($p < 0.01$) of relative improvements.

4 RESULTS

This section describes the results of our experiments and answers research questions proposed in §3.

4.1 Performance of single-module TDSs on different domains (RQ1)

To answer **RQ1**, we assess the performance of the single-module baseline S2SAttnLSTM and its three variants with settings (V1, V2, and V3) described in §3.2 on different domains. The results are shown in Table 2.

We can see that none of those models can consistently outperform the others on all domains and all metrics. That is to say, a model can achieve its best performance only in some particular situations. To be specific, S2SAttnLSTM achieves its best performance only on the *Hotel* domain in terms of *BLEU* and *Taxi* domain in terms of *Success*. S2SAttnLSTM/V1 outperforms all other models on the *Restaurant* domain on all metrics and on the *Hotel* domain (except for *BLEU*). S2SAttnLSTM/V2 beats the others on the *Attraction* and *Taxi* domains in terms of all metrics. S2SAttnLSTM/V3 performs best on the *Booking* and *General* domains in terms of *BLEU* and *Score*. Overall, S2SAttnLSTM/V1 specializes in, and leads on, the *Hotel* and *Restaurant* domains. S2SAttnLSTM/V2 acts as an expert bot specialized for the *Attraction*, *Taxi*, *UNK* domains, and S2SAttnLSTM/V2 serves as an expert bot for the *Train*, *Booking*, *General* domains. Generally, the experimental results verify the assumption and motivation of our MTDS framework.

4.2 Overall performance (RQ2)

To answer our main research question, **RQ2**, we evaluate the performance of TokenMoE and the baselines (S2SAttnLSTM, TokenMoE

[2] http://dialogue.mi.eng.cam.ac.uk/index.php/corpus/

and their variants with settings V1, V2, V3). The results are shown in Table 3.

Table 3: Comparison between TokenMoE, the benchmark baseline S2SAttnLSTM, and their variant models using setting V1, V2, V3, respectively. Bold results indicate a statistically significant improvement over the strongest baseline (paired t-test, $p < 0.01$).

	Inform (%)	Success (%)	BLEU (%)	Score
S2SAttnLSTM [5]	67.20	57.20	17.83	80.03
S2SAttnLSTM/V1	63.60	52.20	18.10	76.00
S2SAttnLSTM/V2	67.20	58.90	**20.85**	83.90
S2SAttnLSTM/V3	68.60	59.30	19.41	83.36
TokenMoE/V1	64.00	52.50	18.95	77.20
TokenMoE/V2	62.60	54.30	18.90	77.35
TokenMoE/V3	62.90	54.00	18.34	76.79
TokenMoE	**75.30**	**59.70**	16.81	**84.31**

First, TokenMoE outperforms all baseline models by a large margin in terms of all metrics. Especially, TokenMoE significantly outperforms the benchmark single-module baseline S2SAttnLSTM, by 8.1% of *Inform* and 2.5% of *Sucecess*, which maintains the same settings as the original paper [5]. This shows that TokenMoE has an advantage of task completion by providing more appropriate entities and answering the requested attributes as many as possible.

Second, TokenMoE greatly outperforms TokenMoE/V1 by 11.7% on *Inform* and 7.5% on *Success*. This is true with S2SAttnLSTM and S2SAttnLSTM/V1 except that the improvements are smaller, i.e., 3.6% on *Inform* and 5.0% on *Sucecess*. On the one hand, this means that the attention mechanism is effective. On the other hand, this also shows that the attention mechanisms under our TokenMoE can be more effective and have an even more important role to play. That is to say, the MTDS framework has more potential to improve by separating the modeling of expert and chair bots. TokenMoE/V2 is inferior to S2SAttnLSTM/V2 when changing the LSTM cell as GRU. Similarly, TokenMoE/V3 is less effective than S2SAttnLSTM/V3 when decreasing the number of hidden units. This indicates that TokenMoE is more sensitive to the number of parameters, which we think is due to the fact that TokenMoE has a hard time learning the expert bots and their coordination with a small parameter space.

