Gated Mechanism Enhanced Multi-Task Learning for Dialog Routing

Ziming Huang[1]*; **Zhuoxuan Jiang**[1]*; **Ke Wang**[2]*; **Juntao Li**[3], **Shanshan Feng**[4], **Xian-Ling Mao**[5]
[1]Tencent, [2]Alibaba DAMO Academy, [3]Soochow University,
[4]Harbin Institute of Technology(Shenzhen), [5]Beijing Institute of Technology
hzmyouxiang@gmail.com, jzhx@pku.edu.cn, wk258730@alibaba-inc.com,
ljt@suda.edu.cn, victor_fengss@foxmail.com, maoxl@bit.edu.cn

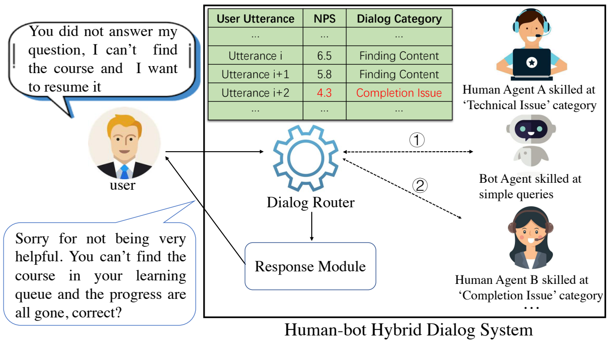
Abstract

Currently, human-bot symbiosis dialog systems, e.g., pre- and after-sales in E-commerce, are ubiquitous, and the dialog routing component is essential to improve the overall efficiency, reduce human resource cost, and enhance user experience. Although most existing methods can fulfil this requirement, they can only model single-source dialog data and cannot effectively capture the underlying knowledge of relations among data and subtasks. In this paper, we investigate this important problem by thoroughly mining both the data-to-task and task-to-task knowledge among various kinds of dialog data. To achieve the above targets, we propose a Gated Mechanism enhanced Multi-task Model (G3M), specifically including a novel dialog encoder and two tailored gated mechanism modules. The proposed method can play the role of hierarchical information filtering and is non-invasive to existing dialog systems. Based on two datasets collected from real world applications, extensive experimental results demonstrate the effectiveness of our method, which achieves the state-of-the-art performance by improving 8.7%/11.8% on RMSE metric and 2.2%/4.4% on F1 metric.

1 Introduction

Traditionally, a lot of human resource cost is spent on supporting the calling/online centers for customer care, such as pre-and after-sales for E-commerce and banking. With the rapid development of AI techniques for dialog systems, various bot agents have been deployed in those scenarios to reduce parts of human workload. Both the bot and human agents constitute a new and practical symbiosis dialog system which can keep a balance between service quality and human resource cost (Oraby et al., 2017).

In such a human-bot symbiosis environment, it is commonly seen that users' needs continuously change, or sometimes they are dissatisfactory with the current servicing agents. They would request the dialog system to replace the agents with more qualified ones, because usually each bot or human agent is trained for specific domains. For example, some agents are good at handling after-sales complaints, while some others would do well in providing technical guidance. To make the whole system operate efficiently, a dialog routing component is necessary to take charge of when to transfer the conversation and which domain of agents is suitable to provide service.



User Utterance	NPS	Dialog Category
...	...	...
Utterance i	6.5	Finding Content
Utterance i+1	5.8	Finding Content
Utterance i+2	4.3	Completion Issue
...	...	...

You did not answer my question, I can't find the course and I want to resume it
user
Sorry for not being very helpful. You can't find the course in your learning queue and the progress are all gone, correct?
Dialog Router
Response Module
①
②
Human Agent A skilled at 'Technical Issue' category
Bot Agent skilled at simple queries
Human Agent B skilled at 'Completion Issue' category
...
Human-bot Hybrid Dialog System

Figure 1: An example of dialog routing in human-bot symbiosis dialog system.

To illustrate the process of dialog routing task more clearly, Figure 1 shows an example. At the beginning, the user's query is operated by the bot agent through Flow ①, and the dialog router would continuously monitor the user's satisfactory degree by inferring Net Promoter Score (NPS) and current dialog category. When the inferred NPS (4.8) is lower than a hand-crafted threshold (5) at the ith dialog turn, the router chooses to switch the dialog to one of human agents skilled at the inferred category via Flow ②. Here human agent B is good at the category of 'Completion Issue' dialogs. From this picture, we could realize that how to implement the accurate matching between user's needs and skilled agents is greatly correlated with the whole system's efficiency and user experience.

