Structured and Natural Responses Co-generation for Conversational Search

Chenchen Ye
National University of Singapore
chenchenye.ccye@gmail.com

Lizi Liao*
Singapore Management University
liaolizi.llz@gmail.com

Fuli Feng
University of Science and Technology of China, fulifeng93@gmail.com

Wei Ji
Sea-NExT Joint Lab,
National University of Singapore
jiwei@nus.edu.sg

Tat-Seng Chua
Sea-NExT Joint Lab,
National University of Singapore
dcscts@nus.edu.sg

ABSTRACT

Generating fluent and informative natural responses while maintaining representative internal states for search optimization is critical for conversational search systems. Existing approaches either 1) predict structured dialog acts first and then generate natural response; or 2) map conversation context to natural responses directly in an end-to-end manner. Both kinds of approaches have shortcomings. The former suffers from error accumulation while the semantic associations between structured acts and natural responses are confined in single direction. The latter emphasizes generating natural responses but fails to predict structured acts. Therefore, we propose a neural co-generation model that generates the two concurrently. The key lies in a shared latent space shaped by two informed priors. Specifically, we design structured dialog acts and natural response auto-encoding as two auxiliary tasks in an interconnected network architecture. It allows for the concurrent generation and bidirectional semantic associations. The shared latent space also enables asynchronous reinforcement learning for further joint optimization. Experiments show that our model achieves significant performance improvements.

CCS CONCEPTS

• **Information systems** → **Users and interactive retrieval**.

KEYWORDS

conversational search, co-generation, bidirectional association

ACM Reference Format:

Chenchen Ye, Lizi Liao, Fuli Feng, Wei Ji, and Tat-Seng Chua. 2022. Structured and Natural Responses Co-generation for Conversational Search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3477495.3532063

*Corresponding author.

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07...$15.00
https://doi.org/10.1145/3477495.3532063

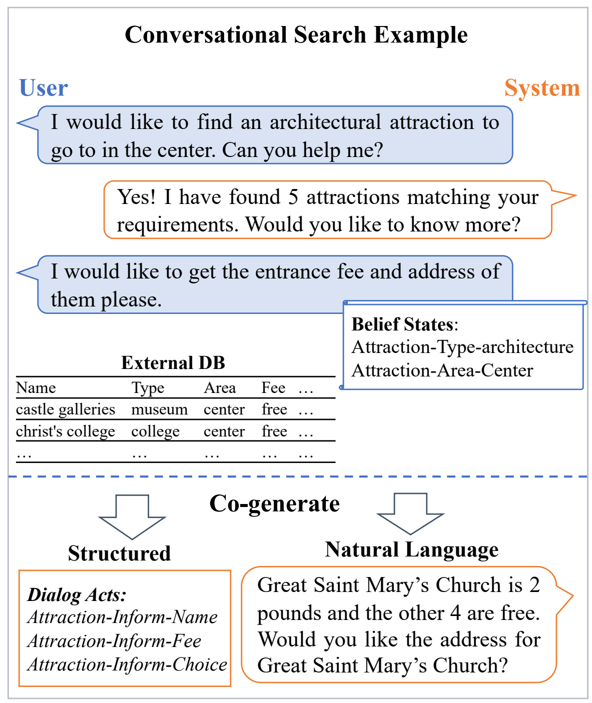


Figure 1: An example of conversational search where both the structured dialog acts and natural responses are crucial.

1 INTRODUCTION

With the rise of various personal assistants, conversational search has received a surge in popularity and attention [1, 10, 23, 28, 32]. Such systems aim to facilitate people with services such as hotel or restaurant booking through natural language conversations [5, 6, 17]. Different from the open-domain dialogues, its ultimate goal is to provide satisfactory natural responses to the end users and generate structured representations such as dialog acts for internal search optimization. For instance, Figure 1 shows a conversation segment, from which we can notice that both the structured dialog acts and natural language responses are essential for building an effective and efficient conversational search system [36, 45].

Traditionally, conversational response generation is conducted as a pipeline with multiple modules one after another. A standard architecture for such methods generally decompose the task into several subtasks, namely natural language understanding, state tracking, policy learning and natural response generation. Basically, natural language responses are generated based on formerly predicted dialog acts [15, 27, 41, 51]. However, such methods suffer from the error accumulation problem. For instance, errors occurred in dialog act prediction will mislead the following natural response generation. Also, such sequential modeling forces the semantic associations between structured dialog acts and natural responses to flow in only one direction. The heuristic or clue gained from the latter is unable to benefit dialog act prediction as a former stage.

Another group of research efforts directly learn the mapping between conversation context to natural language response in an end-to-end manner [26, 38, 52]. For example, Li et al. [21] leveraged an encoder-decoder RNN to define a policy over an infinite action space consisting of all possible utterances and optimize it via reinforcement learning (RL) with dialogue simulations [19, 34]. Zhao et al. [52] narrowed down the action spaces as latent variables and Wang et al. [38] further modeled the hierarchical structure with the option framework [35]. Such end-to-end methods explicitly ignore the structured dialog acts while resort to latent vectors at ease. Although such manner makes the context-to-response modeling more focused, the latent vectors learned are hard to use or interpret. They fail to generate structured responses such as dialog acts for search optimization, *e.g.* query execution or error debugging.

To overcome these, we aim to co-generate structured dialog acts and natural language responses at the same time. A viable way is to treat dialog act prediction as another sequence generation task and combine it with the original natural response generation as two decoder branches in multitask learning. Wang et al. [39] followed such structure and further designed dynamic attention to guide response generation with attended dialog acts. However, the interrelationships among dialog acts and natural responses are only modeled locally. The required synchronization between the two decoder branches is hard to realize due to varied vocabularies and sequence lengths. Moreover, the generators only focus on the responses in the current turn while fail to foresee the future in conversational search, hence might result in short-sighted responses.

In this work, we thus propose a neural Co-Generation model (**Co-Gen**) to realize global bidirectional semantic associations between the two forms of responses and reduce errors via farsighted optimization. Generally speaking, we construct a shared latent space for various forms of response generation and shape it with two informed priors. Specifically, we treat auto-encoding of structured dialog act and natural response as two separate auxiliary tasks. It inherently captures the generative factors of these responses while the semantic associations between them are globally matched via KL divergence. This allows for concurrent generation of the two forms of responses but does not require synchronization operations as the regularization works in a global way. Moreover, with the shared latent space, we manage to conduct asynchronous reinforcement learning smoothly to foresee the future for better search completion and success rates.

The contributions of this work can be summarized as follows:

- We design an interconnected network to co-generate structured and natural responses that allows for bidirectional semantic associations. Being regularized by auto-encoding priors, the network learns meaningful semantic space.
- We optimize the shared latent space of the variational model using asynchronous reinforcement learning. The task completion rewards promote more intelligent responses.
- Experiments show that the proposed Co-Gen significantly outperforms several state-of-the-art models both on automatic evaluation metrics and human evaluation, leading the board on the total performance in official records [1].

