Predict and Use Latent Patterns for Short-Text Conversation

Hung-Ting Chen[*], **Yu-Chieh Chao**[*], **Ta-Hsuan Chao**[*], **Wei-Yun Ma**

Academia Sinica

{timchen0618, vpj870331, alexchao2007, ma}@iis.sinica.edu.tw

Abstract

Many neural network models nowadays have achieved promising performances in Chit-chat settings. The majority of them rely on an encoder for understanding the post and a decoder for generating the response. Without given assigned semantics, the models lack the fine-grained control over responses as the semantic mapping between posts and responses is hidden on the fly within the end-to-end manners. Some previous works utilize sampled latent words as a controllable semantic form to drive the generated response around the work, but few works attempt to use more complex semantic forms to guide the generation. In this paper, we propose to use more detailed semantic forms, including latent responses and part-of-speech sequences sampled from the corresponding distributions, as the controllable semantics to guide the generation. Our experimental results show that the richer semantics are not only able to provide informative and diverse responses, but also increase the overall performance of response quality, including fluency and coherence.

1 Introduction

With the advent of machine learning and neural networks, machines can handle more and more human jobs, including Chit-chat setting tasks. The sequence to sequence model (Vinyals and Le 2015; Shang, Lu, and Li 2015; Sordoni et al. 2015) is one of the first successful models that generate text by encoder-decoder architecture. However, without given assigned semantics, the models usually lack the fine-grained control over responses and are prone to producing generic and meaningless responses such as "I don't know" or "So am I".

To guide generation to produce more complex results, some works involve utilizing controllable semantics. Numerous prior studies (Xing et al. 2016; Dziri et al. 2018; Mou et al. 2016; Yao et al. 2017; Gao et al. 2019a) use keywords, either selected by neural networks or retrieval models, as semantic features to control the generation of the output text. Results indicated that the extracted latent word can become an auxiliary semantic feature to guide the response generation. However, the aforementioned methods mainly use simple semantic forms to assist generation. Also, most of the sentences they produce do not have fluent sentence patterns.

In this paper, we propose an end-to-end model that creates informative and diverse responses by providing more complex semantics. We investigate two different latent patterns - latent sentences and latent part-of-speech (POS) sequences for guiding semantics. First, as many previous works use a word to assist dialogue generation, we use a complete sentence instead. Not only can a sentence provide some pattern information, but additional word usage information to yield better dialogue responses. Second, inspired by Shen et al. (2019), who uses POS sequences to control the generation of Chinese poems, we serve POS sequences as hidden semantics to aid the Chit-chat task. We have shown that this method greatly helps improve the generation of dialogues, and that our generations greatly depend on the POS sequences we chose or generated.

Our model consists of two parts, the latent sequence predictor (which we would experiment on latent sentences and latent POS sequences), and the dialogue generator, which generates responses by the input post and the corresponding latent semantics we chose. After pre-training both parts (latent sequence predictor and dialogue generation network), we combine the two networks into one end-to-end dialogue generation network. Finally, follow Gao et al. (2019a), we fine-tune the whole network using a reinforcement learning (RL) algorithm.

Our contributions can be summarized as the following:

- We proposed an end-to-end model with RL that predicts and utilizes complex latent patterns to guide the generation of dialogues with given input text.

- We used both latent sentence and latent POS sequence as our semantic pattern and out-performed several state-of-the-art dialogue generation models.

- We implemented our dialogue generation models and all of our source codes and datasets are available at [1]

2 Related Work

Many researchers have proposed methods to resolve the generic response problem. Maximum Mutual Information (MMI) objective function (Li et al. 2016a), diverse

---

[*]indicates equal contribution

---

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

beam search (Li, Monroe, and Jurafsky 2016), segment-by-segment reranking (Shao et al. 2016) are all used to address the issue. These approaches only consider modifying the generation at decoding steps, not at training time. Training schemes such as reinforcement learning (Ranzato et al. 2015; Li et al. 2016b; Cuayáhuitl 2017; Saleh et al. 2019) and adversarial training (Li et al. 2017; Xu et al. 2017) are also applied. One major drawback of these methods is that they cannot explicitly control what to produce in the output; they only model 'quality' concerning if the responses are human-like or if they are similar to their targets.

Regarding control of semantics in a response, two main categories of methods prevail in the dialogue generation domain, as presented in the following subsections.

2.1 Rewriting and Editing a Sentence

Retrieval methods (Ji, Lu, and Li 2014; Yan, Song, and Wu 2016; Yan et al. 2016) have proved their capability to produce more human-like and fluent outputs in generation tasks. Many have proposed to rewrite the retrieved candidate sentences to improve fluency and relevance to the input context. The candidates can serve as a relatively good (starting point) since they are fluent and grammatically correct, and then editing is performed to improve relevance and coherence. Plenty of NLP tasks have adopted the technique, including summarization (Cao et al. 2018; Hossain, Ghazvininejad, and Zettlemoyer 2020), image caption generation (Kuznetsova et al. 2013; Mason and Charniak 2014), code generation (Hashimoto et al. 2018) and machine translation (Guu et al. 2018; Cao and Xiong 2018; Hossain, Ghazvininejad, and Zettlemoyer 2020).

In dialogue generation specifically, Song et al. (2016) combine retrieval and generative systems by referencing a candidate response through multi-source attention and post-reranking. (Weston, Dinan, and Miller 2018) refine the retrieved response by treating it as an additional context. Wu et al. (2019) treat the retrieved response as context and consider the lexical difference between the retrieved and the original post by augmenting the encoder-decoder model with an edit vector. Our work shares some similarities with these methods since we also reference a candidate response in the generation process. However, we directly select a candidate response through neural models without doing retrieval in the first place, eliminating the need for preprocessing of input data. Also, the model can be trained in an end-to-end manner and inference as a whole.