Currently, most existing deployed systems are

*This work was done when they were in IBM Research China, and Zhuoxuan Jiang is the corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 3064–3073
October 12–17, 2022.

heuristic and support to trigger the dialog routing when users actively click a button or say/type something like ‘transfer to human’. This method is less user-friendly and cannot fully leverage the ability of contemporary deep leaning techniques. Recently, some learning based methods are proposed. Yu et al. well defines the dialog routing as text regression and classification problems, and proposes a learning network with CNN and RNN modules to encode dialog data (Yu et al., 2020). However, it only models the token-level utterance data and misses the information from other kinds of dialog data[1]; Meanwhile the model structure is shared by the two tasks (in single-task way) which cannot capture the underlying knowledge within data-to-task (intrinsic) and task-to-task (extrinsic) relations. In this paper, we investigate the dialog routing task from the multi-task learning perspective to further capture the intrinsic and extrinsic information.

Intuitively, the decision making process of dialog router could be decomposed to two subtasks, NPS prediction and dialog category classification. We adopt the same NPS prediction task as (Yu et al., 2020) since the supervised data can be obtained without any human-labeling effort. Furthermore, we observe that the data-to-task (intrinsic) and task-to-task (extrinsic) information should be leveraged to make better routing performance. On the one hand, we can argue that the more accurate NPS is predicted, the more precise the dialog category is classified, and vice verse. On the other hand, in addition to the token-level utterance data, other kinds of dialog data can also affect the final task performance. For example, if a user utterance is about the intent of ‘ask_technical_problem’, the intent information is an obvious indicator that the user should be served by the agent who is skilled at ‘Technical Issue’ category. Note that the categories of agents and dialog intents of users are often different, and usually the former is much less than the later.

To achieve the above motivation, in this paper, we propose a novel Gated Mechanism enhanced Multi-task Model (G3M). Firstly, the model extends the BERT encoder to encode various kinds of dialog data in a hierarchical way. Moreover, two modules of gated mechanism are proposed to explicitly model the data-to-task and task-to-task information under multi-task learning framework. Another advantage of G3M is its good compatibility, such that it can be easily integrated into existing human-bot dialog systems in a plug-in manner. We conduct various experiments on two datasets collected from the real world. The results demonstrate our method can achieve the state-of-the-art performance on both tasks, and the ablation experiment proves our model is effective to simultaneously capture the intrinsic and extrinsic information.

In summary, this paper’s contributions include:

- We argue that both the data-to-task and task-to-task information are important to achieve better dialog routing. Hence, we propose a new multi-task learning solution, called Gated Mechanism enhanced Multi-task Model (G3M), to implement the motivation.
- We extend the BERT encoder to encode various kinds of dialog data in a hierarchical manner, and develop two modules of gated mechanism to explicitly model the data-to-task and task-to-task information.
- We conduct extensive experiments on two real world datasets. The results prove our model’s effectiveness, which can achieve the state-of-the-art performance.

2 Related Work

Compared with other dialog system tasks, there are not many related work on dialog routing study. Recently, a learning network with attention based CNN and RNN is proposed (Yu et al., 2020). It mainly models the token-level dialog data and is a single-task model, i.e., the NPS regression and dialog classification tasks share the same model structure except the last inference layer. This method cannot sufficiently leverage the underlying knowledge within subtasks and various kinds of dialog data. Another similar work is a demo system for dialog transition (Huang et al., 2021) which tries a vanilla multi-task learning method (i.e., a dialog encoder tailed with two prediction subtasks). However, it still hardly fully utilizes the knowledge among dialog data and subtasks. Different from them, we further investigate the problem by simultaneously modeling data-to-task and task-to-task information, achieving a much advanced performance.

The dialog routing task is fundamentally decomposed to two subtasks, dialog NPS regres-

[1] Other kinds of dialog data include speaker roles, utterance-level sequence, intent information, etc.