2 RELATED WORK

Researchers have continuously worked on many crucial problems of conversational search. Dalton et al. [7, 8] created benchmarks that track important information in dialogue context and then perform retrieval and ranking processes over candidate responses. Many methods explored this task setting, *e.g.*, to rank related information for current conversational answer by reasoning over a word proximity network [16], or to construct modular components for conversational ranking, including utterance rewriting, related candidate passage retrieval and re-ranking [28]. In this work, we instead focus on the generation angle. In general, structured dialog acts prediction and natural language response generation are closely related in the research of dialogue systems. Hence, our work relates to both the traditional pipe-lined methods and the popular end-to-end based methods. We use the term "end-to-end" throughout this paper to emphasize methods that do not require any middle labels or generate any intermediate results such as dialog acts.

2.1 Pipe-lined Response Generation

Pipe-lined systems typically require several separated modules: natural language understanding to extract user's intents (*e.g. inform*) and slot values (*e.g. area-center*), state tracking to update belief states [22, 24, 49], policy module to decide the system's next action [50], and natural language generation (NLG) to generate natural language responses. Such separated modules are trained independently with different supervision. More importantly, the policy module is conducted first and then used for the later response generation [25, 41, 51] to solve problems more efficiently [20, 21]. A classic solution employs reinforcement learning (RL) to learn a dialog policy that models the optimal action distribution conditioned on the dialogue state [42]. For traditional modular systems, the action space is defined by hand-crafted semantic representations such as dialog acts [4]. However, it requires that the entire action space can be hand-crafted [33], which cripples a system's ability to handle complex conversations. More importantly, the errors occurred in the policy module will inevitably be accumulated to the later response generation module, which might be misled by the noisy or wrong inputs. Also, such pipe-lined generation manner forces the semantic associations between dialog acts and natural responses to transmit in only one direction, that is from dialog acts to natural response. Models will not be able to leverage knowledge learned from natural response generation to guide the corresponding dialog act prediction.

Another line of research efforts rely on language modeling to generate middle results sequentially and are rather close to pipelinelined methods. These models largely benefit from the large pretrained transformer-based models such as BERT [11] and GPT-2

[1] https://github.com/budzianowski/multiwoz

[31]. By connecting dialogue context, intermediate results and response into a long sequence, such systems typically rely on language modeling techniques to directly optimize the data likelihood while neglecting the strategy planning altogether [15, 30, 44].

2.2 End-to-end Response Generation

Research in end-to-end response generation is largely inspired by the success of sequence-to-sequence modeling for chit-chat systems. Under this line, various kinds of sequence-to-sequence models have played their important parts in the response generation task, from basic neural recurrent models such as RNN, Bi-LSTM, hierarchical recurrent encoder-decoder to advanced ones such as DialoGPT [48] and prompt-learning [13]. We see great success in applying these models for open domain conversations or chit-chat, however, it is a non-trivial task to transfer such success to goal-oriented application scenarios such as conversational search.

In the closely connected task-oriented dialogue research community, researchers have managed to combine sequence-to-sequence models with reinforcement learning using task completion rewards [46, 47]. Initially, the action space for RL is generally defined as the entire vocabulary such as in [21] where every response output word is considered to be an action selection step. It blows up the size of action space hence the trajectory length, which easily leads to slow and sub-optimal convergence [9, 14]. Therefore, Zhao et al. [52] proposed to construct a latent space between the context encoder and the response decoder as the action space. Better performance is obtained. Lubis et al. [26] further leveraged auxiliary tasks to shape the latent variable distribution and Wang et al. [38] modeled the hierarchical structure between the dialogue policy and NLG with the option framework [35]. Better response results have been achieved for these methods partly because they make the learning of 'context-to-natural response' mapping more focused. However, these methods fail to generate structured dialog acts for search optimization. Their latent variables learned are hard to interpret or use under a specific search setting. For example, when something went wrong in the generated natural language responses, it is hard to check the reason relying on latent vectors, while structured dialog acts might give us some useful hints on the contrary. Also, structured dialog acts provide us more convenient information pieces to optimize the search interaction flow, *e.g.* forming queries.

We hence resort to generate structured dialog acts and natural responses at the same time to support these functions. Although there are few studies tried to work in this direction such as [39], our work overcomes several limitations such as local semantic associations between various response forms and short-sighted generation *etc.*

3 METHOD

We formally introduce the response co-generation task and our proposed Co-Gen approach as shown in Figure 2. Formally, let $S_t = \{u_1, r_1, \cdots, u_{t-1}, r_{t-1}, u_t\}$ denotes the conversation history at turn t, where u_i and r_i are the i-th turn user and system utterance respectively. We represent the conversation context C_t as the combination of window sized S_t and the state vector [2]. The objective of the co-generation task is to generate the dialog acts sequence $a_t = x_1, x_2, \cdots, x_m$ of m words and a natural language response $r_t = y_1, y_2, \cdots, y_n$ of n words based on the context.

Generally speaking, the proposed Co-Gen model works as an encoder-decoder framework under the multitask learning setting as shown in Figure 2. The dialog act decoder and the natural response decoder share the latent space z, which is shaped by calculating KL-divergence with the informed priors learned via auto-encoding tasks. Such latent vector z also enables the adoption of reinforcement learning for further optimization. For more details, we will first give some preliminaries about the general end-to-end RL framework for response generation which is closely related to our design. Then, the shared latent space is introduced with our auto-encoding schemes. Following the description of the co-generation branches for different response forms, we further give details about the asynchronous reinforcement learning.

3.1 Preliminaries

Since the optimization framework is closely related to the popular end-to-end RL framework [52], we first introduce some preliminary concepts here. In such framework, the response generation is typically performed in two steps: supervised learning (SL) pretraining and RL finetuning. In the SL step, the model learns to generate a response $\boldsymbol{r}$ based on the observed conversation context $\boldsymbol{c}$. It updates the network parameters π to maximize the log likelihood of the whole training data:

$$L_{SL} = \mathbb{E}_{\boldsymbol{r},\boldsymbol{c}}[\log p_\pi(\boldsymbol{r}|\boldsymbol{c})]. \tag{1}$$

After achieving a good parameter setting $\hat{\pi}$ via SL, the RL step starts from it and further updates the model parameters *w.r.t.* the task-specific goal, reflected as a reward. The RL steps usually uses policy gradients, *e.g.* the REINFORCE algorithm [43]. Suppose a dialogue has T turns, for a specific time-step t, the discounted return is defined as $O_t = \sum_{i=t}^{T} \gamma^{i-t} o_i$, where o_t is the immediate reward for turn t and $\gamma \in [0, 1]$ is the discount factor. During fine-tuning, the model tries to maximize the expected return from the first time-step onward. The mathematical formulation for such expected return is hence $J = \mathbb{E}[\sum_{t=0}^{T} \gamma^t o_t]$.

3.1.1 *Word-level RL.*

For the RL step, the initial word-level methods treat every output word as an action step hence the policy gradient is calculated as:

$$\nabla J(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{T}\sum_{j=0}^{|r_t|} O_{tj} \nabla_\pi \log p_\pi(w_{tj}|w_{<tj}, \boldsymbol{c}_t)],$$

where j is the index of each token w in the response r_t and O_{tj} denotes the discounted return of the j-th token at turn t. In this policy gradient form, the action space is the vocabulary size of the system $|V|$, and the trajectory length is $\sum_{t=0}^{t=T} |r_t|$, making RL in this space extremely challenging.

