Multi-Domain Dialogue Acts and Response Co-Generation

Kai Wang[1], Junfeng Tian[2], Rui Wang[2], Xiaojun Quan[1]*, Jianxing Yu[1]
[1]School of Data and Computer Science, Sun Yat-sen University, China
[2]Alibaba Group, China
wangk73@mail2.sysu.edu.cn, {quanxj3, yujx26}@mail.sysu.edu.cn
{tjf141457, masi.wr}@alibaba-inc.com

Abstract

Generating fluent and informative responses is of critical importance for task-oriented dialogue systems. Existing pipeline approaches generally predict multiple dialogue acts first and use them to assist response generation. There are at least two shortcomings with such approaches. First, the inherent structures of multi-domain dialogue acts are neglected. Second, the semantic associations between acts and responses are not taken into account for response generation. To address these issues, we propose a neural co-generation model that generates dialogue acts and responses concurrently. Unlike those pipeline approaches, our act generation module preserves the semantic structures of multi-domain dialogue acts and our response generation module dynamically attends to different acts as needed. We train the two modules jointly using an uncertainty loss to adjust their task weights adaptively. Extensive experiments are conducted on the large-scale MultiWOZ dataset and the results show that our model achieves very favorable improvement over several state-of-the-art models in both automatic and human evaluations.

1 Introduction

Task-oriented dialogue systems aim to facilitate people with such services as hotel reservation and ticket booking through natural language conversations. Recent years have seen a rapid proliferation of interests in this task from both academia and industry (Bordes et al., 2017; Budzianowski et al., 2018; Wu et al., 2019). A standard architecture of these systems generally decomposes this task into several subtasks, including natural language understanding (Gupta et al., 2018), dialogue state tracking (Zhong et al., 2018) and natural language
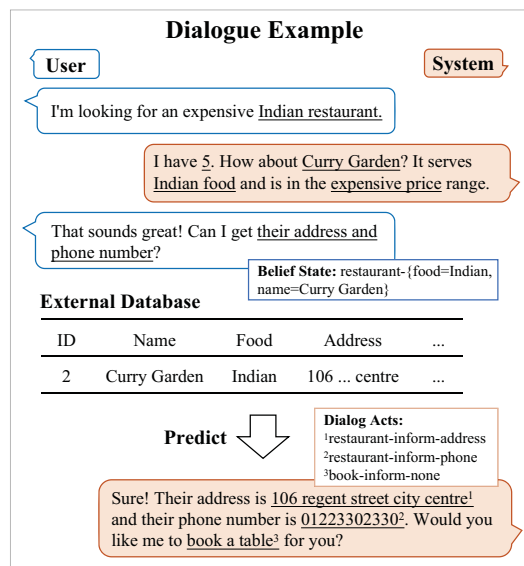

Figure 1: An example of dialogue from the MultiWOZ dataset, where the dialogue system needs to generate a natural language response according to current belief state and related database records.

generation (Su et al., 2018). They can be modeled separately and combined into a pipeline system.

Figure 1 shows a dialogue example, from which we can notice that the natural language generation subtask can be further divided into dialogue act prediction and response generation (Chen et al., 2019; Zhao et al., 2019; Wen et al., 2017). While the former is intended to predict the next action(s) based on current conversational state and database information, response generation is used to produce a natural language response based on the action(s).

In order for dialogues to be natural and effective, responses should be fluent, informative, and relevant. Nevertheless, current sequence-to-sequence models often generate uninformative responses like "I don't know" (Li et al., 2016a), hindering the dialogues to continue or even leading to a failure. Some researchers (Pei et al., 2019; Mehri et al.,
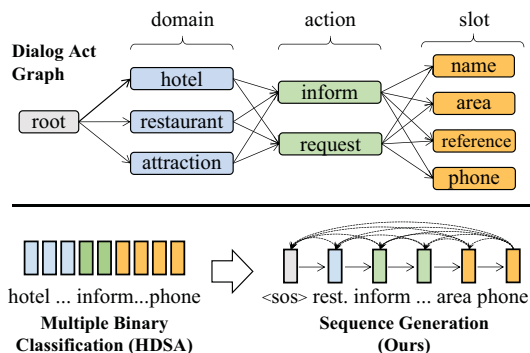

Figure 2: Demonstration of hierarchical dialogue act structures (top) and different approaches (bottom) for dialogue act prediction. Classification approaches separately predict each act item (domain, action and slot), while generation approaches treat each act as a token that can be generated sequentially.