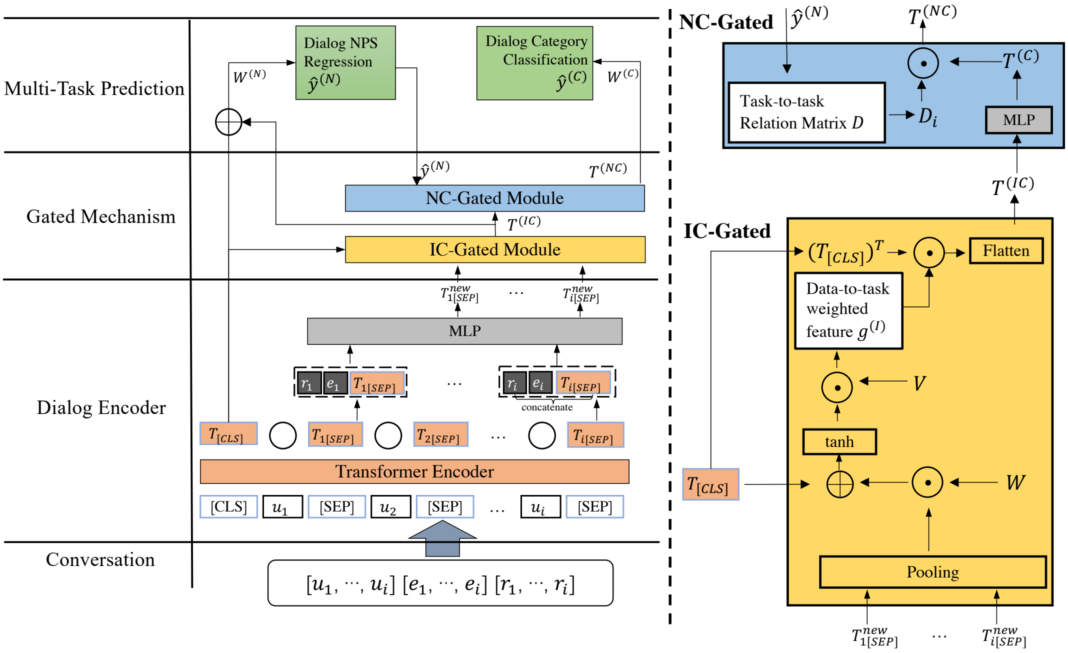

Multi-Task Prediction
Gated Mechanism
Dialog Encoder
Conversation
Dialog NPS Regression $\hat{y}^{(N)}$
Dialog Category Classification $\hat{y}^{(C)}$
$W^{(N)}$
$W^{(C)}$
$\hat{y}^{(N)}$
$T^{(NC)}$
NC-Gated Module
$T^{(IC)}$
IC-Gated Module
$T^{new}_{1[SEP]}$ ··· $T^{new}_{i[SEP]}$
MLP
r_1 e_1 $T_{1[SEP]}$ ··· r_i e_i $T_{i[SEP]}$
concatenate
$T_{[CLS]}$ $T_{1[SEP]}$ $T_{2[SEP]}$ ··· $T_{i[SEP]}$
Transformer Encoder
[CLS] u_1 [SEP] u_2 [SEP] ··· u_i [SEP]
$[u_1, \cdots, u_i]$ $[e_1, \cdots, e_i]$ $[r_1, \cdots, r_i]$
NC-Gated
$\hat{y}^{(N)}$
$T^{(NC)}$
$T^{(C)}$
Task-to-task Relation Matrix D
D_i
MLP
$T^{(IC)}$
IC-Gated
$(T_{[CLS]})^T$
Flatten
Data-to-task weighted feature $g^{(I)}$
V
tanh
$T_{[CLS]}$
W
Pooling
$T^{new}_{1[SEP]}$ ··· $T^{new}_{i[SEP]}$

Figure 2: Architecture overview of Gated Mechanism enhanced Multi-task Model with zoomed in NC-Gated (right top blue part) and IC-Gated (right down yellow part) modules.

sion and dialog classification, which feature encoding various kinds of dialog data and have been separately studied by researchers from the NLP community. The representative research studies include CNN-based models (Kim, 2014), RNN-based models (Wang et al., 2018), regression-based models (Dereli and Saraçlar, 2019; Ngo-Ye and Sinha, 2014) and deep bi-directional transformers model with pre-training (Devlin et al., 2019; Cohan et al., 2019). From these works, we learn that dialog data encoder is the common key module to make sure good performance on final tasks (Dereli and Saraçlar, 2019). Those methods have not been demonstrated to be effective in the multi-task situation of dialog routing, without thoroughly mining the relations among data and subtasks.

Recently, multi-task learning on dialog data has been studied and proven successful, such as joint slot filling and intent prediction (Goo et al., 2018), dialog act sequence labeling (Kumar et al., 2018), and dialog response generation (Ide and Kawahara, 2021). These studies suggest that dialog data contains rich information and multi-task learning could dig out the underlying knowledge among data and subtasks. Moreover, among multi-task learning methods, multi-gated mechanism is widely utilized for information remembering and filtering (Chen et al., 2018; Xiao et al., 2018; Du et al., 2019), which is much suitable for tidying dialog data. Inspired by the above ideas, we propose a novel model to combine them together in our solution for further improving the performance of dialog routing task.

3 Problem Definition

We define the investigated problem formally in this section. In our task, we use the available data in a conversation session as inputs, including an utterance list $U = \{u_1, ..., u_L\}$, a speaker role list $R = \{r_1, ..., r_L\}$, and an intent list $I = \{e_1, ..., e_L\}$, where u_i is an utterance, r_i indicates the role of the ith utterance (user or agent), e_i is known intent of the ith utterance, and L is the number of utterance sequences in a conversation session. Each utterance has a token-level sequence $u_i = \{w_{i1}, ..., w_{iM}\}$, where w_{ij} means the jth token in the ith utterance and M is the length of the token sequence. Here we can leverage the intent information since we use an intent extraction tool[2]. Our model is also scalable to capture more dialog data, such as entity and dialog act information if they are available. We adopt the intent information in our solution based on the intuitive consideration of data-to-task motivation.