2.2 Utilizing Latent Semantics Forms

Others have sought to enhance the quality of responses by utilizing latent features hidden in the post-response pairs since they provide content or functional information to guide the generation. A wide variety of methods are built upon the help of a latent word. Two previous studies (Xing et al. 2016; Dziri et al. 2018) select topic words with LDA (Latent Dirichlet Allocation) models and augment the encoder-decoder with topic-aware attention architecture. Mou et al. (2016) and Yao et al. (2017) both select a latent word with the highest PMI (Pointwise Mutual Information) against the input post. Then the former produces a sentence containing
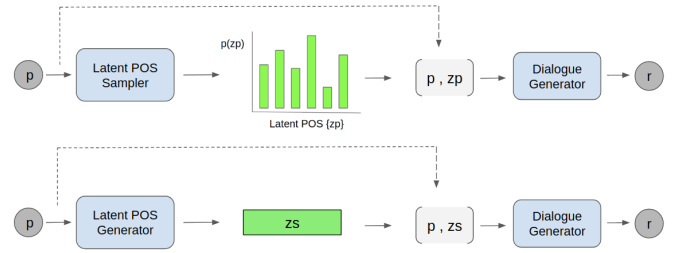

Figure 1: An overview of the proposed model.

the word, and the latter integrates the information of the chosen word with a modified version of Gated Recurrent Unit (GRU). Gao et al. (2019a) first select a latent word from the vocabulary and make the decoder attend to both the latent word and context. The selector and the generator are trained jointly and fine-tuned with reinforcement learning. Gao et al. (2019b) improve the CVAE structure by sampling a word from the vocabulary with a two-stage sampling scheme.

Researchers have also explored the possibility of using other latent semantic forms such as topic (Wang et al. 2017), sentence function (Ke et al. 2018; Bi et al. 2019), frame semantics (Gupta et al. 2020), lexical phrases (Wu et al. 2020), or goals in conversation (Tang et al. 2019). Previous works have mainly focused on fusing the information of much simpler semantic forms into generation, thus altering few words in the returned response and might not be as effective. Instead, we attempt to guide the process with more complex semantic forms, explicitly providing the model with patterns, to follow suit. The model can, therefore, produce more coherent and human-like responses.

3 Models

The task we cope with in this paper is open-domain dialogue generation. A input post $\mathbf{p} \in \mathbf{P}$ is associated with multiple responses $\{\mathbf{r}\}$, where $\mathbf{r} \in \mathbf{R}$. Here $\mathbf{P}$ and $\mathbf{R}$ stands for the set of all the posts and responses, respectively. We formulate the task as a sequence-to-sequence problem; given an input post $\mathbf{p} = \{p_1, ..., p_i, ..., p_n\}$ the model is expected to generate a response $\mathbf{r} = \{r_1, ..., r_t, ..., r_m\}$, where n and m represents the length of the post and response respectively. Latent semantic sequences $\mathbf{z} = \{z_1, ..., z_j, ..., z_l\}$ are introduced in the model. We denote latent sentences as $\mathbf{z_s} \in \mathbf{Z_s} = \mathbf{R}$ and latent part-of-speech tags as $\mathbf{z_p} \in \mathbf{Z_p}$, where $\mathbf{Z_p} = \{pos(\mathbf{r}) | \forall \mathbf{r} \in \mathbf{R}\}$.

Our model is comprised of two major components, as shown in figure 1. The latent sequence predictor aims to predict the latent semantic sequence $\mathbf{z}$ conditioned on the input post $\mathbf{p}$; the dialogue generator produces a response $\mathbf{r}$ according to the input post and the latent semantic information obtained from the latent sequence predictor. Both models are first pretrained and then trained jointly in an end-to-end manner using reinforcement learning.

3.1 Latent Sequence Predictor

We tackle the selection of latent sentences and POS sequences as a classification problem. Due to the huge number

(the same as the number of instances in the corpus) of candidates, it is unlikely that we directly select one from the whole training dataset. Hence we construct a candidate set $\tilde{\mathbf{Z}}_\mathbf{i} \subset \mathbf{Z}_\mathbf{i}$ ($i \in \{s, p\}$) with K_i latent sequences from the latent space $\mathbf{Z}_\mathbf{i}$, as described in 3.3. Then for each input post $\mathbf{p}$ we sample a $\tilde{\mathbf{z}}_\mathbf{i}$ from the candidate set $\tilde{\mathbf{Z}}_\mathbf{i}$ to guide the generation. To encourage the the production of more diverse POS sequences, we also propose a generation model for the Latent POS Predictor, which produces a POS sequence given the input post.

Latent Sentence Predictor   This network samples latent sentence $\tilde{\mathbf{z}}_\mathbf{s}$ from candidate set $\tilde{\mathbf{Z}}_\mathbf{s}$ by estimating the probability distribution $p(\tilde{\mathbf{z}}_\mathbf{s}|\mathbf{p})$. We first encode the input post $\mathbf{p}$ with a bidirectional GRU (biGRU) to obtain an input representation $\mathbf{h}^\mathbf{P}$, and retrieve hidden states of the last time step t of the biGRU encoder as the hidden vector to represent the meaning of the post:

$$h_t^p = [\overrightarrow{h_t^p}, \overleftarrow{h_t^p}]$$

and then compute the probability of each latent sentence as:

$$p(\tilde{\mathbf{z}}_\mathbf{s}|\mathbf{p}) = softmax(MLP(h_t^p))$$

where parameters in the biGRU layer and Multilayer Perceptron (MLP) are trainable parameters. h_t^p denotes the hidden vector of last time step t of the encoder.