3.1.2 *Latent Action Space RL.*

Hence, researchers in [26, 38, 52] introduce a latent variable $\boldsymbol{z}$ to factorize the conditional distribution into $p(\boldsymbol{r}|\boldsymbol{c}) = p(\boldsymbol{r}|\boldsymbol{z})p(\boldsymbol{z}|\boldsymbol{c})$. By treating the latent space $\boldsymbol{z}$ as the action space, correspondingly, the policy gradient becomes:

$$\nabla J(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{T} O_t \nabla_\pi \log p_\pi(z_t|\boldsymbol{c}_t)]. \tag{2}$$

[2] Following the popular setting in [3, 38, 39], we consider context window 2 for dialogue history and use the ground truth state vector because state tracking is not our focus.

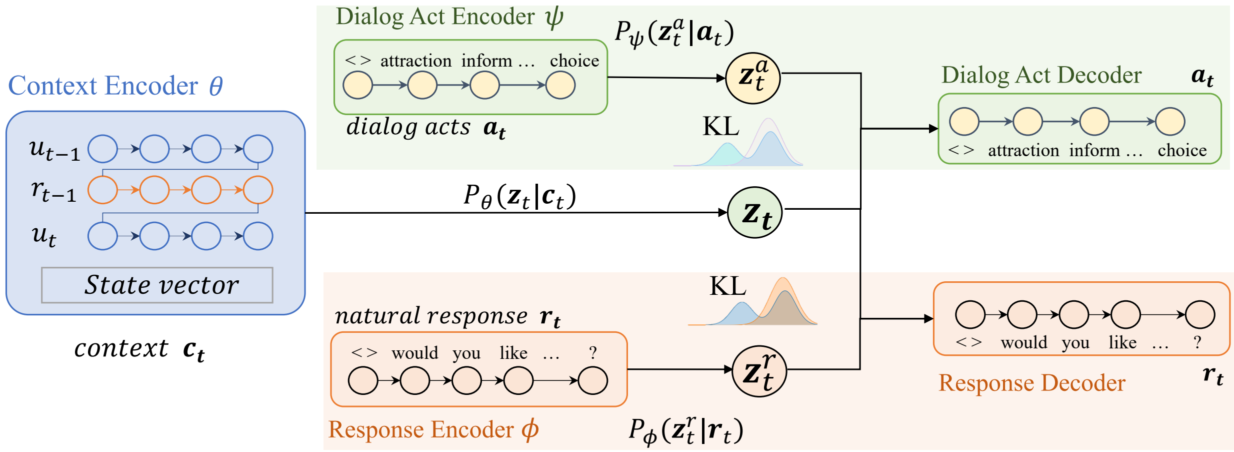


Figure 2: Architecture of the proposed Co-Gen for dialog act and natural response co-generation, where act and response generators share the same latent space z. Two auto-encoding branches (shaded in green/orange) are treated as auxiliary tasks to regularize the latent space and capture generative factors of responses. The encoder ϕ and ψ will be disabled during testing.

Note that by introducing such latent variable, the size of action space and the trajectory length are largely reduced. Also, the policy gradient only focuses on the encoder part π_e while the decoder π_d stays intact.

3.2 Shared Latent Space

In our approach, we also maintain such a latent variable space. Specifically, we construct a context encoder network with parameters θ, which corresponds to $p_\theta(z|c)$. We first encode the concatenated conversation history S_t via bidirectional RNN network with GRU cells and global type attention mechanism:

$$s_t = \text{Bi-GRU}(\ S_t\ ).$$

Then, we obtain the context representation $\boldsymbol{c}_t$ as $\boldsymbol{c}_t = [\boldsymbol{s}_t; \boldsymbol{d}_t]$ where $\boldsymbol{d}_t$ is the oracle state vector following the popular setting in [3, 38, 39] and [;] denotes the concatenation operation.

Similar to [26], we assume that the latent variable z is sampled from a multivariate Gaussian distribution. Hence, we have $p_\theta(z|\boldsymbol{c}) = \mathcal{N}(z|\boldsymbol{\mu}, \Sigma)$. It can be implemented via the reparameterization trick [18] with feed-forward neural networks:

$$\boldsymbol{\mu}, \Sigma = MLP(\boldsymbol{c}_t),$$
$$\epsilon \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma),$$
$$z = MLP(\epsilon),$$

where ϵ is a context dependent random noise. It is drawn from a normal distribution whose mean and covariance matrix are computed from $\boldsymbol{c}_t$ with feed-forward neural networks.

The latent space is shared with the dialog act encoder and the natural response encoder as shown in Figure 2. Basically, we have two auto-encoding streams: $\boldsymbol{a} \rightarrow z^a \rightarrow \boldsymbol{a}$ and $\boldsymbol{r} \rightarrow z^r \rightarrow \boldsymbol{r}$. For example, in the former stream, we chose dialog acts sequence auto-encoding as the auxiliary task using the variational auto-encoding (VAE) model. That is, given a sequence $\boldsymbol{a}$ we train the model to reconstruct the sequence via a latent variable z^a between the encoder and decoder (green shaded box in Figure 2). With an uninformed prior $p(z)$, the pre-training objective for network parameters ψ is:

$$L_{VAE}(\psi) = \mathbb{E}_{q_\psi(z^a|\boldsymbol{a})}[\log p_\psi(\boldsymbol{a}|z^a)] - D_{KL}[q_\psi(z^a|\boldsymbol{a})||p(z^a)],$$

where $q_\psi(z^a|\boldsymbol{a})$ is the posterior. VAE models have been shown to be able to capture generative aspects of the samples they are trained on, resulting in good interpolation between latent variables [18]. By training a VAE on dialog acts sequences, we aim to capture global generative aspects of dialog acts such as intent and domain information in an unsupervised manner.

Similarly, the pre-training objective for network parameters ϕ for the $\boldsymbol{r} \rightarrow z^r \rightarrow \boldsymbol{r}$ stream (orange shaded box in Figure 2) is:

$$L_{VAE}(\phi) = \mathbb{E}_{q_\phi(z^r|\boldsymbol{r})}[\log p_\phi(\boldsymbol{r}|z^r)] - D_{KL}[q_\phi(z^r|\boldsymbol{r})||p(z^r)].$$

In this stream, we aim to capture global generative aspects of natural responses in an unsupervised way.

To sum up, we leverage two auto-encoding streams for dialog acts sequence and natural response respectively and learn two VAE latent spaces z^a and z^r in an unsupervised manner. We propose to utilize the VAE latent spaces z^a and z^r to regularize the latent space z learned from conversation context, making a shared latent space. Hence, we will use the learned posteriors $q_\psi(z^a|\boldsymbol{a})$ and $q_\phi(z^r|\boldsymbol{r})$ as informed priors for co-generation.

3.3 Co-generation Branches

With shared latent space introduced, we now describe the co-generation mechanism. Generally speaking, we calculate two branches: $\boldsymbol{c} \rightarrow z \rightarrow \boldsymbol{a}$ and $\boldsymbol{c} \rightarrow z \rightarrow \boldsymbol{r}$, where each generation branch is guided by the corresponding learned informed prior globally.