For the outputs of our task, at each dialog turn i, our model can make predictions of NPS $y_i^{(N)}$ and dialog category $y_i^{(C)}$. In summary, given $U_i = \{u_1, ..., u_i\}$, $R_i = \{r_1, ..., r_i\}$ and $I_i = \{e_1, ..., e_i\}$, we formally state the problem as follows:

[2] More information can be referred in Section Experiments.

$$y_i^{(N)}, y_i^{(C)} = f(U_i, R_i, I_i). \tag{1}$$

4 Proposed Method

The proposed model generally contains three parts: (1) Dialog Encoder, (2) Gated Mechanism, and (3) Multi-Task Prediction. Figure 2 shows the overview of model's architecture, which will be introduced next.

4.1 Dialog Encoder

To encode the various kinds of dialog data, we leverage and extend the BERT encoder. As presented in Figure 2, given an utterance sequence $\{u_1, ..., u_i\}$, firstly all the utterances are flattened and concatenated to a long token sequence. Then we add a [CLS] token at the beginning and insert [SEP] tokens between any two utterances. Similar to BERT encoder that embeds the whole token sequence by Transformer structure, we get the representation of the ith [SEP] token, $T_{i[SEP]}$, and regard it as the representation of the ith utterance. The representation of head token, $T_{[CLS]}$, would be used in the following Gated Mechanism part as a kind of context information which encodes all the utterances.

With the utterance representations of $T_{i[SEP]}$, the role $\{r_1, ..., r_i\}$ and intent $\{e_i, ..., e_i\}$ information (one-hot vectors) are concatenated to their corresponding utterance representations. After a MLP operation, we can obtain the final representation of each utterance by $T_{i[SEP]}^{new} = MLP(r_i \oplus e_i \oplus T_{i[SEP]})$.

Our method to encode dialog data is inspired by the Sequential Sentence Classification (SCC) model (Cohan et al., 2019) which also is based on BERT encoder and organizes the dialog data in the hierarchical manner. However, our model is different from SCC model in two aspects. Firstly, the representation of head token [CLS] is additionally utilized as context information in the following modules. Secondly, our encoder can integrate the extra role and intent information.

4.2 Gated Mechanism

Besides the basic mechanism via sharing parameters of the encoder in vanilla multi-task learning paradigm, we leverage gated mechanism and propose two modules for dialog routing to better model the task-to-task and data-to-task information.

4.2.1 IC-Gated Module

The **I**ntent-**C**ategory-Gated (IC-Gated) module is designed to model the relation between intent, role and dialog category (data-to-task) information. The right down yellow part in Figure 2 illustrates the detailed module structure.

With the context representation $T_{[CLS]}$ and utterance representations $T_{i[SEP]}^{new}$, we extend a commonly used way (Goo et al., 2018) to capture the relations between intent, role and dialog context by learning a weighted feature for data-to-task (i.e., intent, role to dialog category) modeling. Thus we obtain the weighted feature from various levels of the input dialog data after max pooling operation on the utterance sequence by the following equation:

$$\begin{aligned} g^{(I)} = V \cdot \tanh(&T_{[CLS]} \\ &+ W \cdot Pooling(T_{1[SEP]}^{new}, ..., T_{i[SEP]}^{new})), \end{aligned} \tag{2}$$

where V and W denote the parameters to learn.

With the weighted feature $g^{(I)}$ that attends intent and role information on dialog context, the context information $T_{[CLS]}$ is used again to calculate the final weighted representation of dialog data with a flatten operation as the following equation[3]:

$$T^{(IC)} = Flatten(g^{(I)} \cdot (T_{[CLS]})^{\mathrm{T}}). \tag{3}$$

The IC-Gated module can learn the representation from dialog data and meanwhile consider the related information between intent, role and dialog context, which would benefit the later dialog category classification task.

4.2.2 NC-Gated Module

The **NPS**-**C**ategory-Gated (NC-Gated) module is designed to further model the underlying relation between subtasks (task-to-task). The right top blue part in Figure 2 shows the structure detail.

We set a task-to-task relation matrix $D \in \mathbb{R}^{d^N \times d^C}$ to preserve the empirical distribution of co-occurrence between NPS and dialog categories among dataset with min-max normalization (Singh et al., 2015), where d^N and d^C are the numbers of NPS intervals and dialog categories respectively. We partition the NPS ranging from 0 to 10 into ten intervals. Then with the representation from IC-Gated module $T^{(IC)}$, the output representation of NC-Gated module is calculated by combining a MLP layer and the task-to-task relation matrix D:

[3] $Flatten(\cdot)$ function reshapes the input matrix into a one-dimension vector.

$$T^{(NC)} = D_i \odot MLP(T^{(IC)}), \tag{4}$$

where $\odot$ is Hadamard product to measure the similarity. D_i is the ith row vector which means a representation of the ith NPS interval corresponding to every dialog category, and it depends on the predicted NPS. The reason that uses NPS to query the matrix instead of using predicted dialog category to query NPS, is that we empirically find the effect from the NPS regression task is more significant than the dialog category classification task.