Latent POS Predictor   Just like the latent sentence sampler, our POS sequences can also be sampled from candidate set $\tilde{\mathbf{Z}}_\mathbf{p}$. However, in POS generation, we also used a generation model for latent POS sequences. In the following sections, we will elaborate on both the latent POS sampler and the latent POS generator.

- Latent POS Sampler

  To sample a POS sequence using Transformer model, we simply adopt the encoder part of the Transformer as proposed in (Vaswani et al. 2017) as our sentence encoder:

$$\mathbf{h}^\mathbf{P} = TransformerEncoder(\mathbf{p})$$
$$\mathbf{h}^\mathbf{P} = [h_1^p, h_2^p, ..., h_n^p]$$

  where n is the length of the input post. Then we treat the last time step of the hidden vectors h_n^p as our sentence vector. Finally, a MLP (Multi-Layer Perceptron) classifier is used to sample the probability distribution of latent POS space:

$$p(\tilde{\mathbf{z}}_\mathbf{p}|\mathbf{p}) = softmax(\mathbf{MLP}(h_n^p))$$

  where parameters in the MLP block are trainable.

- Latent POS Generator

  Instead of sampling a POS sequence from all POS sequence candidates, we adopt the Transformer to generate a POS sequence $\tilde{\mathbf{z}}_\mathbf{pg}$ based on the input post $\mathbf{p}$.

3.2   Dialogue Generator

This model aims to generate sequence $\mathbf{r}$ from predicted latent semantics and input post $\mathbf{p}$.

Dialogue Generator with Latent Sentence   With a sampled $\tilde{\mathbf{z}}_\mathbf{s}$ from latent sentence predictor, we first encode both the input post $\mathbf{p}$ and latent sentence $\tilde{\mathbf{z}}_\mathbf{s}$ through two independent bidirectional GRU network for input representations $\mathbf{h}_\mathbf{p}^\mathbf{g}$ and $\mathbf{h}_{\tilde{\mathbf{z}}_\mathbf{s}}^\mathbf{g}$ respectively. Both representations are then leveraged to decode the output sequence $\mathbf{y}$:

$$h_p^g = \mathbf{W}_\mathbf{p}[\overrightarrow{h_p^g}, \overleftarrow{h_p^g}]$$
$$h_{\tilde{z}_s}^g = \mathbf{W}_{\tilde{\mathbf{z}}_\mathbf{s}}[\overrightarrow{h_{\tilde{z}_s}^g}, \overleftarrow{h_{\tilde{z}_s}^g}]$$

where $\mathbf{W}_\mathbf{p}$ and $\mathbf{W}_{\tilde{\mathbf{z}}_\mathbf{s}}$ are learnable linear mapping layers.

On each time step t, the decoder receives the word embedding of the previous word (while training, this is the previous word of the target response; at test time, it is the previous word emitted by the decoder), and has decoder state $\mathbf{s}_\mathbf{t}$. The attention calculation is implemented as in (Bahdanau, Cho, and Bengio 2014):

$$e_i^t = v\, tanh(\mathbf{W}_\mathbf{h}h_i + \mathbf{W}_\mathbf{s}s_t + b_{attn})$$
$$\alpha_t = softmax(e_t)$$
$$c_t = \sum_i \alpha_i^t h_i$$

where v, $\mathbf{W}_\mathbf{h}$, $\mathbf{W}_\mathbf{s}$ and b_{attn} are learnable parameters. c_t is the context vector, which is the weighted sum of the encoder hidden states scaled by the attention distribution $\alpha_\mathbf{t}$. In our work, we present multi-level attention mechanism, which includes decoder state attends to encoding representation of both the input post and the latent sentence. Two attention mechanisms are implemented independently.

In this work, we adopt pointer generator network as proposed in (See, Liu, and Manning 2017). Together with the intention to solve undesirable behavior exhibited by recurrent neural networks, we expect the pointer-generator network to benefit our GRU decoder at every time step by copying those factual details from the latent sentence. And for those remaining indices, the model produces words by generation, leading the final response sentence to be coherent with the input post. To balance between copy and generation, we need to calculate $\mathbf{p}_\mathbf{gen}$ as:

$$p_{gen} = \sigma W_{gen}[s_t, c_t^p, c_t^{\tilde{z}_s}]$$
$$l_{copy} = 1 - p_{gen}$$

where p_{gen} denotes generation probability, while l_{copy} denotes copy probability (copying word from the latent sentence). W_{gen} is a learnable parameter, and σ is the sigmoid function. Within the concatenated vectors, $\mathbf{c}_\mathbf{t}^\mathbf{P}$ represents the input post context vector and $c_t^{\tilde{z}_s}$ represents the latent sentence context vector.

Finally, the probability distribution over the extended vocabulary (pre-set vocabulary & OOV words) is:

$$P(w) = p_{gen}P_{vocab}(w) + (1 - p_{gen})\sum_{i:w_i=w} a_{i,t}^{\tilde{z}_s}$$

where $a_{i,t}^{\tilde{z}_s}$ denotes latent sentence attention distribution. Note that if w is an out-of-vocabulary (OOV) word, then

$P_{vocab}(w)$, which represents the probability distribution over pre-set vocabulary, is zero; similarly if w does not appear in the latent sentence, then $\sum_{i:w_i=w} a_{i,t}^{\tilde{z_s}}$, which stands for the probability distribution over words in latent sentence, is zero.

Dialogue Generator with POS sequence Inspired by (Shen et al. 2019), we concatenate the predicted POS sequence $\tilde{\mathbf{z_p}}$ (or in the generated POS sequence case, $\tilde{\mathbf{z_{pg}}}$) right behind the post at the input layer. Next, we forward the concatenated form into Transformer architecture as in (Vaswani et al. 2017) to generate the corresponding response:

$$\hat{\mathbf{r}} = Transformer([\mathbf{p}, \tilde{\mathbf{z_p}}])$$

The technique has been proved capable of producing sentences matching the input POS sequences, hence providing enough guidance for generating responses with valid structure.