Third, all models achieve about 10% higher values in terms of *Inform* than in terms of *Success*. This shows that the big challenge of the dialogue generation task is how to answer requested attributes in a real-time manner. The *BLEU* scores of all models are quite low compared with the state-of-the-art result (45.6%) of machine translation [11] but are similar to the state-of-the-art result (18.9%) for dialogue generation [5]. This supports prior claims that the *BLEU* score is not an ideal measurement for dialogue generation and explains the reason why we use *Score* to choose our best model. Table 4 shows an example of the baseline S2SAttnLSTM and TokenMoE output, which indicates that a lower *BLEU* still does mean more appropriate response with more detail information.

4.3 Exploration of learning schemes for TokenMoE (RQ3)

To address **RQ3**, we explore how token-level MOE and learning schemes used in TokenMoE affect the performance. In Table 5, we report the results of four variants of TokenMoE (see Table 1). The detailed settings of each variant are as follows:

Table 4: An example of the generated responses of S2SAttnLSTM and TokenMoE. A user would prefer to get detail information of the train before booking a ticket.

Model	Response
S2SAttnLSTM	i have [value_count] trains that match your criteria . would you like me to book it for you ?
TokenMOE	i have train [train_id] that leaves at [value_time] and arrives at [value_time] . would you like me to book it ?

- S1 regards μ_k and λ as learnable parameters while the others regard them as hyperparameters.
- S2 uses token-level MOE but does not use the global-and-local learning scheme, i.e., $\lambda = 0$ in Eq. 8.
- S3 does not use token-level MOE and directly uses the prediction probability of the chair bot, i.e., $\mathbf{p}_j = \mathbf{p}_j^{k+1}$ in Eq. 4, which actually degenerates into S2SAttnLSTM with the proposed global-and-local learning.
- S4 uses both token-level MOE and global-and-local learning.

Table 5: Comparison of TokenMoE with different learning schemes (S1, S2, S3, S4) and the benchmark baseline S2SAttnLSTM. Bold results indicate a statistically significant improvement over the strongest baseline (paired t-test, $p < 0.01$).

	Inform (%)	Success (%)	BLEU	Score
S2SAttnLSTM/V2	67.20	58.90	**20.85**	83.90
TokenMoE/S1	66.20	54.90	19.11	79.66
TokenMoE/S2	66.50	56.90	19.48	81.18
TokenMoE/S3	70.60	**60.60**	18.67	84.27
TokenMoE/S4	**75.30**	59.70	16.81	**84.31**

First, S1 is worse than the other three variants on all metrics, which shows that it is not effective to learn μ_k and λ. The reason is that TokenMoE may fall into the *optimization trap* due to learning μ_k and λ. That is, TokenMoE learns a very small weight for the local loss of each expert (i.e., $\mu_k \approx 0$) and a large weight for the global loss of the chair bot (i.e., $\lambda \approx 1$). Afterwards, this loss will never decrease any more, so the model learns nothing useful.

Second, S2 is even worse than S2SAttnLSTM/V2 on all metrics which means the performance cannot be improved with the proposed token-level MOE alone. We believe the reason is that token-level MOE makes the model harder to learn, i.e., the model needs to learn not only each prediction distribution by the expert and chair bots but also their combinations. This can be verified by the fact that with token-level MOE and global-and-local learning, S4 further improves *Inform* by 4.7% compared with S3. Our explanation is that the global-and-local learning makes token-level MOE easier to learn by incorporating supervisions on both the prediction distribution of each expert (local loss in Eq. 6) and their combination (global loss in Eq. 8).

Third, S3 is better than S2SAttnLSTM/V2 in terms of *Inform* and *Success*. Also, it achieves the best performance on *Success*. This shows that TokenMoE with an appropriate scheme is expert in task accomplishment for a TDS, i.e., TokenMoE/S3 is able to generate more correct entities and answer more requested attributes. The reason behind this is quite clear: with global-and-local learning, each expert is trained to specialize on a particular domain, which means the chair and the experts are able to extract more manifold candidate tokens, each of them holds a unique preference distribution over the output vocabulary. For example, a *Booking* expert has a high probability to produce the intent-oriented token "booked" in the response "Your order has been booked". In contrast, without global-and-local learning, the single model prefers to generate more generic tokens (e.g., "thanks") that occur most frequently in all domains.