Note that except for the multi-task learning paradigm that can capture the relation between tasks via parameter sharing, we use the relation matrix D as prior knowledge (or attention weights) and design the NC-Gated module for better representation learning of dialog data by further capturing the task-to-task information.

4.3 Multi-Task Prediction

Dialog routing has two subtasks, namely NPS regression and dialog category classification. For the former task, the enhanced representation $T^{(IC)}$ learned by the IC-Gated module from data-to-task information and the dialog context representation $T_{[CLS]}$ are concatenated together to make the prediction of NPS. The predicted NPS $\hat{y}^{(N)}$ can be defined as:

$$\hat{y}^{(N)} = W^{(N)}(T_{[CLS]} \oplus T^{(IC)}) + b^{(N)}. \tag{5}$$

$W^{(N)}$ and $b^{(N)}$ are parameters to learn.

The dialog classification task uses the enhanced representation $T^{(NC)}$ as input, which is learned by the NC-Gated module from task-to-task information. The predicted dialog category $\hat{y}^{(C)}$ is calculated by a softmax function:

$$\hat{y}^{(C)} = Softmax(W^{(C)}T^{(NC)} + b^{(C)}). \tag{6}$$

$W^{(C)}$ and $b^{(C)}$ are parameters to learn.

4.4 Model Training

To jointly train our model from the two tasks together under the multi-task learning paradigm, we design a loss function by combining a mean squared loss for NPS regression and a cross-entropy loss for dialog classification. To avoid over-fitting, L_2 regularization terms are adopted. The overall objective function is defined as:

$$\mathcal{L} = \underbrace{\frac{\alpha}{N}\sum_{k=1}^{N}[\hat{y}_k^{(N)} - y_k^{(N)}]^2}_{\text{NPS regression task}} - \underbrace{\frac{\beta}{N}\sum_{k=1}^{N}\sum_{y} y_k^{(C)} \log(\hat{y}_k^{(C)})}_{\text{dialog classification task}} + \underbrace{\frac{\gamma}{2}(\|W^{(N)}\|_2^2 + \|W^{(C)}\|_2^2)}_{\text{regularization terms}}, \tag{7}$$

where N is the number of all training samples, $\hat{y}_k^{(N)}$ and $\hat{y}_k^{(C)}$ are the inferred NPS and category labels, and $y_k^{(N)}$ and $y_k^{(C)}$ are the ground truth. α, β and γ are hyper-parameters to balance the weights of terms.

5 Experiments

5.1 Corpus

To our knowledge, there is no public corpus contains both the NPS and category labels, we collect two corpora from the real-world dialog systems to evaluate our method.

Learning Corpus is collected from an online education platform, where the dialog system for customer care is constituted of 7 dialog categories with all human agents: Badge Issue, Completion Issue, Finding Content, System Help, Technical Issue, Ticket Status, and Other.

MacHelp Corpus is collected from an after-sale technical support platform for users who have troubles with their Mac laptops, where the dialog system is constituted of 4 dialog categories for human agents: Login Problem, Apps Issue, Raise Tickets, Account Issue.

5.2 Corpus Preprocessing

For the two raw corpora, we preprocess them into a consistent format for model training and testing. Table 1 shows an example of the data structure of dialog session. In a dialog session, we have an utterance sequence, a role sequence to indicate the speaker role of each utterance (0 means user and 1 means agent), and an intent sequence labeled by using an online tool which is separately trained from supervised information for the domain of customer care[4]. We use the off-the-shelf extraction tool and obtain 66 and 28 types of intents on the two corpora respectively.

Within a dialog session, we partition the data to several training and testing samples. At the

[4] https://www.ibm.com/products/watson-assistant. The intent detection accuracy is about 92% in additional assessment.

role	**utterance**	**intent**
1 (agent)	Hello, how may I assist you?	greeting
0 (user)	Hello	greeting
0 (user)	I am looking some training which will help me to learn more about Github.	request
1 (agent)	In Your Learning webpage, you have a search field, you can type what you want to learn about, and it will bring you all courses related to that topic.	guidance
User NPS: 9.0 **Dialog Category**: Finding Content		

Table 1: Data format example of a dialog session.

ith dialog turn, a sample is defined as $S_i = \{\{u_1, ..., u_i\}, \{r_1, ..., r_i\}, \{e_1, ..., e_i\}, y_i^{(N)}, y_i^{(C)}\}$. Ground truths are obtained by the following rules:

- The NPS of u_{i+1} is regarded as the label (i.e., $y_i^{(N)}$) for NPS regression task.
- The agent category of u_{i+1} is regarded as the label (i.e., $y_i^{(C)}$) for dialog classification task.
- If there is no feedback provided by user on the session, our task degrades to gated-mechanism single-task dialog classification.