3.3 Pretraining and Data Preparation

We first pretrain the latent sequence predictor and the dialogue generator separately. For the latent sentence predictor and latent POS sampler, they are trained to solve a sentence classification problem. The creation of data for pretraining is explained in details below:

Candidate Set of Latent Sentences We attempt to construct a candidate set of size K_s representative enough of the entire response set $\mathbf{R}$. First, we use kmeans-clustering to aggregate responses into C clusters, and select K_s/C responses from each cluster. We then assign an index, i.e., label for classification, to each of the latent responses in $\tilde{\mathbf{Z_s}}$. Finally, we assign the label of $\mathbf{z_s^*}$ to each $(\mathbf{p}, \mathbf{r})$, where $\mathbf{z_s^*}$ is the most similar response in $\tilde{\mathbf{Z_s}}$ with the target response $\mathbf{r}$. The similarity between responses is defined as the Euclidean distance of the BERT[2] sentence encoding of the two.

Candidate Set of Latent POS Sequences First, we derive the part-of-speech tagging of each response in $\mathbf{R}$ using Jiagu[3], a natural language processing tool. Likewise, the candidate POS sequences must be representative of all possible POS sequences in $\mathbf{Z_p}$. Thus we pick the K_p most common POS sequences in $\mathbf{Z_p}$ as our candidate set $\tilde{\mathbf{Z_p}}$. (Some responses might share the same POS sequence.) After constructing the latent POS sequence space, we create the training data for the latent POS sampler. We first assign classification label to each of the $\mathbf{z_p} \in \tilde{\mathbf{Z_p}}$. For each input post-response pair $(\mathbf{p}, \mathbf{r})$, we find the most similar $\mathbf{z_p^*}$ to the POS sequence of r and label the pair with the corresponding index of $\mathbf{z_p^*}$. That is, we wish to predict the class label of $\mathbf{z_p^*}$ given the input post $\mathbf{p}$. The similarity between POS sequences is calculated by alignment[4].

---

[2] https://github.com/hanxiao/bert-as-service

[3] https://github.com/ownthink/Jiagu

[4] https://biopython.org/DIST/docs/api/Bio.pairwise2-module.html

Alignment is a dynamic programming algorithm, which finds the best concordance between all characters in two sequences.

The generation of POS sequences and responses are both considered sequence-to-sequence generation problems. The latent semantic generator reads in the input post and outputs a POS sequence. We try to optimize the standard Negative Log Likelihood (NLL) loss function. For the dialogue generator, we simply concatenate the POS sequence $\mathbf{z_p}$ of response $\mathbf{r}$ after the post $\mathbf{p}$ and input the newly created sequence $[\mathbf{p};\mathbf{z_p}]$ into the Seq2Seq model.

3.4 Joint Training With Reinforcement Learning

To acquire better latent semantic sequences for the generation model, we fine-tune both models end-to-end with a reinforcement-learning objective. We treat the latent sequence predictor as an agent and the prediction of latent semantics as actions. We intend to choose the policy which enables the model to gain maximum total rewards. Note that during joint training, we apply the reinforcement learning algorithm only on the latent sequence predictor, whereas the dialogue generator is directly optimized through standard cross-entropy loss. We apply the REINFORCE (Williams, 1992) algorithm, a Monte-Carlo based policy gradient method; the formula of updating parameters of the latent sequence predictor (denoted as θ) is as follows. $Q(\hat{\mathbf{z}})$ stands for estimated return if we sample a latent $\hat{\mathbf{z}}$ ($\hat{\mathbf{z}}$ can be either a latent sentence or a latent POS sequence), and note that we only calculate the return after the full response is generated.

- For selecting a sequence (either a sentence or a POS sequence)

$$\theta \leftarrow \theta + \nabla_\theta Q(\hat{\mathbf{z}}) \log p_\theta(\hat{\mathbf{z}}|\mathbf{p})$$

- For generating a POS sequence

$$\theta \leftarrow \theta + \sum_{j=0}^{l} \nabla_\theta Q(\mathbf{z^{\hat{p}}}) \log p_\theta(\mathbf{z_j^{\hat{p}}}|\mathbf{p})$$

We model the estimated return as a reward function measuring the similarity between the predicted and target response. (We found empirically that it is challenging for the model to match all the responses during joint training since it has to explore the large latent space to select appropriate latent semantic sequences for all the responses.) We update the latent sequence predictor only according to the maximum reward acquired from the bag of responses. The reward function $Q(\hat{\mathbf{z}})$ is designed as follows; here $\{\mathbf{r}\}$ denote the set of multiple resopnses associated with the post $\mathbf{p}$

$$Q(\hat{\mathbf{z}}) = R(\hat{\mathbf{r}}, \{\mathbf{r}\}) = \max_{\mathbf{r} \in \{\mathbf{r}\}} \mathbf{R}(\hat{\mathbf{r}}, \mathbf{r}) \qquad (1)$$

, where

$$\mathbf{R}(\hat{\mathbf{r}}, \mathbf{r}) = F1(\hat{\mathbf{r}}, \mathbf{r})$$

Here, a variation of F1 score calculates the overlap between the generated response and the target. Results have shown the joint training scheme enhances overall response quality.

Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Seq2Seq	25.57	7.39	1.71	0.64
CVAE (Zhao, Zhao, and Eskenazi 2017)	20.13	6.82	1.36	0.52
HGFU (Yao et al. 2017)	27.31	9.59	**4.08**	**2.14**
GMDR (Gao et al. 2019a)	21.26	8.27	3.93	2.08
Ours-Two Stage (latent sentence)	17.19	4.72	0.42	0.10
Ours-Two Stage (sample POS)	**22.28**	**5.31**	1.11	0.00
Ours-Two Stage (generate POS)	22.84	6.08	1.07	0.00
Ours-RL (latent sentence)	19.21	6.83	1.61	0.23
Ours-RL (sample POS)	**36.42**	**12.35**	3.55	0.99
Ours. RL (generate POS)	30.98	10.21	3.25	1.29

Table 1: Automatic evaluation results on Weibo.

4 Experiments

4.1 Dataset

Our method is evaluated on the Weibo Bechmark Dataset, an open-domain Chinese dialogue dataset with over 400 million training pairs. The testing set is composed of 3,200 data pairs. Here, we use the modified version released by (Gao et al. 2019a). [5]

4.2 Evaluation

We evaluate the dialogues predicted by the BLEU score (Papineni et al. 2002), a widely used metric measuring the similarity between the predicted response and the ground truth. We provide the values of BLEU 1-4 for all the experiments.

However, since the automatic evaluation metrics cannot effectively reflect the goodness of dialogue generation, we also did human evaluation on 100 randomly selected sentences from our and other comparing models. Three native speakers are asked to label the responses according to following quality indicators:

- Fluency: BLEU score does not consider word ordering, thus fluency can only be judged by human evaluation.

- Relevance: Determine whether the generated response replies or is relevant with the input post.

- Informativeness: Generic and meaningless responses such as "I don't know" or "So am I." would get low scores in this category.

4.3 Baseline Methods

Seq2Seq We use a vanilla sequence-to-sequece model implemented with transformer encoder-decoder. The parameters are identical to those of our dialogue generator. Beam search decoding is also used in testing time.

HGFU A modified GRU incorporating a cue word in the generation process proposed by (Yao et al. 2017). For fair comparison, we use the latent word predicted by the latent word inference network as the cue word.

K_p	500	1000	5000	10000	50000
coverage	1.31%	1.93%	4.41%	6.03%	11.58%

Table 2: Top K_p POS occurrence number coverage table

CVAE Variational AutoEncoder (VAE) based latent space model (Zhao, Zhao, and Eskenazi 2017) implemented with LSTM (Long Short-Term Memory) cells. We modify the model structure so that it can fit in the single-turn dialogue setting.

GMDR An additional latent word inference network combined with a joint-attention generation network (Gao et al. 2019a), attending on both the selected word and the post. Reinforcement learning is applied to train the whole model end-to-end.

4.4 Implementation Details

Data Preparation

- Latent Sentence

We set $C = 1000$ and $K_s = 50000$ for the construction of the candidate set $\tilde{\mathbf{Z}}\mathbf{s}$. The pretraining data for the latent sentence sampler thus includes input post $\mathbf{P}$ as input and distribution of $\mathbf{Z_s^*}$ as output.

- Latent POS Sequence

We expect the candidate set $\tilde{\mathbf{Z}}\mathbf{p}$ to be representative enough of $\mathbf{Zp}$. Thus we made a simple analysis of the coverage of top K_p latent POS sequences chosen in table 2. For example, if we choose 500 latent POS tag sequences as our latent space ($K_p = 500$), we could cover 1.31% of the whole dataset. Thus, we get our latent POS sequence latent space $\tilde{\mathbf{Z}}\mathbf{p}$ according to the K_p we choose. In our experiments, we choose $K_p = 500, 1000$, and 10000 to test how different coverage affect the performance of our model.

- Training Details

Details related to our experiments are shown in Appendix, which elaborates the settings including hyper-parameters and the design of our model architectures.

Models	Fluency	Relevance	Informativeness
HGFU	3.69	2.65	2.60
GMDR	**4.51**	3.03	2.73
Ours(select POS)	3.98	**3.36**	**3.34**

Table 3: Result of Human Evaluation.

5 Results

5.1 Automatic Evaluation

The evaluation results of our model are shown at Table 1. As we can see, our Ours-RL(sample POS) outperforms other models in BLEU 1 and 2. The reason is that the sentence pattern we provided narrows down the word selection of each time step. For example, if the corresponding POS tag for the token at this time step is v, then the model will generate a verb with higher probability. Thus, our model can choose more accurate and relevant word choice than other models, and thus achieves outstanding performance in BLEU 1 and 2. However, since our model forces the pattern of our generation (which is normally different from the original response), the longer word matching metric BLEU 3 and 4 doesn't perform as well. We also report the performance of combining the pretrained models as Ours-Two Stage, i.e., directly using the latent sequence predicted by pretrained latent sequence predictor in the generation process. The BLEU scores improved greatly after fine-tuning with our reinforcement learning algorithm since both the latent sentence predictor and dialogue generation model are optimized in an end-to-end manner, enabling more appropriate prediction of latent sequences. The BLEU score of Ours-RL(sample POS) improved **63%** and **132%** and achieved **36.42** and **12.35** in BLEU-1 and BLEU-2 scores.

However, our latent sentence part does not yield good BLEU scores because the generation of this model is greatly influenced by the latent sentence we have chosen, as shown in . Thus, if the latent sentence we have chosen is very different from the original response, the BLEU scores would drop.

5.2 Human Evaluation

The result of human evaluation is shown in table 3. When evaluating the indicators (fluency, relevance, and informativeness), the scores given by the native speakers consists of five levels, where five means best in each indicators and one is the worst. The model that we use to compare with other models is latent POS sampler and $K_p = 500$.

Results show that our model obtains the highest scores in both relevance and informativeness. Since our model adds pattern information, the response we generate is prone to be more informative. It is 22% more informative than GMDR, and 28% more imformative than HGFU. However, since the outputs of the generated sentences are confined to the POS patterns, they might sacrifice fluency slightly, but is still better than HGFU.