In $\boldsymbol{c} \rightarrow z \rightarrow \boldsymbol{a}$, given a training dataset of $\{\boldsymbol{c}, \boldsymbol{a}\}$ pairs, the base optimization method is via stochastic variational inference by maximizing the evidence lowerbound (ELBO) on the data log likelihood:

$$L_{full}(\theta) = p_{q(z|\boldsymbol{a},\boldsymbol{c})}(\boldsymbol{a}|z) - D_{KL}[q(z|\boldsymbol{a},\boldsymbol{c})||p_\theta(z|\boldsymbol{c})],$$

where $p(\boldsymbol{a}|z)$ is realized by the dialog act decoder and $q(z|\boldsymbol{a},\boldsymbol{c})$ is approximated by training a neural encoder network. However, there is a major limitation here: it suffers from exposure bias at latent space, *i.e.* the decoder only sees z sampled from $q(z|a,c)$ and never experiences z sampled from $p_\theta(z|c)$, which is always used at testing time. Therefore, we use the 'lite' version proposed in [52]:

$$L_{lite}(\theta) = p_{q(z|c)}(a|z) - \beta D_{KL}[q(z|c)||p(z)],$$

where β is a hyper-parameter between 0 and 1. In this version, the posterior network becomes the same as our context encoder $p_\theta(z|c)$. To regularize the latent space, we directly use the informed prior learned from the auto-encoding pre-training for dialog acts sequence: $q_\psi(z^a|a)$. Hence, the final objective for dialog acts sequence generation is:

$$L_{lite\_A}(\theta) = p_{p_\theta(z|c)}(a|z) - \beta \cdot D_{KL}[p_\theta(z|c)||q_\psi(z^a|a)].$$

Similarly, in $c \to z \to r$, we have the regularized objective for natural response generation as:

$$L_{lite\_R}(\theta) = p_{p_\theta(z|c)}(r|z) - \alpha \cdot D_{KL}[p_\theta(z|c)||q_\phi(z^r|r)],$$

where α is another hyper-parameter between 0 and 1. Note that the two informed priors $q_\psi(z^a|a)$ and $q_\phi(z^r|r)$ are both aligned with the same $p_\theta(z|c)$. We actually push them to be similar to each other indirectly. In this way, we manage to encourage bi-directional semantic associations between them globally.

3.4 Asynchronous Reinforcement Learning

With the aforementioned network structure as described in Figure 2, we can train a relatively good parameter setting using supervised learning. Now starting from a well-trained SL checkpoint, we further fine-tune it via RL to obtain better results.

Specifically, we apply reinforcement learning on the shared latent space z. The policy gradient is calculated as Equation 2. As discussed, the policy gradient only works on the encoder part π_e which corresponds to our context encoder θ. We thus further apply REINFORCE to optimize our decoder in an asynchronous manner. For example, in the $z \to r$ branch, we apply GRU cells to decode word by word sequentially. Denoting the decoder as G, it is responsible for transforming z into the detailed response sequence r. By treating every output token as an action step, the policy gradient is defined as:

$$\nabla J(\mathrm{G}) = \mathbb{E}_\mathrm{G}[\sum_{t=0}^{T}\sum_{j=0}^{|r_t|}\mathcal{R}_{tj}\nabla \log \mathrm{G}(w_{tj}|w_{<tj}, z_t, c_t)], \qquad (3)$$

where $|r_t|$ is the number of tokens in the response at turn t and j is the token index in the response. $\mathcal{R}_{tj}$ denotes the discounted return of the j-th token at turn t. Note that here the Equation further incorporates BLEU score as part of the reward. This is different from the Equation 2 using only task completion rewards.

The goal of the whole reinforcement learning process is to find the best maximizers that can maximize the reward value regarding both encoder part and the decoder part. The two policies are defined in Equation (2) and (3), respectively. If we synchronously update these two policies, the composite state will be inconsistent before and after the update each time. Consequently, the value does not always monotonically improve during the learning process. It will affect the convergence of both policies. Therefore, we update the two asynchronously during learning. Experiments show that this leads to the convergence of these policies to a local maximizer.

4 EXPERIMENTS

4.1 Datasets

We conduct experiments on the most widely used conversational search benchmark datasets MultiWoz 2.0 [2] and MultiWoz 2.1 [12] to evaluate our proposed co-generation model. It contains over ten thousand dialogues that spans over seven distinct domains. All the conversations are collected by human-to-human conversations via the crowdsourcing WOZ setting. In which, every conversation is generated where the user is given a pre-defined goal and the system attempts to fulfill the goal by interacting with the user. We follow the same delexicalized method provided by [2] to preprocess the dataset, which is widely applied in other works [38, 52]. MultiWoz 2.1 is a modified version of MultiWoz 2.0 which mainly fixes the noisy state annotations and corrects a small fraction of conversation utterances. We follow the public divisions to split the datasets into training, validation and testing sets [2, 12].

4.2 Training Details

Here we list the specific hyper parameters for Co-Gen model. We set both the maximum length for the user's utterance in context and the maximum length for the system's utterance in response to 50. The embedding size for each word is set to 100. For the input conversation history, target dialog acts sequence and natural response, all these encoders are one-layer bidirectional RNN that uses GRU cells of size 300. The encoded result is projected to the latent content space, where the size of the shared latent variable z is 200. For decoders, they are also one-layer RNN with a separate embedding layer and GRU cells.

During SL training, we set batch size as 32 and the maximum number of training epochs as 50. Adam optimizer is used with an initial learning rate of 0.001 and weight decay 1e-05. The KL divergence hyper-parameters α and β are all set to be 1.0 empirically in our experiments. After supervised training of the model, we further fine-tune the model with asynchronous RL. In which, each conversation is evaluated with the goal (*e.g.* calculating the *Success* rate) and the *BLEU* scores. We fix each batch as a complete dialogue. Stochastic gradient descent (SGD) is used, and the learning rate in the asynchronous optimization process for the two policies is both 0.09 with weight decay 1e-05. Generally speaking, the experiments of Co-Gen were run on a Nvidia GeForce RTX 2080Ti graphic card, which consumed around 2.5 hours for SL training and less than 1 hours for RL finetuning. Hence, it is not very expensive to reproduce our results as shown in Table 1.

4.3 Test Settings

Experiments are conducted on the context-to-response generation task similar to the one originally proposed in [2]. Given the conversation context, the model is trained to generate appropriate responses in each turn. In our proposed model, both the structured dialogue acts and the natural language response will be generated and evaluated. What's more, as the proposed model constructs a shared latent space and leverages the auto-encoding scheme to further regularize the space, we will also demonstrate whether the learned semantic space is meaningful or not.

4.4 Evaluation Metric

Following existing works, we adapt three automatic metrics measured in percentage to evaluate the generated natural responses from a conversational search system such as *Inform* rate, *Success* rate and *BLEU* score. *Inform* rate measures whether the system has provided the correct entity (*e.g.*, the name of restaurant). *Success* rate shows the ratio of correct answers provided for request slots in the generated utterances. The fluency of the generated response is measured by *BLEU* [29] score. The combined *Score* is computed as $(BLEU + 0.5 \times (Inform + Success))$ [2] to fairly evaluate the performance of a dialogue system as a popular total score. To evaluate the generated structured dialog acts, we adopt Entity F1 [41] to evaluate the entity coverage accuracy (including all slot values, days, numbers, and reference, etc). We also adopt the Act F1 from [3, 39] to judge the act coverage accuracy such as domain, action and slots.

Furthermore, we carry out human evaluation to measure the quality of generated responses. We add three criteria as follows: *Fluency* measures whether the generated response is fluent, grammatically correct and smooth; *Coherence* reflects how coherent the response is and whether it follows the conversation flow; *Informativeness* shows whether the response provides relevant and useful information to the user. We provide detailed descriptions to the human evaluators during evaluation.