2019) sought to combine multiple decoders into a stronger one to avoid such responses, while others (Chen et al., 2019; Wen et al., 2015; Zhao et al., 2019; Wen et al., 2017) represent dialogue acts in a global, static vector to assist response generation.

As pointed out by Chen et al. (2019), dialogue acts can be naturally organized in hierarchical structures, which has yet to be explored seriously. Take two acts *station-request-stars* and *restaurant-inform-address* as an example. While the first act rarely appears in real-world dialogues, the second is more often. Moreover, there can be multiple dialogue acts mentioned in a single dialogue turn, which requires the model to attend to different acts for different sub-sequences. Thus, a global vector is unable to capture the inter-relationships among acts, nor is it flexible for response generation especially when more than one act is mentioned.

To overcome the above issues, we treat dialogue act prediction as another sequence generation problem like response generation and propose a co-generation model to generate them concurrently. Unlike those classification approaches, act sequence generation not only preserves the inter-relationships among dialogue acts but also allows close interactions with response generation. By attending to different acts, the response generation module can dynamically capture salient acts and produce higher-quality responses. Figure 2 demonstrates the difference between the classification and the generation approaches for act prediction.

As for training, most joint learning models rely on hand-crafted or tunable weights on development sets (Liu and Lane, 2017; Mrkšić et al., 2017; Ras-

togi et al., 2018). The challenge here is to combine two sequence generators with varied vocabularies and sequence lengths. The model is sensitive during training and nontrivial to generate an optimal weight. To address this issue, we opt for an uncertainty loss (Kendall et al., 2018) to adaptively adjust the weight according to task-specific uncertainty. We conduct extensive studies on a large-scale task-oriented dataset to evaluate the model. The experimental results confirm the effectiveness of our model with very favorable performance over several state-of-the-art methods.

The contributions of this work include:

- We model dialogue act prediction as a sequence generation problem that allows to exploit act structures for the prediction.

- We propose a co-generation model to generate act and response sequences jointly, with an uncertainty loss used for adaptive weighting.

- Experiments on MultiWOZ verify that our model outperforms several state-of-the-art methods in automatic and human evaluations.

2 Related Work

Dialogue act prediction and response generation are closely related in general in the research of dialogue systems (Chen et al., 2019; Zhao et al., 2019; Wen et al., 2017), where dialogue act prediction is first conducted and used for response generation. Each dialogue act can be treated as a triple (domain-action-slot) and all acts together are represented in a one-hot vector (Wen et al., 2015; Budzianowski et al., 2018). Such sparse representation makes the act space very large. To overcome this issue, Chen et al. (2019) took into account act structures and proposed to represent the dialogue acts with level-specific one-hot vectors. Each dimension of the vectors is predicted by a binary classifier.

To improve response generation, Pei et al. (2019) proposed to learn different *expert* decoders for different domains and acts, and combined them with a *chair* decoder. Mehri et al. (2019) applied a cold-fusion method (Sriram et al., 2018) to combine their response decoder with a language model. Zhao et al. (2019) treated dialogue acts as latent variables and used reinforcement learning to optimize them. Reinforcement learning was also applied to find optimal dialogue policies in task-oriented dialogue systems (Su et al., 2017;

Williams et al., 2017) or obtain higher dialog-level rewards in chatting (Li et al., 2016b; Serban et al., 2017). Besides, Chen et al. (2019) proposed to predict the acts explicitly with a compact act graph representation and employed hierarchical disentangled self-attention to control response text generation.

Unlike those pipeline architectures, joint learning approaches try to explore the interactions between act prediction and response generation. A large body of research in this direction uses a shared user utterance encoder and train natural language understanding jointly with dialogue state tracking (Mrkšić et al., 2017; Rastogi et al., 2018). Liu and Lane (2017) proposed to train a unified network for two subtasks of dialogue state tracking, i.e., knowledge base operation and response candidate selection. Jiang et al. (2019) showed that joint learning of dialogue act and response benefits representation learning. These works generally demonstrate that joint learning of the subtasks of dialogue systems is able to improve each other and the overall system performance.