5.3 Case Study and Analysis

Table 4 shows one of the dialogue predicted by our mod-

els and other models. The model GMDR, proposed in (Gao et al. 2019a) generates dialogue by a latent word *darling*, and thus predicted a simple sentence *All of them are my darlings*. However, our models predicted more complex results, both in word usage and sentence patterns. In our first method, which uses latent sentences to assist, chose *The sky of the liberated area is a sunny day.* as the latent sentence. Thus, it affected the generation *Oh my god* (because in Chinese, *day* 天 is equal to *god* 天). As for the sentence predicted by POS, it is easy to see why our best results are those produced by the sampler. Since we have a representative candidate POS sequence set, the POS sequence chosen greatly adjusted the output sentence pattern. Thus, we obtain an informative and relevant response *They are so cute, I'll be blessed, too. I miss my darling so much.*

5.4 Analysis On the Selection of Latent Space

Table 5 shows the results of how different latent space size K_p affect the performance of our model. We can see that the best results occur at $K_p = 500$, and thus according to table 2, our candidate set of latent POS sequences covers about 1.3% of our responses in the training dataset.

5.5 Faithfulness to Given Semantic Sequences

To prove that the selected latent semantic sequence is actually helpful for generation, we inspect if the generated response follows the provided word or POS sequence.

For model utilizing latent sentences, we inquire whether the model mimics the word selection (providing content) and the word ordering (providing word usage, sentence pattern, and content ordering) of the given latent sentence. As for word selection, we calculate the percentage of overlapping words with the latent sentences in generated responses. For word order, we calculate the percentage of overlapping n-grams. The result is presented in 6.

As can be seen from the result, the model copies over two-third of words and about one-forth of the words from the latent sentence. It clearly illustrates that the model indeed reference to the latent sentence for better response generation. However, the model might also suffer from over-copying. If the selected sentence is not relevant enough to the input post, the response will not be as good. Thus we believe the overall response quality will improve if we could ensure better latent sentence selection, which may be a potential future study.

For models using latent POS sequences, the output responses almost always follow the given POS sequence during pretraining. However, when fine-tuning, the model does not strictly follow the given POS sequences since it learns to improvise a little to generate more coherent responses. We believe our model follows the guidance of the underlying patterns of latent POS sequence, hence producing high-quality responses. It also maintains a certain extent of freedom concerning the generation of word sequences, ensuring the fluency of responses. To support the claim, We measure the faithfulness of generation to the provided POS sequence using the edit distance between the POS tags of output responses and the given POS sequence. If the edit distance between the response POS tags and the given POS sequence

Input	这些都是我的宝贝们 All of these are my darlings.
Seq2Seq	我也想说，我也想吃了。 I also want to say; I want to eat, too.
HGFU	都是我喜欢的，好可爱 All of them are my favorite, they're so cute.
GMDR	都是我的宝贝们。 All of them are my darlings.
CVAE	还有那个可以吗？ Is that available?
Ours(Latent Sentence)	我的天我的天，喜欢。 [Latent Sentence: 解放区的天是晴朗的天] Oh my god oh my god, like it. [Latent Sentence: The sky of the liberated area is a sunny day.]
Ours(Generate POS)	好可爱啊，好可爱，好喜欢。 So cute, so cute, and I like it so much.
Ours(Sample POS)	好可爱啊，我也要幸福的。好想我的宝宝啊。 They are so cute, I'll be blessed, too. I miss my darling so much.

Table 4: Evaluation results of different sizes of latent space

K_p	BLEU-1	BLEU-2	BLEU-3	BLEU-4
500	36.42	12.35	3.55	1.29
1000	28.60	9.12	3.09	1.07
10000	22.17	7.51	2.70	1.03

Table 5: Evaluation results of different sizes of latent space

	unigram	bigram	trigram	4-gram
overlap (%)	71.11	44.86	32.84	26.76

Table 6: N-gram overlap between latent sentences and responses
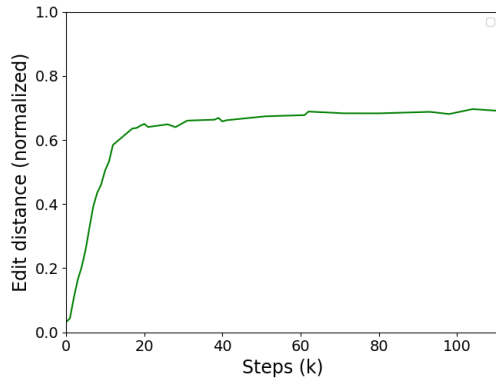

Figure 2: Edit distance during fine-tuning process of RL.

is small, it indicates that the model gathers information from the patterns provided by the latent POS and then predicts a response with similar patterns. The edit distance is normalized by the length of the selected POS sequence to simulate the mismatch percentage. Figure 2 shows the result during fine-tuning for our best model. Judging from the result, the model first strictly follows the selected latent POS sequence but later begins to generate responses more freely. We thus prove statistically that our model indeed strikes a balance between generating word segments on its own and referencing to the selected or generated POS sequence. The automatic evaluation results also proves our theory, since fine-tuned models significantly outperform the two-stage models.

6 Conclusion

We proposed an end-to-end response generation network that utilized complex latent semantics that aid the generation of the dialogue. Our model used latent sentences and latent part-of-speech sequences respectively to provide wording and sentence pattern information. Also, our combined end-to-end model can be optimized by a reinforcement learning algorithm, and allows the model to strike a balance between following the latent semantics or not. Our results showed that the network we proposed increased the quality of responses by being more informative and relevant to the input post compared with existing baselines.