4.5 Baseline Models

We denote the variation of the proposed Co-Gen as Co-Gen$_{_SL}$, which corresponds to the proposed model without RL finetuning. They are compared with the following models: SFN [27], UBAR [44], HDSA [40], DialoGPT [48], LaRL [52], LAVA [26] , HDNO [38] and MarCo [39]. These baselines can be organized into three groups, *i.e.*, pipe-lined, end-to-end and co-generation methods. All these models leverage oracle dialogue states. More details about these methods are given below:

– **SFN** [27]: It learns neural dialogue modules corresponding to the structured components of traditional dialogue systems. It obtains strong results both with (denoted as SFN) and without reinforcement learning (SFN$_{_SL}$).
– **UBAR** [44]: It is based on fine-tuning the pretrained GPT-2 where a whole dialogue session is treated as a single training sequence. The sequence is composed of user utterance, belief state, database result, system act, and system response of every dialog turn.
– **HDSA** [3]: It is a two-stage model that uses BERT to predict a one hot dialog act vector for guiding the following response generation task. The structure of dialog acts is modeled as a multi-layer hierarchical graph.
– **DialoGPT**[48]: It is also based on fine-tuning the pretrained GPT-2 but it only focuses on the context-to-response mapping where all middle labels are ignored.
– **LaRL** [52]: This model is the first to represent dialogue act as latent vectors in task-oriented dialogues. During the RL training, it only updates the corresponding policy part while the decoding part is not involved in RL training.
– **LAVA** [26]: Built upon LaRL, it further leverages three auxiliary tasks to shape the latent variable distribution, making the latent representations truly encodes the characteristics of different actions.
– **HDNO** [38]: It adopts the option framework [35] to model the hierarchical relation between dialogue policy and NLG. No middle results are required. We also report its SL only version as HNDO$_{_SL}$.
– **MarCo** [39]: It co-generates dialog acts and natural responses as two sequence generation tasks while uses dynamic attention to capture local semantic associations.

4.6 Automatic Evaluation Results

4.6.1 Results on Natural Language Responses. The main results for natural response generation are shown in Table 1. The proposed Co-Gen method achieves the best performance regarding the overall performance reflected by the combined *Scores.* It significantly outperforms all the comparing methods. For example, Co-Gen improves the best performing baseline HDNO by 2.17% in the MultiWoz 2.0 dataset and 1.31% in the MultiWoz 2.1 dataset. Also, Co-Gen shows balanced results over both the strategy learning for task completion and surface style realization. The former is evaluated by *Inform* rate and *Success* rate, while the latter is evaluated by the *BLEU* score. Specifically, we observe a general trend that RL applied methods can largely boost the strategy part as expected, because the task completion rates are directly considered as rewards. For example, in pipeline-based methods, the RL applied SFN outperforms its SL counterparts, especially in task completion metrics like *Inform* rate and *Success* rate. This is also true in end-to-end RL based methods such as HDNO. Hence, in the proposed Co-Gen model, RL rewards are also incorporated, which enables the model to fore-see more turns in a trial-and-error manner. This also makes it possible for the model to generate more intelligent responses. Moreover, we observe that for RL-applied methods, overemphasizing task completion may often lead to corrupted responses. For example, on MultiWoz2.0, though LAVA achieves a leading high score in *Inform* and *Success* rate, it only obtains a low *BLEU* score of 12.02. In comparison, Co-Gen obtains more balanced results with reasonably high *Inform* and *Success* rate and more natural responses reflected by its 20.42 *BLEU* score. For methods that do not apply RL such as MarCo and HDSA, their performance results are relatively lower, especially regarding the task completion metrics.

Moreover, for language modeling based methods, the results show that how to model the task is the key to achieve good performance via the powerful large-scale pretraining models such as GPT-2. For example, the DialoGPT model does not perform well and there are large performance gaps between DialoGPT and UBAR. Because the modeling of the later is more for task-oriented dialogues and various intermediate labels are involved. Although UBAR manages to achieve high combined *Scores*, the lack of foreseeing the future is still a main shortcoming for such models.

4.6.2 Results on Structured Responses. Co-Gen also manages to generate structured dialogue acts sequences, where each action is organized as a *(domain, action, slot)* tuple. We separately check Act F1 score for dialog acts and the Entity F1 for the slot values. The results on MultiWoz 2.0 are listed in Table 2, where BiLSTM, Word-CNN and Transformer are baselines from [3]. There are mainly three groups of methods. The first group 'Act Prediction Only' means

Table 1: Overall natural language response results on MultiWoz 2.0 and MultiWoz 2.1.

Group	Method	MultiWoz 2.0				MultiWoz 2.1			
		Inform	Success	BLEU	Score	Inform	Success	BLEU	Score
Pipe-lined	SFN_$_{SL}$	90.00	74.20	18.35	100.45	63.10	53.10	17.56	75.66
	SFN	94.40	83.10	16.34	105.09	87.80	76.20	10.57	92.57
	UBAR	94.00	83.60	17.22	106.02	89.6	78.6	17.34	101.44
	HDSA	82.90	68.90	23.60	99.50	86.30	70.60	22.36	100.81
End-to-End	DialoGPT	73.40	48.00	12.16	72.86	72.10	50.10	12.62	73.72
	LaRL	93.49	84.98	12.01	101.25	92.39	85.29	13.72	102.56
	LAVA	97.50	94.80	12.02	108.17	96.39	83.57	14.02	104.00
	HDNO_$_{SL}$	78.60	70.40	19.26	93.76	78.80	66.70	18.46	91.21
	HDNO	95.80	84.50	18.61	108.76	93.20	81.90	18.35	105.90
Co-Generate	MarCo	92.30	78.60	20.02	105.47	92.50	77.80	19.54	104.69
	Co-Gen_$_{SL}$	92.10	77.40	20.91	105.66	88.90	80.00	20.67	105.12
	Co-Gen (ours)	94.70	86.70	20.42	**111.12**	91.20	85.20	19.09	**107.29**

these methods are specifically trained to do dialog act prediction as classification task while cannot generate natural responses. The second 'Pipe-lined' group are all pipe-lined methods which generates both, but in sequential way. While 'Co-generate' means methods generate both concurrently. Since the end-to-end RL based methods do not have such outputs, we skip the comparison.

As shown in Table 2, methods in the 'Act Prediction Only' group obtain relatively low performances while the Transformer based one performs a bit better among them. As they only work on *(domain, action, slot)* tuples, they do not have Entity F1 results. For pipe-line methods, UBAR performs the best. This might be due to that it gains learning capability from the powerful GPT-2 model. However, its Entity F1 score is largely inflated, as it does not differentiate the values of different domains. For example, *hotel_area, restaurant_area, attraction_area* are same for UBAR but different for other models. Hence, the high score 82.3 is not comparable. Besides these, the proposed model Co-Gen works the best across Act F1 and Entity F1. These demonstrate that the proposed Co-Gen method not only generates good natural responses, but also generates the structured dialog acts well.

Table 2: Structured response generation results on MultiWoz 2.0. Note that the end-to-end group methods fail to generate such responses, hence no results are shown here.