Note that although the ground truths of NPS and category label are naturally obtained from the users' feedback and human agent category, to guarantee the data quality, three humans are invited to review the correctness of all the labels, and we only adopt the samples agreed by all of them. As a result, we obtain about 90% valid sessions from the corpora. Table 2 shows the statistic information of the final two processed datasets.

The real NPS is from user's feedback by clicking or typing a score, and the two corpora have different NPS ranges (one is an integer in [0,10] and the other is a decimal in [0,5]). Therefore we use zero-mean normalization (z-score) (Singh et al., 2015) method to normalize the scores with a mean of 0 and a standard deviation of 1 as the following: z-score $= \frac{x_i - \overline{x}}{\sigma}$, where x_i is NPS, $\overline{x}$ is the mean value and σ is the standard deviation. By our statistics, we have about 47% and 56% of dialog sessions that have valid NPS labels for Learning and MacHelp datasets respectively, which makes the dialog routing become a much more challenging task in the real world.

5.3 Training Settings

We implement our model based on the Transformer library[5]. We use the Adam (Kingma and Ba, 2014) optimizer to train models on each dataset for 5 epochs. The learning rate is set as $2e$-5 and the dropout rate is set as 0.1. We use the largest batch size that can fit in the memory of GPU. α, β and γ are set as 0.9, 1, 0.01 respectively by our empirical experiments. All the experiments are conducted on V100 32GB GPUs. Each dataset is randomly split into training/validating/testing sets in the proportion of 8:1:1. All the parameters are tuned on the validation set and the results follow the 5-fold cross validation in testing set.

	Learning	MacHelp
avg. tokens per utterance	10.19	10.29
avg. turns per dialog session	22.23	30.43
# of total tokens	66,031	154,023
# of total dialog sessions	22,897	37,944
# of dialog categories	7	4
# of intents	66	28

Table 2: Statistics for Learning and MacHelp datasets

5.4 Baselines

We compare our proposed model to several baselines which belong to two groups. One includes single-task models for NPS prediction and dialog category classification separately, while the other contains methods under multi-task learning paradigm. To make a fair comparison between baselines and the G3M, we concatenate the role and intent information ahead each utterance only if the baseline model can employ that information. Thus all the baseline models have the consistent input information as ours. The evaluated baselines are listed as follows:

- fastText (Joulin et al., 2017) is a simple and efficient model for regression or classification.
- HAN (Yang et al., 2016) is a document classification model based on hierarchical attention network by considering both word-level and sentence-level information.
- textReg (Dereli and Saraçlar, 2019) is a regression model based on convolution neural networks, which achieves the state-of-the-art performance on financial prediction.
- CNN-BiRNN-Att (CBA) (Yu et al., 2020) is a cascade model with CNN, bi-directional RNN and attention mechanism. We adapt the last layer to support either the regression or classification tasks.
- BERT (Devlin et al., 2019) is a model which can be fine-tuned by adjusting the task-specific output layers. We use the pre-trained

[5] https://github.com/huggingface/transformers

BERT-base-cased model to separately fine-tune the two tasks with default parameters.

- XLNET (Yang et al., 2019) is a generalized auto-regressive pre-training method which can be fine-tuned by tailing a task-specific output layers. We fine-tune the two tasks with this pre-trained model similar to BERT.
- Joint-CNN-BiRNN-Att (Joint-CBA) is a multi-task model by extending the last layer of CBA model to simultaneously predict NPS and dialog categories.
- Vanilla Multi-task Model (VMM) follows the vanilla multi-task learning paradigm and uses the standard BERT encoder to encode only the utterance data (Huang et al., 2021), without any gated-mechanism modules.

To measure the performance, we use the root mean squad error (RMSE) for the NPS regression task and micro F1 score (Micro-F1) for the dialog category classification task.

5.5 Results and Analysis

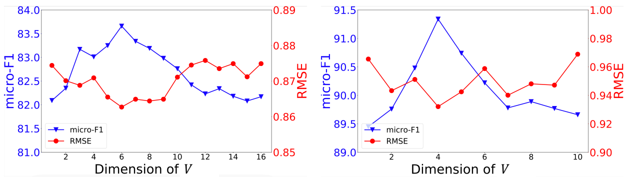
We compare our model with various single-task and multi-task baselines on the two datasets, and the first two sections of Table 3 report the results. The performance of NPS regression task is reflected by RMSE metric, while that of dialog category classification task is embodied by Micro-F1 metric. We can learn some observations from Table 3.