7 Future Work

Our next step on this issue would be using a mixed sequence as latent semantics to help guide the generation of dialogues. For example, we can integrate both POS tags and words into a sequence to assist our model generate more informative and human-like results. Also, we can try other semantic forms to assist our generation, like Semantic Role Labeling (SRL), in our future work.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Bi, W.; Gao, J.; Liu, X.; and Shi, S. 2019. Fine-grained sentence functions for short-text conversation. *arXiv preprint arXiv:1907.10302* .

Cao, Q.; and Xiong, D. 2018. Encoding gated translation memory into neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3042–3047.

Cao, Z.; Li, W.; Li, S.; and Wei, F. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 152–161.

Cuayáhuitl, H. 2017. Simpleds: A simple deep reinforcement learning dialogue system. In *Dialogues with social robots*, 109–118. Springer.

Dziri, N.; Kamalloo, E.; Mathewson, K. W.; and Zaiane, O. 2018. Augmenting neural response generation with context-aware topical attention. *arXiv preprint arXiv:1811.01063* .

Gao, J.; Bi, W.; Liu, X.; Li, J.; and Shi, S. 2019a. Generating multiple diverse responses for short-text conversation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6383–6390.

Gao, J.; Bi, W.; Liu, X.; Li, J.; Zhou, G.; and Shi, S. 2019b. A discrete cvae for response generation on short-text conversation. *arXiv preprint arXiv:1911.09845* .

Gupta, P.; Bigham, J. P.; Tsvetkov, Y.; and Pavel, A. 2020. Controlling Dialogue Generation with Semantic Exemplars. *arXiv preprint arXiv:2008.09075* .

Guu, K.; Hashimoto, T. B.; Oren, Y.; and Liang, P. 2018. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics* 6: 437–450.

Hashimoto, T. B.; Guu, K.; Oren, Y.; and Liang, P. S. 2018. A retrieve-and-edit framework for predicting structured outputs. In *Advances in Neural Information Processing Systems*, 10052–10062.

Hossain, N.; Ghazvininejad, M.; and Zettlemoyer, L. 2020. Simple and Effective Retrieve-Edit-Rerank Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2532–2538.

Ji, Z.; Lu, Z.; and Li, H. 2014. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988* .

Ke, P.; Guan, J.; Huang, M.; and Zhu, X. 2018. Generating informative responses with controlled sentence function. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1499–1508.

Kuznetsova, P.; Ordonez, V.; Berg, A.; Berg, T.; and Choi, Y. 2013. Generalizing image captions for image-text parallel corpus. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 790–796.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2016a. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of NAACL-HLT*, 110–119.

Li, J.; Monroe, W.; and Jurafsky, D. 2016. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562* .

Li, J.; Monroe, W.; Ritter, A.; Galley, M.; Gao, J.; and Jurafsky, D. 2016b. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541* .

Li, J.; Monroe, W.; Shi, T.; Jean, S.; Ritter, A.; and Jurafsky, D. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547* .

Mason, R.; and Charniak, E. 2014. Domain-specific image captioning. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, 11–20.

Mou, L.; Song, Y.; Yan, R.; Li, G.; Zhang, L.; and Jin, Z. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. *arXiv preprint arXiv:1607.00970* .

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Ranzato, M.; Chopra, S.; Auli, M.; and Zaremba, W. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732* .

Saleh, A.; Jaques, N.; Ghandeharioun, A.; Shen, J. H.; and Picard, R. 2019. Hierarchical reinforcement learning for open-domain dialog. *arXiv preprint arXiv:1909.07547* .

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368* .

Shang, L.; Lu, Z.; and Li, H. 2015. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364* .

Shao, L.; Gouws, S.; Britz, D.; Goldie, A.; Strope, B.; and Kurzweil, R. 2016. Generating long and diverse responses with neural conversation models .

Shen, L.-H.; Tai, P.-L.; Wu, C.-C.; and Lin, S.-D. 2019. Controlling Sequence-to-Sequence Models-A Demonstration on Neural-based Acrostic Generator. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, 43–48.

Song, Y.; Yan, R.; Li, X.; Zhao, D.; and Zhang, M. 2016. Two are better than one: An ensemble of retrieval- and generation-based dialog systems. *arXiv preprint arXiv:1610.07149* .

Sordoni, A.; Galley, M.; Auli, M.; Brockett, C.; Ji, Y.; Mitchell, M.; Nie, J.-Y.; Gao, J.; and Dolan, B. 2015. A neural network approach to context-sensitive generation of

conversational responses. *arXiv preprint arXiv:1506.06714* .

Tang, J.; Zhao, T.; Xiong, C.; Liang, X.; Xing, E. P.; and Hu, Z. 2019. Target-guided open-domain conversation. *arXiv preprint arXiv:1905.11553* .

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Vinyals, O.; and Le, Q. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869* .

Wang, D.; Jojic, N.; Brockett, C.; and Nyberg, E. 2017. Steering output style and topic in neural response generation. *arXiv preprint arXiv:1709.03010* .

Weston, J.; Dinan, E.; and Miller, A. H. 2018. Retrieve and refine: Improved sequence generation models for dialogue. *arXiv preprint arXiv:1808.04776* .

Wu, Y.; Wei, F.; Huang, S.; Wang, Y.; Li, Z.; and Zhou, M. 2019. Response generation by context-aware prototype editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7281–7288.

Wu, Z.; Galley, M.; Brockett, C.; Zhang, Y.; Gao, X.; Quirk, C.; Koncel-Kedziorski, R.; Gao, J.; Hajishirzi, H.; Ostendorf, M.; et al. 2020. A controllable model of grounded response generation. *arXiv preprint arXiv:2005.00613* .

Xing, C.; Wu, W.; Wu, Y.; Liu, J.; Huang, Y.; Zhou, M.; and Ma, W.-Y. 2016. Topic aware neural response generation. *arXiv preprint arXiv:1606.08340* .