Group	Method	Act F1	Entity F1
Act Prediction Only	BiLSTM	71.4	NA
	Word-CNN	71.5	NA
	Transformer	73.1	NA
Pipe-lined	SFN	63.7	77.1
	UBAR	84.6	82.3[3]
	HDSA	77.3	65.7
Co-Generate	MarCo	73.9	59.9
	Co-Gen	**87.6**	**77.2**

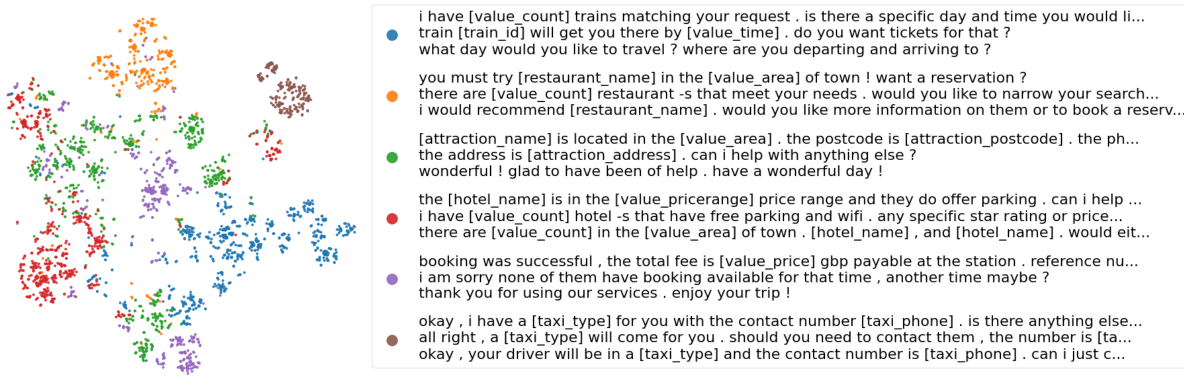
4.6.3 Semantic Meanings of the Shared Latent Space. As shown in Figure 3, we visually assess the latent content space by first clustering the latent content vector of each system response in the testing set into six clusters, and then projecting them with t-SNE [37] to analyze the formed clusters. Through inspecting the randomly selected system utterances as shown in the right hand side, we find that the clusters of latent content vectors of both Co-Gen and HDNO possess some semantic meanings. For example, the cluster in blue dots in Co-Gen is related to train booking and the cluster in yellow dots is related to restaurant recommendation, while the cluster in brown dots in HDNO is related to the general phrases for goodbye at the end of service. However, it is also obvious that the clusters from Co-Gen as shown in Figure 3 (a) are relatively better separated, which demonstrates clearer semantic meanings expressed by these latent vectors. This might be due to the successful shaping effect of the pre-trained informed priors from auto-encoding tasks for dialog acts sequence and natural response.

[3]The evaluation script of UBAR does not differentiate entity domains hence largely inflates the result. This number is thus not comparable to others.

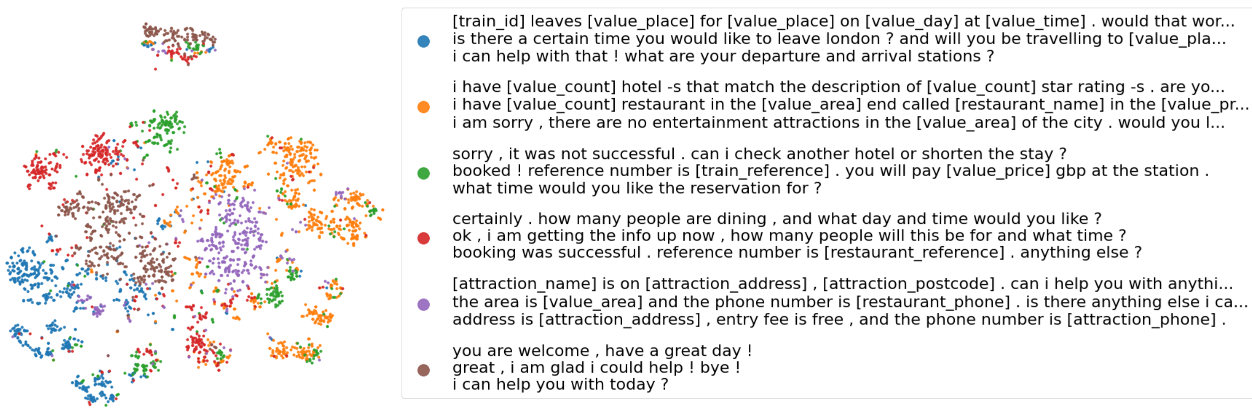
4.7 Human Evaluation

We conduct a human evaluation by recruiting eight undergraduate students as participants to perform two pairs of comparison from the human perspective between the responses generated by Co-Gen vs. HDNO and Co-Gen vs. MarCo. For each pair of response comparison, We randomly sample 150 samples from the testing dataset. During the evaluation, each conversation sample is presented to the participants with the user utterance, ground-truth response as the reference, and two generated responses from Co-Gen and the counterpart model separately. The participants are unaware of the source model for the generated responses to ensure a fair comparison. The ranking is based on three criteria: (i) *fluency*: the response is grammatically correct, natural, and smooth. (ii) *coherence*: the response is coherent and follows the flow of the dialogue reasonably. (iii) *informativeness*: the response provided related information to solve the user's requests and complete the task.

After gathering the replies, the calculated statistics are shown in Figure 4, where the "Win", "Tie", and "Lose" parts in the stacked bars represents the proportions of Co-Gen outperforms, ties with, and loses to its counterparts under each criterion. From the results, we can observe that the generated responses by Co-Gen outperforms HDNO in all three aspects, indicating its strong capability in correctly inferring system action to fulfill user requests and generating human-like responses. In the second comparison, we note that



(a) Co-Gen (ours)



(b) HDNO

Figure 3: The shared latent content vectors of Co-Gen and HDNO clustered in six categories visualized via the T-SNE algorithm. We randomly show three turns of system utterances for each cluster (best view in color).

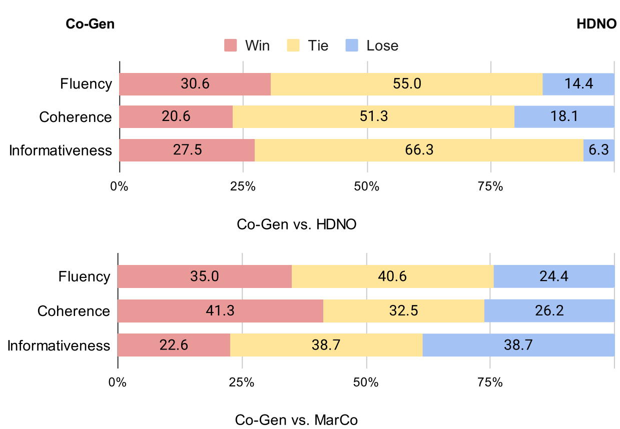


Figure 4: Results of human evaluation in response quality. Two pairs of comparison have been conducted, where the first figure presents the results of Co-Gen vs. HDNO, and the second figure corresponds to Co-Gen vs. MarCo.

Co-Gen obtains higher winning percentages than MarCo in both *fluency* and *coherence* while under-performing on *informativeness*. We further analyzed the bad cases of our model on informativeness in this pair comparison, and find that both model can provide enough information to solve user requests, while MarCo sometimes emphasizes some repeated information that has appeared in previous turns' responses. Such generation preferrence helps it to be more informative, but also results in token-level repetition and conversation level incoherence. In all, our model Co-Gen can deduce more accurate system actions that truly figure out the user request and provide an appropriate response accordingly.