However, it is worth noting that both S2 and S3 are worse than S2SAttnLSTM/V2 on *BLEU* and S4 is even worse than S2 and S3. This indicates that token-level MOE and global-and-local learning have a negative influence on the response fluency evaluated by *BLEU*. A possible reason is that various candidate tokens from the chair and experts make the dialogue contexts more complex, which increases the difficulty of generating a fluent response. Another reason is that *BLEU* is not an ideal metric for dialogue generation task, as we discussed in §4.2.

5 RELATED WORK

There are two dominant frameworks for TDSs: modularized pipeline TDSs and end-to-end single-module TDSs.

5.1 Modularized pipeline TDSs

Modularized pipeline TDSs frameworks consists of a pipeline with several modules. Examples include Natural Language Understanding (NLU) [2, 7], Dialogue State Tracking (DST) [21, 34], Policy Learning (PL), and Natural Language Generation (NLG) [10, 32]. Each module has an explicitly decomposed function for a specialized subtask, which is beneficial to track errors. Young et al. [33] summarize typical pipeline TDSs that are constitutive of distinct modules following a POMDP paradigm. Crook et al. [8] develop a TDS platform that is loosely decomposed into three modules, i.e., initial processing of input, dialogue state updates, and policy execution. Yan et al. [31] present a TDS for completing various purchase-related tasks by optimizing individual upstream-dependent modules, i.e., query understanding, state tracking and dialogue management. However, the pipeline setting of these methods will unavoidably incur upstream propagation problem [6], module interdependence problem [6] and joint evaluation problem [33]. Unlike the methods listed above, our MTDS constists of a group of modules including a chair bot and several expert bots. This design addresses the module interdependence problem since each module is independent among the others. Besides, the chair bot alleviates the error propagation problem because it is able to manage the overall errors through an effective learning schemes.

5.2 End-to-end single-module TDSs

End-to-end single-module systems address the TDS task with only one module, which maps a *dialogue context* to a *response* directly [29]. There is a growing focus in research on end-to-end approaches for TDSs, which can enjoy global optimization and facilitate easier adaptation to new domains [6]. Sordoni et al. [25] show that using an Recurrent Neural Network (RNN) to generate text conditioned on the dialogue history results in more natural conversations. Later improvements have been made by adding an attention mechanism [17, 26], by modeling the hierarchical structure of dialogues [23], or by jointly learning belief spans [16]. However, existing studies on end-to-end TDSs mostly use a single-module underlying model to generate responses for complex dialogue contexts. This is practically problematic because dialogue contexts are very complicated with multiple sources of information [7]. In addition, previous studies show that it is abnormal to find a single model that achieves the best results on the overall task based on empirical studies from different machine learning applications [9, 19].

Different from the methods listed above, which use a single module to achieve TDSs, our MTDS uses multiple modules (expert and chair bots), which makes good use of the specialization of different experts and the generalization of chair for combining the final outputs. Besides, our MTDS model is able to track who is to blame when the model makes a mistake.

6 CONCLUSION AND FUTURE WORK

This paper we have presented a neural Modular Task-oriented Dialogue System (MTDS) framework composed of a chair bot and several expert bots. We have developed a TokenMoE model under this MTDS framework, where the expert bots make multiple token-level predictions at each timestamp and the chair bot predicts the final generated token by fully considering the whole outputs of all expert bots. Both the chair bot and the expert bots are jointly trained in an end-to-end fashion.

We have conducted extensive experiments on the benchmark dataset MultiWOZ and evaluated the performance in terms of four automatic metrics (i.e., *Inform*, *Success*, *BLEU*, and *Score*). We find that no general single-module TDS model can constantly outperform the others on all metrics. This empirical observation facilitates the design of a new framework, i.e., MTDS framework. We also verify the effectiveness of TokenMoE model compared with the baseline using a single-module model. Our TokenMoE outperforms the best single-module model (S2SAttnLSTM/V2) by 8.1% of *inform rate* and 0.8% of *success rate*. Besides, it significantly beats S2SAttnLSTM, the benchmark single-module baseline, by 3.5% of *Inform* and 4.2% of *Sucecess*. In addition, the experimental results show that learning scheme is an important factor of our TokenMoE model.