Firstly, the performance of our proposed model G3M is consistent on both datasets and can outperform all the baselines. More specifically, for the NPS regression task, G3M's RMSE is 8.72% and 11.83% lower than the best RMSE score of baseline models (VMM for Learning dataset and Joint-CBA for MacHelp dataset) on two datasets. And for the dialog category classification task, G3M upgrades the performance by achiving 2.17% and 4.40% higher Micro-F1 scores compared with the best baseline method (VMM). All the results demonstrate that our model is effective and achieves the state-of-the-art performance on both subtasks.

Secondly, comparing the single-task models with multi-task models among baselines, it is consistent that multi-task models can surpass the single-task counterparts, especially from the comparison between Joint-CBA and VMM with their single-task versions (CBA and BERT). The results suggest that the underlying relation between the two tasks is helpful and multi-task learning paradigm can well capture the knowledge.

Models		Learning		MacHelp	
		RMSE	Micro-F1	RMSE	Micro-F1
Single-task	fastText	0.9711	77.85	1.2365	85.88
	HAN	0.9981	78.85	1.1276	84.19
	textReg	0.9689	77.98	1.1099	85.55
	CBA	0.9697	79.42	1.1383	86.02
	BERT	0.9702	80.04	1.1025	86.39
	XLNET	0.9648	80.21	1.1084	86.25
Multi-task	Joint-CBA	0.9555	80.38	1.0624	87.21
	VMM	0.9302	81.88	1.0767	87.49
	G3M	**0.8491**	**83.66**	**0.9367**	**91.34**
Ablation	- IC-Gated module	0.8892	82.57	0.9883	90.63
	- NC-Gated module	0.8878	82.73	0.9831	90.36
	- both gated modules	0.9187	82.01	1.0591	88.27

Table 3: Comparison between G3M and baselines on Learning and MacHelp datasets. RMSE is for NPS prediction and Micro-F1 (%) is for dialog classification.



(a) Learning Corpus (b) MacHelp Corpus

Figure 3: The parameter effect on different dimensions of V in IC-Gated module of G3M.

Thirdly, comparing three multi-task models, we observe that G3M is better than the other two methods. We can conclude that both our proposed dialog encoder and gated mechanism are effective, since Joint-CBA and VMM do not contain either of the modules. Another observation is that under the same multi-task learning paradigm, BERT based dialog encoder (VMM) is better than the CNN-RNN based dialog encoder (Joint-CBA).

5.6 Ablation Study

To probe G3M's effectiveness in terms of gated module level, we conduct ablation experiments and report the results in the bottom section of Table 3. We could obtain three variant models by once deducting one or two gated modules: G3M without IC-Gated module, G3M without NC-Gated module, and G3M without either module. Based on the experiments, we have some findings.

Comparing the three ablated variant models of G3M, both modules contribute positively to the two tasks. Therefore, they are effective to model the data-to-task and task-to-task information among dialog data. Specifically, it is not very clear that which module is more important than the other. However, ablating both them would damage the final performance.

Furthermore, we compare the performance between G3M without both gated modules and multi-task baselines (Joint-CBA and VMM), we find the former model performs better than the later, which may indicate that our proposed dialog encoder is superior than the CNN-RNN based and BERT based dialog encoders.

5.7 Parameter Effect

In the IC-Gated module, there is an important parameter to be preset by humans, which is the dimension of V. It determines the size of weighted feature $g^{(I)}$ and therefore controls to what degree the data-to-task information can be leveraged. We investigate the effect of various dimensions of V on both datasets. From the curves of RMSE and Micro-F1 scores in Figure 3, we find that the final performance is related to this parameter. Empirically, the optimal setting is 6 for Learning dataset and 4 for MacHelp dataset. We can also observe too small or too large dimensions may both play a harmful effect on extracting important knowledge from the data-to-task information. Another interesting phenomenon is that the optimal parameter values are coincidentally close or equal to the numbers of dialog categories to be classified, and we would explore the potential correlation in the future.

5.8 Case Study

We conduct a case study to demonstrate how our method could work to improve the system efficiency and user experience by integrating our dialog routing component into a human-bot symbiosis dialog system. Table 4 lists the dialog utterances along with the predicted NPS and dialog category. Here the user (U) has a requirement to solve course enrollment issue. At first, the bot agent (A1) serves the user which is skilled at 'Finding Content' dialogs, but it wrongly understands the user that she would find some course materials. By using the dialog routing component to monitor the dialogs, the predicted NPS decreases from 5.8 to 4.8, which means the user is dissatisfactory with the A1 and the routing to other agent should be triggered[6]. As a result, based on the dialog category classification by dialog routing component, the dialog is transferred to a human agent (A2) who is skilled at 'Completion Issue' dialogs. Thanks to the right time to transfer and the correctly assigned agent, the NPS is up to 7.8 and the user is finally satisfactory with this service.