4.8 Example Responses Generated

To get a better sense of how these methods perform, we showcase some system utterances generated in the same dialogue by different baselines and our proposed Co-Gen. Since most of the dialogues in MultiWoz 2.0 and MultiWoz 2.1 are similar, we only show some results on MultiWoz 2.0. As listed in Table 3, the generated utterances of Co-Gen is apparently more fluent and task completion oriented. For example, it manages to keep the whole dialogue on the topic of

Table 3: Some delexicalized responses generated by the baselines and Co-Gen on MultiWoz 2.0.

Method	Delexicalized Dialogue
User:	**i am looking to go somewhere in the [value_area] to enjoy some entertainment. can you help?**
GT Sys:	**i cannot find any entertainment venue -s in the [value_area]. would you like to try something different ? a museum perhaps?**
Co-Gen:	there are no colleges in the [value_area]. would you like to try a different area or attraction type?
MarCo:	[attraction_name] is a nightclub in the [value_area]. it s free to get in.
HDNO:	i am sorry, there are no swimming pools in the [value_area]. would you like me to look in a different area?
LAVA:	i am sorry, there are no multiple sports attractions in the [value_area]. would you like to try another area?
UBAR:	i am sorry, there are no [value_type] in the [value_area]. would you like to try a different area?
User:	**no museums, what colleges are there around town?**
GT Sys:	**sure, there are [value_count] colleges in that area.**
Co-Gen:	there are [value_count] colleges in the [value_area]. i recommend the [attraction_name]. it s free to get in.
MarCo:	there are [value_count] colleges in the [value_area]. [attraction_name] is located at [attraction_address] and has free admission.
HDNO:	[attraction_name] is a great place. would you like me to book a ticket for you?
LAVA:	there are [value_count] colleges in the [value_area]. i recommend [attraction_name]. would you like more information?
UBAR:	there are [value_choice] [value_type] in the [value_area]. [value_name] on [value_address] is [value_price] and [value_name] on [value_address] is [value_price].
User:	**can i please get the phone number, postcode and entrance fee?**
GT Sys:	**[attraction_name] is [value_count] i would recommend, their phone number is [attraction_phone], post code is [attraction_postcode], and there is no entrance fee. can i help you with anything else?**
Co-Gen:	[attraction_name] is a great college in the [value_area]. the phone number is [attraction_phone] and the postcode is [attraction_postcode].
MarCo:	[attraction_name] is located at [attraction_address], postcode [attraction_postcode], and their phone number is [attraction_phone]. the entrance fee is [value_count] pounds.
HDNO:	the phone number is [attraction_phone]. the postcode is [attraction_postcode]. is there anything else i can help you with?
LAVA:	the [attraction_name] is located at [attraction_address], postcode [attraction_postcode]. the phone number is [attraction_phone].
UBAR:	the phone number is [value_phone]. the postcode is [value_postcode] and the entrance fee is [value_price].
User:	**i also need a train that leaves on [value_day] and arrives by [value_time].**
GT Sys:	**where are you traveling from? there are [value_count] trains from [value_place] that arrive by that time.**
Co-Gen:	there are several trains that meet your criteria. where are you departing from and what is your destination?
MarCo:	there are [value_count] trains leaving from [value_place] to [value_place]. where would you like to depart from and arrive ?
HDNO:	where will you be departing from?
LAVA:	i have [value_count] trains that meet your criteria. would you like to narrow it down by departure site or arrival time?
UBAR:	where will you be departing from?
User:	**i am leaving [value_place] and going to [value_place].**
GT Sys:	**the first train leaves at [value_time].**
Co-Gen:	i have train [train_id] that leaves at [value_time] and arrives at [value_time]. would you like me to book it?
MarCo:	[train_id] leaves at [value_time]. would you like me to book that for you ?
HDNO:	[train_id] leaves at [value_time]. would you like me to book a ticket for you?
LAVA:	i have train [train_id] that leaves at [value_time] and arrives at [value_time]. would you like me to book that for you?
UBAR:	[value_id] leaves at [value_leave] and arrives at [value_arrive]. would you like me to book that for you?
User:	**can you book [value_count] tickets?**
GT Sys:	**your reservation has been made. the total fee is [value_price] gbp payable at the station. reference number is [train_reference].**
Co-Gen:	booking was successful, the total fee is [value_price] gbp payable at the station. reference number is [train_reference]. can i help with anything else?
MarCo:	i have booked you [value_count] seats on the [train_id]. the total fee is [value_price] gbp payable at the station. your reference number is [train_reference].
HDNO:	booking was successful, the total fee is [value_price] gbp payable at the station. reference number is [train_reference]. is there anything else i can help you with?
LAVA:	booking was successful, the total fee is [value_price] gbp payable at the station. reference number is [train_reference].
UBAR:	booking was successful, the total fee is [value_price] payable at the station. reference number is: [value_reference]. can i help you with anything else today?

college recommendation and successfully book train tickets for the user during the conversation.

In more details, for those methods fine-tuned with RL, Co-Gen is able to foresee the future better for response generation than other baselines such as HDNO and LAVA. This is evidenced by the response on 'no colleges' in the first turn, while other models such as HDNO and LAVA mention 'swimming pools' and 'multiple sports attractions' instead. This is also evidenced in the second turn that Co-Gen manages to generate 'free to get in' which corresponds to 'entrance fee' asked in the subsequent turn. Moreover, in comparison with other baselines trained with RL, the generated utterances of Co-Gen is more stringent in generating slots. Especially, LAVA tends to generate as many slots as possible to increase the success rate. *e.g.* generating the extra *[attraction_address]* in the third turn. This is the common issue of most RL methods on task-oriented dialogue system. However, in Co-Gen the situation is better. This might be due to our asynchronous RL optimization scheme where task completion goals and surface generation goals are separately optimized in an iterative fashion. The improved *BLEU* score in Table 1 for Co-Gen also demonstrates this.

For methods without RL fine-tuning, it is interesting that we also observe the phenomenon of over-generating slot placeholders in UBAR generated responses, such as the one in the second turn. It generates *[value_choice], [value_type], [value_area], [value_name], [value_address] and [value_price]* in a single turn, which is rather different from the ground truth response where only one *[value_count]* is contained. Since UBAR purely relies on the powerful language modeling GPT-2 model and does not leverage RL, this might be due to the context seen in former turns. Such phenomenon is also observed in the co-generation baseline MarCo, which also tends to give redundant information.

5 CONCLUSION

In conclusion, we proposed a neural co-generation framework for generating structured dialog acts and natural language responses concurrently for conversational search systems. It roots from a shared latent space that is shaped by two informed prior distributions. Accordingly, we formed two auto-encoding branches for structured dialog acts and natural responses as two auxiliary tasks to capture the generative factors of them. The joint training in an interconnected network structure makes the learned latent variables possess well-separated semantic meanings. Furthermore, we designed an asynchronous reinforcement learning mechanism to fine-tune the network with long-term rewards, which enables the model to foresee the future for better search completion and success rates. We carried out extensive experiments on two public datasets in comparison with a wide range of baselines. Both automatic and human evaluation are involved. The superior performance demonstrates that the proposed Co-Gen model generates better responses in both forms.