3 Architecture

Let $T = \{U_1, R_1, \ldots, U_{t-1}, R_{t-1}, U_t\}$ denote the dialogue history in a multi-turn conversational setting, where U_i and R_i are the i-th user utterance and system response, respectively. $D = \{d_1, d_2, \ldots, d_n\}$ includes the attributes of related database records for current turn. The objective of a dialogue system is to generate a natural language response $R_t = y_1 y_2 \ldots y_n$ of n words based on the current belief state and database attributes.

In our framework, dialogue acts and response are co-generated based on the transformer encoder-decoder architecture (Vaswani et al., 2017). A standard transformer includes a multi-head attention layer that encodes a value V according to the attention weights from query Q to key K, followed by a position-wise feed-forward network ($\mathcal{G}_f$):

$$O = V + \mathcal{G}_f(MultiHead(Q, K, V)) \quad (1)$$

where $Q, K, V, O \in \mathbb{R}^{n \times d}$. In what follows we use $\mathcal{F}(Q, K, V)$ to denote the transformer.

Encoder   We use $E = \text{Emb}([T; D])$ to represent the concatenated word embeddings of dialogue history T and database attributes D. The transformer $\mathcal{F}(Q, K, V)$ is then used to encode E and output its hidden state H^e:

$$H^e = \mathcal{F}(E, E, E) \quad (2)$$

Decoder   At each time step t of response generation, the decoder first computes a self-attention h_t^r over already-generated words $y_{1:t-1}$:

$$h_t^r = \mathcal{F}(e_{t-1}^r, e_{1:t-1}^r, e_{1:t-1}^r) \quad (3)$$

where e_{t-1}^r is the embedding of the $(t-1)$-th generated word and $e_{1:t-1}^r$ is an embedding matrix of e_1^r to e_{t-1}^r. Cross-attention from h_t^r to dialogue history T is then executed:

$$c_t^r = \mathcal{F}(h_t^r, H^e, H^e) \quad (4)$$

The resulting vectors of Equations 3 and 4, h_t^r and c_t^r, are concatenated and mapped to a distribution of vocabulary size to predict next word:

$$p(y_t|y_{1:t-1}) = \text{softmax}(W_r[c_t^r; h_t^r]) \quad (5)$$

4 The MARCO Approach

Based on the above encoder-decoder architecture, our model is designed to consist of three components, namely, a shared encoder, a dialogue act generator, and a response generator. As shown in Figure 3, instead of predicting each act token individually and separately from response generation, our model aims to generate act sequence and response concurrently in a joint model which is optimized by the uncertainty loss (Kendall et al., 2018).

4.1 Dialogue Acts Generation

Dialogue acts can be viewed as a semantic plan for response generation. As shown in Figure 2, they can be naturally organized in hierarchical structures, including domain level, action level, and slot level. Most existing methods treat dialogue acts as triples represented in one-hot vectors and predict the vector values with binary classifiers (Wen et al., 2015; Budzianowski et al., 2018). Such representations ignore the inter-relationships and associations among acts, domains, actions and slots. For example, the slot *area* may appear in more than one domain. Unlike them, we model the prediction of acts as a sequence generation problem, which takes into consideration the structures of acts and generates each act token conditioned on its previously-generated tokens. In this approach, different domains are allowed to share common slots and the search space of dialogue act is greatly reduced.