Role	Utterances
U	Hi
A1	Hello, how may I assist you?
U	I have enrolled in Deep Learning Course but today I can't see this enrollment
U	I have completed more than 80%
U	Can you please help me to find / resume the course
A1	I'm sorry that we are not able to do that
U	Why so
U	and what help you can do?
Current NPS: 5.8 Current dialog category: Finding Content	
A1	You could get more information in the completion dashboard in the system
U	You did not answer my question, I can't find the course and I want to resume it
U	Can you please help me to find / resume the course
A1	I'm sorry that we are not able to do that
Current NPS: 4.8 (**under the threshold and trigger routing**) Current dialog category: Completion Issue	
A2	Sorry for not being very helpful.
A2	You can't find the course in your learning queue and the progress are all gone, correct?
U	Exactly
Current NPS: 6.7 Current dialog category: Completion Issue	
A2	Is it an internal course in the Learning system, right?
U	Yes
Current NPS: 6.9 Current dialog category: Completion Issue	
A2	Sorry for your confusion, sometimes it may be caused by technical issues.
A2	A ticket has been created for you so that it will be reflected correctly.
A2	Normally the technical team will get back to you in 24 hours
U	Thanks, bye
A2	Bye
Current NPS: 7.8 Current dialog category: Completion Issue	

Table 4: A dialog session along with the predicted NPS and dialog category by equipping a dialog routing component with our method.

This case could also reveal that our method would not affect the existing dialog systems and the dialog routing system can be deployed in a plug-in manner, which has a good compatibility and is convenient for both dialog routing system and dialog system to upgrade their ability.

6 Conclusion

In current ubiquitous human-bot symbiosis dialog systems for customer care, the dialog routing component is necessary to improve the overall system efficiency, reduce human resource cost, and enhance user experience. In this paper, we argue that the data-to-task and task-to-task information among various kinds of dialog data and subtasks should be jointly leveraged to perform better dialog routing ability. We propose a Gated Mechanism

[6] We can set a dissatisfaction threshold based on real situations, e.g. 5 in this case.

enhanced Multi-task Model (G3M) to implement that motivation. Specifically, we design a new dialog encoder to learn various kinds of dialog data by extending the BERT encoder, and two gated mechanism modules are proposed to capture data-to-task and task-to-task information. Extensive experiments on two real-world datasets demonstrate the effectiveness of our proposed model, which can achieve the state-of-the-art performance.

Acknowledgement

We thank all the anonymous reviewers for their valuable feedback and insightful comments. This work is partially supported by the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20220488).

References

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1267–1276.

Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Daniel S Weld. 2019. Pretrained language models for sequential sentence classification. *arXiv preprint arXiv:1909.04054*.

Neşat Dereli and Murat Saraçlar. 2019. Convolutional neural networks for financial text regression. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 331–337.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zefeng Du, Peijie Huang, Yuhong He, Wei Liu, and Jiankai Zhu. 2019. A knowledge-gated mechanism for utterance domain classification. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 142–154. Springer.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

Ziming Huang, Zhuoxuan Jiang, Hao Chen, Xue Han, and Yabin Dang. 2021. Dialog router: Automated dialog transition via multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 16038–16040.

Tatsuya Ide and Daisuke Kawahara. 2021. Multi-task learning of generation and classification for emotion-aware dialogue response generation. *CoRR*, abs/2105.11696.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2017. Fasttext. zip: Compressing text classification models. In *The International Conference on Learning Representations (ICLR)*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Conference on Empirical Methods in Natural Language Processing*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Harshit Kumar, Arvind Agarwal, Riddhiman Dasgupta, and Sachindra Joshi. 2018. Dialogue act sequence labeling using hierarchical encoder with crf. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Thomas L Ngo-Ye and Atish P Sinha. 2014. The influence of reviewer engagement characteristics on online review helpfulness: A text regression model. *Decision Support Systems*, 61:47–58.

Shereen Oraby, Pritam Gundecha, Jalal Mahmud, Mansurul Bhuiyan, and Rama Akkiraju. 2017. How may i help you?: Modeling twitter customer serviceconversations using fine-grained dialogue acts. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 343–355. ACM.

Bikesh Kumar Singh, Kesari Verma, and AS Thoke. 2015. Investigations on impact of feature normalization techniques on classifier's performance in breast tumor classification. *International Journal of Computer Applications*, 116(19).

Jenq-Haur Wang, Ting-Wei Liu, Xiong Luo, and Long Wang. 2018. An lstm approach to short text sentiment classification with word embeddings. In *Proceedings of the 30th conference on computational linguistics and speech processing (ROCLING 2018)*, pages 214–223.

Liqiang Xiao, Honglun Zhang, and Wenqing Chen. 2018. Gated multi-task network for text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 726–731. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Yipeng Yu, Ran Guan, Jie Ma, Zhuoxuan Jiang, and Jingchang Huang. 2020. When and who? conversation transition based on bot-agent symbiosis learning network. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 4056–4066. International Committee on Computational Linguistics.