Xu, Z.; Liu, B.; Wang, B.; Sun, C.-J.; Wang, X.; Wang, Z.; and Qi, C. 2017. Neural response generation via gan with an approximate embedding layer. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 617–626.

Yan, R.; Song, Y.; and Wu, H. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 55–64.

Yan, R.; Song, Y.; Zhou, X.; and Wu, H. 2016. "Shall I Be Your Chat Companion?" Towards an Online Human-Computer Conversation System. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 649–658.

Yao, L.; Zhang, Y.; Feng, Y.; Zhao, D.; and Yan, R. 2017. Towards implicit content-introducing for generative short-text conversation systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2190–2199.

Zhao, T.; Zhao, R.; and Eskenazi, M. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960* .

8   Appendix

8.1   Training Details

Our experiments consists of two parts: using latent sentences and latent POS sequences.

- Models using Latent Sentence

In the case using latent sentences as semantic information, we utilize GRU network as our fundamental model architecture. For the latent sentence predictor, we adopt 1-layer bi-directional GRU as our sentence encoder of $512 * 2$ dimensions. On top of the encoder is a classifier with 3 fully-connected layers of 512 dimensions. When it comes to Dialogue Generator with Latent Sentence model, the sentence encoder for both input post and latent sentence is a 1-layer bi-directional GRU of $512 * 2$ dimensions. On the decoder phase, 1 layer uni-directional GRU of 512 dimensions is used, with attention mechanism as proposed in Bahdanau, Cho, and Bengio (2014). For those models using GRU network, they are trained with Adam optimizer. In latent sentence predictor, the learning rate is set to 0.002, and learning rate decay is 0.5 within each epoch. On Dialogue Generator part, the learning rate is set to 0.0002, and learning rate decay is also 0.5. During the testing phase, we apply beam-search to generate the output sequence with beam size 4.

- Models using latent POS sequence

For experiment utilizing latent POS sequences, the dialogue generator is a standard $6 - layer$ Transformer encoder-decoder model. We follow the model architecture and the parameter of the base model in Vaswani et al. (2017). To select a latent POS sequence, the predictor model is a 6-layer transformer encoder with a 3-layer classifier the same as the Latent Sentecnce Predictor on top of the last time step of hidden states. To generate a POS sequence, the predictor model is a transformer encoder-decoder model of the same configuration as that of the dialogue generator. All Transformer models are trained with the Adam optimizer. For the pretrained models, we adopt noam learning rate decay as proposed by Vaswani et al. (2017); the warmup step is set to 8000. The initial learning rate is set to $\mathbf{10^{-5}}$ for joint fine-tuning, and is decayed every epoch. During testing, the beam size is set to 3.

8.2   More Results

Here we present more responses from the testing set.

Input	06 月18 日03 时天津4.0 级地震的震中图 The epicenter map of the magnitude 4.0 earthquake in Tianjin at 3 o'clock, June 18.
Seq2Seq	我也想说，我的是谁说的 I also want to say; who said mine?
HGFU	我还以为是我的电脑 I thought it is my computer.
GMDR	我去，这也太厉害了吧 I'll go; this is spectacular.
CVAE	唉… …想起中国的那么多年！ Ah! Think of those many years in China.
Ours(Latent Sentence)	我也感觉到了，有点晃 [Latent Sentence: 我也感觉到了，躺床上有点晃] I have the same feeling too, it's shaking. [Latent Sentence: I have the same feeling too, it's shaking in bed.]
Ours(Generate POS)	我的天啊，我还没感觉。 My god, I haven't felt it.
Ours(Sample POS)	地震了，我还没感觉到。 I didn't recognize the earthquake at that time.

Table 7: Example responses from the testing set

Input	开会终于结束啦，八个半小时。回家 The meetings are finally over; it's eight and a half hours. Going home!
Seq2Seq	我也想说，我也想问一句话。 I also want to say; I want to ask something, too.
HGFU	这是在给广告费做广告么？ Is this paying advertisement charges for advertisement?
GMDR	终于结束了，恭喜！ It finally comes to an end, congratulations!
CVAE	吸血吧，想回来。 Drink some blood, I want to go back.
Ours(Latent Sentence)	辛苦啦！好好休息吧！ [Latent Sentence: 辛苦啦！好好休息啊！] That must be rough. Have a rest. [Latent Sentence: That must be rough. Have a rest.]
Ours(Generate POS)	你这是去哪啊？我也想去。 Where are you going? I want to go too.
Ours(Sample POS)	老师，您辛苦了，注意身体啊！我在等您的微博呢！ Teacher, that must be rough. Be aware of your health! I am waiting for your message on Weibo!

Table 8: Example responses from the testing set

Input	竖着是中国，横着是CHINA！这就是中国书法 It is '中国' (chinese character for China) when we look in one way, and it is 'CHINA' in the other! This is the Chinese calligraphy.
Seq2Seq	我也想说，我也想问一句话。 I also want to say; I want to ask something, too.
HGFU	我怎么觉得是中国的设计 How do I think that it is Chinese design?
GMDR	我喜欢，我喜欢。 I like it; I like it.
CVAE	汇源分析的不错，大家众拿出来开心 The source is analyzed well, everyone takes it out and have some fun.
Ours(Latent Sentence)	这对联至少有了对联 [Latent Sentence: 这对联至少清朝就有了] The couplet possesses a couplet at least. [Latent Sentence: The couplet has already existed since Ching dynasty at least.]
Ours(Generate POS)	中国人的字也太有才了！ The Chinese characters are such brilliant inventions!
Ours(Sample POS)	这是画的吗? 好漂亮啊，我也想要这样的字啊。 Are these painted? They are so beautiful, and I also want characters like these.

Table 9: Example responses from the testing set