In the future, we look forward to applying our method for personalized response generation in conversational search when more such data is available. We would also like to further improve the strategy planning part in handling conversational search situations unseen during training and analyze how our model performs in real conversation interaction with more unseen situations.

ACKNOWLEDGMENTS

This research was supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant, and the Sea-NExT Joint Lab.

REFERENCES

[1] Avishek Anand, Lawrence Cavedon, Hideo Joho, Mark Sanderson, and Benno Stein. 2020. Conversational search (dagstuhl seminar 19461). In *Dagstuhl Reports*, Vol. 9.

[2] 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 *EMNLP*. 5016–5026.

[3] Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically Conditioned Dialog Response Generation via Hierarchical Disentangled Self-Attention. In *ACL*. 3696–3709.

[4] Yun-Nung Chen, William Yang Wang, and Alexander I Rudnicky. 2013. Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing. In *ASRU*. 120–125.

[5] Leigh Clark, Nadia Pantidi, Orla Cooney, Philip Doyle, Diego Garaialde, Justin Edwards, Brendan Spillane, Emer Gilmartin, Christine Murad, Cosmin Munteanu, et al. 2019. What makes a good conversation? Challenges in designing truly conversational agents. In *CHI*. 1–12.

[6] Chen Cui, Wenjie Wang, Xuemeng Song, Minlie Huang, Xin-Shun Xu, and Liqiang Nie. 2019. User attention-guided multimodal dialog systems. In *SIGIR*. 445–454.

[7] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. CAsT 2020: The Conversational Assistance Track Overview. In *TREC*.

[8] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. TREC CAsT 2019: The Conversational Assistance Track Overview. *ArXiv* abs/2003.13624 (2020).

[9] Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. 2017. Learning cooperative visual dialog agents with deep reinforcement learning. In *ICCV*. 2951–2960.

[10] Yashar Deldjoo, Johanne R Trippas, and Hamed Zamani. 2021. Towards multimodal conversational information seeking. In *SIGIR*, Vol. 21.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. 4171–4186.

[12] Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Peter Ku, Anuj Kumar Goyal, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. 2019. MultiWOZ 2.1: A Consolidated Multi-Domain Dialogue Dataset with State Corrections and State Tracking Baselines. arXiv:1907.01669 [cs.CL]

[13] Xiaodong Gu, Kang Min Yoo, and Sang-Woo Lee. 2021. Response Generation with Context-Aware Prompt Learning. *arXiv preprint arXiv:2111.02643* (2021).

[14] He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling Strategy and Generation in Negotiation Dialogues. In *EMNLP*. 2333–2343.

[15] Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796* (2020).

[16] Magdalena Kaiser, Rishiraj Saha Roy, and Gerhard Weikum. 2020. Conversational Question Answering over Passages by Leveraging Word Proximity Networks. *SIGIR*.

[17] Johannes Kiesel, Arefeh Bahrami, Benno Stein, Avishek Anand, and Matthias Hagen. 2018. Toward voice query clarification. In *SIGIR*. 1257–1260.

[18] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[19] Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020. Interactive path reasoning on graph for conversational recommendation. In *SIGKDD*. 2073–2083.

[20] Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or No Deal? End-to-End Learning of Negotiation Dialogues. In *EMNLP*. 2443–2453.

[21] Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep Reinforcement Learning for Dialogue Generation. In *EMNLP*. 1192–1202.

[22] Lizi Liao, Le Hong Long, Yunshan Ma, Wenqiang Lei, and Tat-Seng Chua. 2021. Dialogue State Tracking with Incremental Reasoning. *TACL* 9 (2021), 557–569.

[23] Lizi Liao, Le Hong Long, Zheng Zhang, Minlie Huang, and Tat-Seng Chua. 2021. MMConv: An Environment for Multimodal Conversational Search across Multiple Domains. In *SIGIR*. 675–684.

[24] Lizi Liao, Tongyao Zhu, Le Hong Long, and Tat Seng Chua. 2021. Multi-domain dialogue state tracking with recursive inference. In *WWW*. 2568–2577.

[25] Bing Liu and Ian Lane. 2017. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. In *INTERSPEECH*. 2506–2510.

[26] Nurul Lubis, Christian Geishauser, Michael Heck, Hsien-chin Lin, Marco Moresi, Carel van Niekerk, and Milica Gasic. 2020. LAVA: Latent Action Spaces via Variational Auto-encoding for Dialogue Policy Optimization. In *COLING*. 465–479.

[27] Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured Fusion Networks for Dialog. In *SIGDIAL*. 165–177.

[28] Ida Mele, Cristina Ioana Muntean, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, and Ophir Frieder. 2020. Topic propagation in conversational search. In *SIGIR*. 2057–2060.

[29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*. 311–318.

[30] Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. SOLOIST: Few-shot Task-Oriented Dialog with A Single Pre-trained Auto-regressive Model. *arXiv preprint arXiv:2005.05298* (2020).

[31] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[32] Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *CHIIR*. 117–126.

[33] 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 *SIGDIAL*. 147–157.

[34] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *SIGIR*. 235–244.

[35] Richard S Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112, 1-2 (1999), 181–211.

[36] Johanne R Trippas, Damiano Spina, Paul Thomas, Mark Sanderson, Hideo Joho, and Lawrence Cavedon. 2020. Towards a model for spoken conversational search. *Information Processing & Management* 2 (2020), 102162.

[37] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[38] Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. 2020. Modelling Hierarchical Structure between Dialogue Policy and Natural Language Generator with Option Framework for Task-oriented Dialogue System. In *ICLR*.

[39] Kai Wang, Junfeng Tian, Rui Wang, Xiaojun Quan, and Jianxing Yu. 2020. Multi-Domain Dialogue Acts and Response Co-Generation. In *ACL*. 7125–7134.

[40] Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *EMNLP*. 1711–1721.

[41] 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 *EACL*. 438–449.

[42] Jason D Williams and Steve Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language* 21, 2 (2007), 393–422.

[43] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3 (1992), 229–256.

[44] Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. UBAR: Towards Fully End-to-End Task-Oriented Dialog System with GPT-2. In *AAAI*. 14230–14238.

[45] Hamed Zamani and Nick Craswell. 2020. Macaw: An extensible conversational information seeking platform. In *SIGIR*. 2193–2196.

[46] Shuo Zhang and Krisztian Balog. 2020. Evaluating conversational recommender systems via user simulation. In *SIGKDD*. 1512–1520.

[47] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *CIKM*. 177–186.

[48] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020. DIALOGPT: Large-Scale Generative Pre-training for Conversational Response Generation. In *ACL*. 270–278.

[49] Zheng Zhang, Lizi Liao, Minlie Huang, Xiaoyan Zhu, and Tat-Seng Chua. 2019. Neural multimodal belief tracker with adaptive attention for dialogue systems. In *WWW*. 2401–2412.

[50] Zheng Zhang, Lizi Liao, Xiaoyan Zhu, Tat-Seng Chua, Zitao Liu, Yan Huang, and Minlie Huang. 2020. Learning Goal-oriented Dialogue Policy with Opposite Agent Awareness. *arXiv preprint arXiv:2004.09731* (2020).

[51] Tiancheng Zhao and Maxine Eskenazi. 2016. Towards End-to-End Learning for Dialog State Tracking and Management using Deep Reinforcement Learning. In *SIGDIAL*. 1–10.

[52] 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 *NAACL*. 1208–1218.