The act generation starts from a special token "⟨SOS⟩" and produces dialogue acts $A = a_1 a_2 \ldots a_n$ sequentially. During training, the act
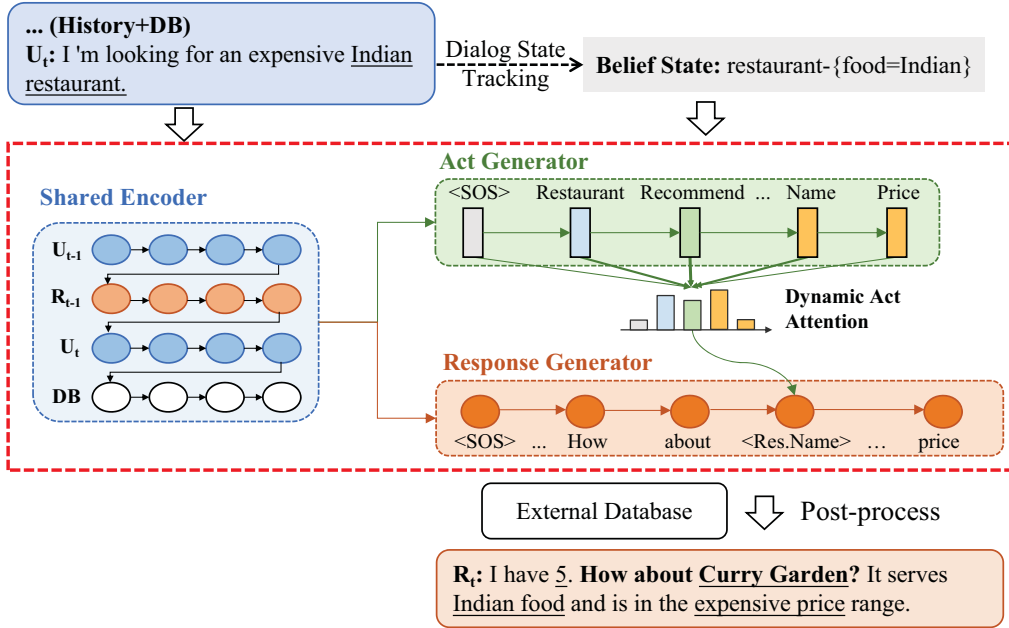
Figure 3: Architecture of the proposed model for act and response co-generation, where act and response generators share the same encoder. The response generator is allowed to attend to different act hidden states as needed using dynamic act attention. The two generators are trained jointly and optimized by the uncertainty loss.

sequence is organized by domain, action and slot, while items at each level are arranged in dictionary order, where identical items are merged. When decoding each act token, we first represent the current belief state with an embedding vector v_b and add it to each act word embedding e_t^a as:

$$u_t^a = W_b v_b + e_t^a. \quad (6)$$

Finally, the decoder of Section 3.2 is used to generate hidden states H^a and act tokens accordingly.

4.2 Acts and Response Co-Generation

Dialogue acts and responses are closely related in dialogue systems. On one hand, system responses are generated based on dialogue acts. On the other, their shared information can improve each other through joint learning.

Shared Encoder Our dialogue act generator and response generator share one same encoder and input, but having different masking strategies for the input to focus on different information. In particular, only the current utterance is kept for act generation, while the entire history utterances are used for response generation.[1]

---

[1] Empirical evidences show that act generation is more related to the current utterance, while response generation benefits more from long dialogue history.

Dynamic Act Attention A response usually corresponds to more than one dialogue act in multi-domain dialogue systems. Nevertheless, existing methods mostly use a static act vector to represent all the acts, and add the vector to each response token representation. They ignore the fact that different subsequences of a response may need to attend to different acts. To address this issue, we compute dynamic act attention o_t^r from the response to acts when generating a response word:

$$o_t^r = \mathcal{F}(h_t^r, H^a, H^a) \quad (7)$$

where h_t^r is the current hidden state produced by Equation 3. Then, we combine o_t^r and h_t^r with response-to-history attention c_t^r (by Equation 4) to estimate the probabilities of next word:

$$p(y_t|y_{1:t-1}) = \text{softmax}(W_r[h_t^r; c_t^r; o_t^r]) \quad (8)$$

Uncertainty Loss The cross-entropy function is used to measure the generation losses, $\mathcal{L}_a(\theta)$ and $\mathcal{L}_r(\theta)$, of dialogue acts and responses, respectively:

$$\mathcal{L}_a(\theta) = -\sum_{j=1}^{T_a} \log p(a_j^{*(i)}|a_{1:j-1}^{(i)}, T, D, v_b) \quad (9)$$

$$\mathcal{L}_r(\theta) = -\sum_{j=1}^{T_r} \log p(y_j^{*(i)}|y_{1:j-1}^{(i)}, T, D, A) \quad (10)$$

where the ground-truth tokens of acts and response of each turn are represented by A^* and Y^*, while the predicted tokens by A and Y.