In the future work, we hope to explore Sentence-level Mixture-of-Expert (SentenceMoE) and combine it with the current TokenMoE to see whether the hybrid model will further improve the performance. Besides, we plan to try more fine-grained expert bots (e.g., according to user intents or system actions) and more datasets to test our new framework and model.

ACKNOWLEDGMENTS

This research was partially supported by Ahold Delhaize, the Association of Universities in the Netherlands (VSNU), the China Scholarship Council (CSC), and the Innovation Center for Artificial Intelligence (ICAI). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR '15)*. –.

[2] Ankur Bapna, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2017. Sequential dialogue context modeling for spoken language understanding. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL '17)*. 103–114.

[3] Antoine Bordes and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *International Conference on Learning Representations (ICLR '17)*. –.

[4] Pawel Budzianowski, Iñigo Casanueva, Bo-Hsiang Tseng, and Milica Gasic. 2018. *Towards end-to-end multi-domain dialogue modelling*. Technical Report. Cambridge University.

[5] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. MultiWOZ-A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP '18)*. 5016–5026.

[6] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *ACM SIGKDD Explorations Newsletter* 19, 2 (2017), 25–35.

[7] Po-Chun Chen, Ta-Chung Chi, Shang-Yu Su, and Yun-Nung Chen. 2017. Dynamic time-aware attention to speaker roles and contexts for spoken language understanding. In *Proceedings of 2017 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU '17)*. 554–560.

[8] Paul Crook, Alex Marin, Vipul Agarwal, Khushboo Aggarwal, Tasos Anastasakos, Ravi Bikkula, Daniel Boies, Asli Celikyilmaz, Senthilkumar Chandramohan, Zhaleh Feizollahi, et al. 2016. Task completion platform: A self-serve multi-domain goal oriented dialogue platform. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL '16)*. 47–51.

[9] Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems (MCS '00)*. 1–15.

[10] Ondrej Dušek and Filip Jurcıcek. 2016. A context-aware natural language generator for dialogue systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL '16)*. 185–190.

[11] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP '18)*. 489–500.

[12] Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL '17)*. 37–49.

[13] Jiang Guo, Darsh J Shah, and Regina Barzilay. 2018. Multi-source domain adaptation with mixture of experts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP '18)*. 4694–4703.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[15] Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR '15)*. –.

[16] Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL '18)*. 1437–1447.

[17] Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL '16)*. 994–1003.

[18] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP '15)*. 1412–1421.

[19] Saeed Masoudnia and Reza Ebrahimpour. 2014. Mixture of experts: a literature survey. *Artificial Intelligence Review* 42, 2 (2014), 275–293.

[20] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning (ICML '13)*. 1310–1318.

[21] Abhinav Rastogi, Raghav Gupta, and Dilek Hakkani-Tur. 2018. Multi-task learning for joint language understanding and dialogue state tracking. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL '19)*. 376–384.

[22] Patrick Schwab, Djordje Miladinovic, and Walter Karlen. 2019. Granger-causal attentive mixtures of experts: Learning important features with neural networks. In *AAAI Conference on Artificial Intelligence (AAAI '19)*. –.

[23] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence (AAAI '16)*. 3776–3784.

[24] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR '17)*. –.

[25] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT '15)*. 196–205.

[26] Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *ICML Deep Learning Workshop*. –.

[27] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP '15)*. 1711–1721.

[28] Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL '17)*. 438–449.

[29] Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL '17)*. 438–449.

[30] Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL '17)*. 665–677.

[31] Zhao Yan, Nan Duan, Peng Chen, Ming Zhou, Jianshe Zhou, and Zhoujun Li. 2017. Building task-oriented dialogue systems for online shopping. In *Thirty-First AAAI Conference on Artificial Intelligence (AAAI '2017)*. 4618–4626.

[32] Sanghyun Yi, Rahul Goel, Chandra Khatri, Tagyoung Chung, Behnam Hedayatnia, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tur. 2019. Towards coherent and engaging spoken dialog response generation using automatic conversation evaluators. *arXiv preprint arXiv:1904.13015* (2019).

[33] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proc. IEEE* 101, 5 (2013), 1160–1179.

[34] Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL '18)*. 1458–1467.