To optimize the above functions jointly, a general approach is to compute a weighted sum like:

$$\mathcal{L}(\theta) = \alpha\mathcal{L}_a(\theta) + (1-\alpha)\mathcal{L}_r(\theta) \qquad (11)$$

However, dialogue acts and responses vary seriously in sequence length and vocabulary size, making the weight α unstable to tune. Instead, we opt for an uncertainty loss (Kendall et al., 2018) to adjust it adaptively:

$$\mathcal{L}(\theta, \sigma_1, \sigma_2) = \frac{1}{2\sigma_1^2}\mathcal{L}_a(\theta) + \frac{1}{2\sigma_2^2}\mathcal{L}_r(\theta) + \log\sigma_1^2\sigma_2^2 \qquad (12)$$

where σ_1 and σ_2 are two learnable parameters. The advantage of this uncertainty loss is that it models the homoscedastic uncertainty of each task and provides task-dependent weight for multi-task learning (Kendall et al., 2018). Our experiments also confirm that it leads to more stable weighting than the traditional approach (Section 6.3).

5 Experiments

5.1 Dataset and Metrics

MultiWOZ 2.0 (Budzianowski et al., 2018) is a large-scale multi-domain conversational datatset consisting of thousands of dialogues in seven domains. For fair comparison, we use the same validation set and test set as previous studies (Chen et al., 2019; Zhao et al., 2019; Budzianowski et al., 2018), each set including 1000 dialogues.[2] We use the *Inform Rate* and *Request Success* metrics to evaluate dialog completion, with one measuring whether a system has provided an appropriate entity and the other assessing if it has answered all requested attributes. Besides, we use BLEU (Papineni et al., 2002) to measure the fluency of generated responses. To measure the overall system performance, we compute a combined score: $(Inform\,Rate + Request\,Success) \times 0.5 + BLEU$ as before (Budzianowski et al., 2018; Mehri et al., 2019; Pei et al., 2019).

5.2 Implementation Details

The implementation[3] is on a single Tesla P100 GPU with a batch size of 512. The dimension of

---

word embeddings and hidden size are both set to 128. We use a 3-layer transformer with 4 heads for the multi-head attention layer. For decoding, we use a beam size of 2 to search for optimal results, and apply trigram avoidance (Paulus et al., 2018) to fight trigram-level repetition. During training, we first train the act generator for 10 epochs for warm-up and then optimize the uncertainty loss with the Adam optimizer (Kingma and Ba, 2015).

5.3 Baselines

A few mainstream models are used as baselines for comparison with our neural co-generation model (MARCO), being categorized into three categories:

- **Without Act**. Models in this category directly generate responses without act prediction, including LSTM (Budzianowski et al., 2018), Transformer (Vaswani et al., 2017), Token-MoE (Pei et al., 2019) and Structured Fusion (Mehri et al., 2019).

- **One-Hot Act**. In SC-LSTM (Wen et al., 2015), dialogue acts are treated as triples and information flow from acts to response generation is controlled by gates. HDSA (Chen et al., 2019) is a strong two-stage model, which relies on BERT (Devlin et al., 2019) to predict a one-hot act vector for response generation.

- **Sequential Act**. Since our model does not rely on BERT, to make a fair comparison with HDSA, we design the experiments from two aspects to ensure they have the same dialogue act inputs for response generation. First, the act sequences produced by our co-generation model are converted into one-hot vectors and fed to HDSA. Second, the predicted one-hot act vectors by BERT are transformed into act sequences and passed to our model as inputs.

5.4 Overall Results

The overall results are shown in Table 1, in which HDSA (MARCO) means HDSA using MARCO's dialogue act information, and MARCO (BERT) means MARCO based on BERT's act prediction. From the table we can notice that our co-generation model (MARCO) outperforms all the baselines in *Inform Rate*, *Request Success*, and especially in *combined score* which is an overall metric. By comparing the two HDSA models, we can find HDSA derives its main performance from the external BERT, which can also be used to improve our MARCO considerably (MARCO (BERT)). These

Dialog Act	Model	Inform	Success	BLEU	Combined Score
Without Act	LSTM	71.29	60.96	18.80	84.93
	Transformer	71.10	59.90	19.10	84.60
	TokenMoE	75.30	59.70	16.81	84.31
	Structured Fusion	82.70	72.10	16.34	93.74
One-hot Act	SC-LSTM	74.50	62.50	20.50	89.00
	HDSA (MARCO)	76.50	62.30	21.85	91.25
	HDSA	82.90	68.90	**23.60**	99.50
Sequential Act	MARCO	90.30	75.20	19.45	102.20
	MARCO (BERT)	**92.30**	**78.60**	20.02	**105.47**

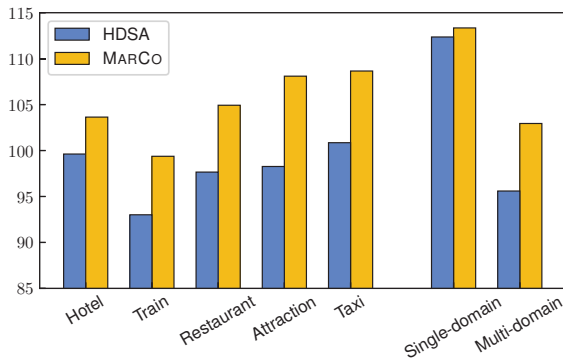Table 1: Overall results on the MultiWOZ 2.0 dataset.

Figure 4: Combined score of MARCO vs. HDSA across different domains. If a dialogue involves more than one domain, it is copied into each domain. Single-domain includes dialogues with only one domain mentioned, while the rest belongs to the multi-domain.

results confirm the success of MARCO by modeling act prediction as a generation problem and training it jointly with response generation.

Another observation is that despite its strong overall performance, MARCO shows inferior BLEU performance to the two HDSA models. The reason behind this is studied and analyzed in human evaluation (Section 7), showing that our model often generates responses inconsistent with references but favored by human judges.

The performance of our model across different domains is also compared against HDSA. The average number of turns is 8.93 for single-domain dialogues and 15.39 for multi-domain dialogues (Budzianowski et al., 2018). As in Figure 4, our model shows superior performance to HDSA across all domains. The results suggest that MARCO is good at dealing with long dialogues.

Results on MultiWOZ 2.1 We also conducted experiments on MultiWOZ 2.1 (Eric et al., 2019),

Model	Inform	Success	BLEU	Score
Transformer	72.50	52.70	19.08	81.68
HDSA	86.30	70.60	**22.36**	100.81
MARCO	91.50	76.10	18.52	102.32
MARCO (BERT)	**92.50**	**77.80**	19.54	**104.69**

Table 2: Overall results on the MultiWOZ 2.1 dataset.

which is an updated version of MultiWOZ 2.0. As shown in Table 2, the overall results are consistent with that on MultiWOZ 2.0.

6 Further Analysis

More thorough studies and analysis are conducted in this section, trying to answer three questions: (1) How is the performance of our act generator in comparison with existing classification methods? (2) Can our joint model successfully build semantic associations between acts and responses? (3) How does the uncertainty loss contribute to our co-generation model?

6.1 Dialogue Act Prediction

To evaluate the performance of our act generator, we compare it with several baseline methods mentioned in (Chen et al., 2019), including BiLSTM, Word-CNN, and 3-layer Transformer. We use MARCO to represent our act generator which is trained jointly with the response generator, and use Transformer (GEN) to denote our act generator without joint training. From Table 3, we notice that the separate generator, Transformer (GEN), performs much better than BiLSTM and Word-CNN, but comparable with Transformer. But after trained jointly with the response generator, MARCO manages to show the best performance, confirming the effect of the co-generation.

Method	F1
BiLSTM	71.4
Word-CNN	71.5
Transformer	73.1
Transformer (GEN)	73.2
MARCO	**73.9**

Table 3: Results of different act generation methods, where BiLSTM, Word-CNN and Transformer are baselines from (Chen et al., 2019). MARCO is our act generator trained jointly with the response generator and Transformer (GEN) is that without joint training.

Model	Inform	Succ	BLEU	Combined
HDSA	82.9	68.9	23.60	99.50
Pipeline$_1$	84.3	54.4	16.00	85.35
Pipeline$_2$	86.6	66.0	18.31	94.61
Joint	**90.3**	**75.2**	**19.45**	**102.20**

Table 4: Results of response generation by joint and pipeline models, where Pipeline$_1$ and Pipeline$_2$ represent two pipeline approaches with or without using dynamic act attention. The performance of HDSA, as the best pipeline model, is provided for comparison.

6.2 Joint vs. Pipeline

To study the influence of the joint training and the dynamic act attention on response generation, we implement two pipeline approaches for comparison. We first train our act generator separately from response generation. Then, we keep its parameters fixed and train the response generator. The first baseline is created by replacing the dynamic act attention (Equation 7) with an average of the act hidden states, while the second baseline uses the dynamic act attention. As shown in Table 4, Pipeline$_2$ with dynamic act attention is superior to Pipeline$_1$ without it in all metrics, but inferior to the joint approach. Our joint model also surpasses the currently state-of-the-art pipeline system HDSA, even HDSA uses BERT. We find that by utilizing sequential acts, the dynamic act attention mechanism helps the response generator capture the local information by attending to different acts.

An illustrative example is shown in Figure 5, where the response generator can attend to the local information such as "day" and "stay" as needed when generating a response asking about picking a different day or shorter stay. We reckon that by utilizing sequential acts, response generation benefits in two ways. First, the dynamic act attention allows the generator to attend to different acts when
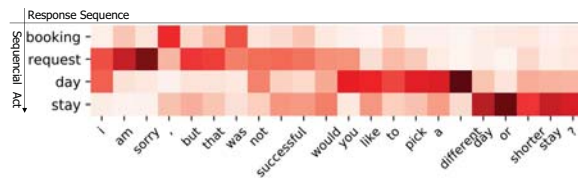

Figure 5: An illustrative example of the dynamic act attention mechanism. Response (row) subsequence can attend to the act (column) token "day" or "stay" as needed when generating a response asking about picking a different day or shorter stay.
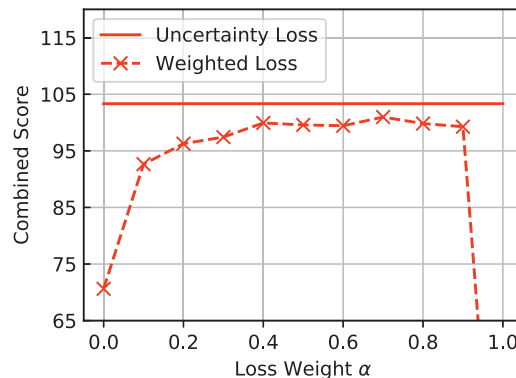

Figure 6: Performance of the uncertainty loss and the weighted-sum loss on the development dataset.

generating a subsequence. Second, the joint training makes the two stages interact with each other, easing error propagation of pipeline systems.

6.3 Uncertainty Loss

We opt for an uncertainty loss to optimize our joint model, rather than a traditional weighted-sum loss. To illustrate their difference, we conduct an experiment on the development set. For the traditional loss (Equation 11), we run for each weight from 0 to 1 stepped by 0.1. Note that since the weights, σ_1 and σ_2, in the uncertainty loss are not hyperparameters but learned internally to each batch, we only record the best score within each round without giving the values of σ_1 and σ_2. As shown in Figure 6, the uncertainty loss can learn adaptive weights with consistently superior performance.

7 Human Evaluation

We conduct a human study to evaluate our model by crowd-sourcing.[4] For this purpose we randomly selected 100 sample dialogues (742 turns in total) from the test dataset and constructed two groups of systems for comparison: MARCO vs. HDSA and

---

[4]The annotation results are available at `https://github.com/InitialBug/MarCo-Dialog/tree/master/human_evaluation`
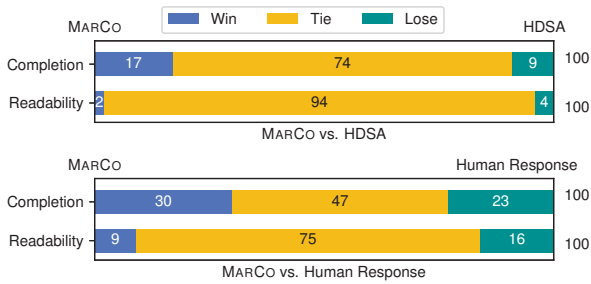
7131

Figure 7: Results of human study in response quality. Two groups of systems are studied, where the top figure corresponds to results of MARCO vs. HDSA and the bottom figure represents MARCO vs. Human Response (ground-truth). "Win", "Tie" or "Lose" respectively indicate the proportions that our MARCO system wins over, ties with or loses to its counterpart.

MARCO vs. Human Response, where Human Response means the reference responses. Responses generated by each group were randomly assigned in pairs to 3 judges, who ranked them according to their completion and readability (Chen et al., 2019; Zhang et al., 2019). *Completion* measures if the response correctly answers a user query, including relevance and informativeness. *Readability* reflects how fluent, natural and consistent the response is.

The results of this study are shown in Figure 7, where "Win", "Tie" or "Lose" mean our MARCO system wins over, ties with or loses to its counterpart, respectively. From the results we note that MARCO outperforms HDSA and Human Response in completion, and ties 94% with HDSA in readability while underperforming Human Response. Overall speaking, MARCO is superior to HDSA and comparable with Human Response. We further analyzed the bad cases of our model in readability and found that our model slightly suffers from token level repetition, a problem that can be solved by methods like the coverage mechanism (Mi et al., 2016; Tu et al., 2016). In completion, our model can understand the users' need and tends to provides them more relevant information, so that they can finish their goals in shorter turns.

We present two examples in Figure 8. In the first example, the user requests the hotel type while HDSA ignores it. The user requests to book one ticket in the second example, yet both HDSA and Human Response ask about the number once again. In contrast, our model directly answers the questions with correct information. To sum up, MARCO successfully improves the dialogue system by generating relevant and informative responses.

Example 1	
User:	I'm trying to plan a trip there but need a cheap place to stay.
	. . .
System:	Finches Bed and Breakfast is cheap.
User:	Okay. Sounds good. Can you tell me the type of hotel and provide me with the hotel's phone number?
Results	
Reference:	Finches is a guesthouse, and their number is 012***53. Can I help with anything else today?
HDSA:	The phone number for Finches is 012***53.
MARCO:	Finches is a guesthouse, and their number is 012***53.
Example 2	
User:	I am looking for a train that leaves on Thursday going to Cambridge.
	. . .
System:	TR1149 departs London Kings Cross at 17:17 and arrives in Cambridge at 18:08. Will that work?
User:	Yes it will. Can you book that for just me and provide the reference number?
Results	
Reference:	Sure thing! How many tickets do you need?
HDSA:	How many tickets would you like me to book for you?
MARCO:	I was able to book one ticket for you. Your reference number is R57G4DU4.

Figure 8: Two examples to show that MARCO successfully improves the dialogue system by generating relevant and informative responses.

8 Conclusion

In this paper, we presented a novel co-generation model for dialogue act prediction and response generation in task-oriented dialogue systems. Unlike previous approaches, we modeled act prediction as a sequence generation problem to exploit the semantic structures of acts and trained it jointly with response generation via dynamic attention from response generation to act prediction. To train this joint model, we applied an uncertainty loss for adaptive weighting of the two tasks. Extensive studies were conducted on a large-scale task-oriented dataset to evaluate the proposed model, and the results confirm its effectiveness with very favorable performance over several state-of-the-art methods.

Acknowledgments

References

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Ultes Stefan, Ramadan Osman, and Milica Gašić. 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)*.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3696–3709, Florence, Italy. Association for Computational Linguistics.

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.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792, Brussels, Belgium. Association for Computational Linguistics.

Zhuoxuan Jiang, Ziming Huang, Dong Sheng Li, and Xianling Mao. 2019. Dialogact2vec: Towards end-to-end dialogue agent by multi-task representation learning. *arXiv preprint arXiv:1911.04088*.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016a. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*, pages 1192–1202.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202.

Bing Liu and Ian Lane. 2017. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, pages 2506–2510. ISCA.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. *arXiv preprint arXiv:1907.10016*.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation. *arXiv preprint arXiv:1605.03148*.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Jiahuan Pei, Pengjie Ren, and Maarten de Rijke. 2019. A modular task-oriented dialogue system using a neural mixture-of-experts. *arXiv preprint arXiv:1907.05346*.

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*, pages 376–384.

Iulian V Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, et al. 2017. A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*.

Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates. 2018. Cold fusion: Training seq2seq models together with language models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*.

Pei-Hao Su, Paweł Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 147–157.

Shang-Yu Su, Kai-Ling Lo, Yi-Ting Yeh, and Yun-Nung Chen. 2018. Natural language generation by hierarchical decoding with linguistic patterns. 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 61–66, New Orleans, Louisiana. Association for Computational Linguistics.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. *arXiv preprint arXiv:1601.04811*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, 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*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve Young. 2017. Latent intention dialogue models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3732–3741. JMLR. org.

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 (Volume 1: Long Papers)*, pages 665–677.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 808–819.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. 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 1208–1218.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive dialogue state tracker. *arXiv preprint arXiv:1805.09655*.