Semantic Guidance of Dialogue Generation with Reinforcement Learning

Cheng-Hsun Hsueh
National Yang-Ming University
jimbokururu27@gmail.com

Wei-Yun Ma
Academia Sinica
ma@iis.sinica.edu.tw

Abstract

Neural encoder-decoder models have shown promising performance for human-computer dialogue systems over the past few years. However, due to the maximum-likelihood objective for the decoder, the generated responses are often universal and safe to the point that they lack meaningful information and are no longer relevant to the post. To address this, in this paper, we propose semantic guidance using reinforcement learning to ensure that the generated responses indeed include the given or predicted semantics and that these semantics do not appear repeatedly in the response. Synsets, which comprise sets of manually defined synonyms, are used as the form of assigned semantics. For a given/assigned/predicted synset, only one of its synonyms should appear in the generated response; this constitutes a simple but effective semantic-control mechanism. We conduct both quantitative and qualitative evaluations, which show that the generated responses are not only higher-quality but also reflect the assigned semantic controls.

1 Introduction

Dialogue generation systems with adequate artificial intelligence responses hold great potential for practical use. A decent human-computer dialogue system should generate coherent and informative responses based on human-provided posts (Li et al., 2017). Sequence-to-sequence models (Sutskever et al., 2014) with long-short term memory (Hochreiter and Schmidhuber, 1997) or gated recurrent networks (Cho et al., 2014) have demonstrated profound improvements in open-domain dialogue systems (Shang et al., 2015; Vinyals and Le, 2015; Luan et al., 2016; Xu et al., 2016; Yao et al., 2017). However, these models often generate overly generic responses (Sordoni et al., 2015; Li et al., 2016a) that are independent of the given posts due to the maximum-likelihood-estimation-based objectives.

To improve the variety of the responses, recent studies usually use semantically conditioned LSTM, relying on additional semantic indicators such as keywords to guide the decoding process. However, keywords typically appear repeatedly in generated utterances with this strategy. To address this, Wen et al. (2015b) propose a special gate mechanism to reduce the influence of the keywords. However, since this design does not directly address the concern in the objectives, repeated keywords still often remain a problem in practice; we confirm this is in our experiments.

To address this issue, in this paper, we introduce the semantically controlled and recorded LSTM (SCR-LSTM) cell, which provides semantic guidance via reinforcement learning (RL) as well as a recording mechanism that records the existence of the desired semantics to ensure that the generated responses indeed include the given or predicted semantics; also, the desired semantics are not to appear repeatedly in the response. For the form of the assigned semantics we use synsets, which provide a more flexible semantic representation for practical use, and any lexical or knowledge taxonomy can be used to serve this role. For a given/assigned/predicted synset, only one of its covering synonyms should appear in the generated response.

In addition, when synsets are used to semantically control the generated responses, the responses may indeed show the assigned semantics, but the responses could be not diverse enough, or the relation to the given posts may be tenuous, because the major goal of the model is to meet the semantic constraints. Therefore, we add a conditional SeqGAN (Yu et al., 2017) to assure that the generated responses are similar to true human responses and are related to the given posts while specifying

semantics to avoid dull or repetitive responses.

As with conventional GAN (Goodfellow et al., 2014), our conditional SeqGAN comprises a generator and a discriminator; however, with the proposed discriminator we seek to not only distinguish machine-generated utterances from human-generated utterances but also distinguish post-independent from post-dependent utterances. The resulting additional SeqGAN architecture generates responses that are more related to the posts.

2 Background

2.1 Semantically conditioned LSTM

To incorporate given dialogue acts into utterance generation, Wen et al. (2015b) propose the semantic controlled LSTM (SC LSTM) cell, a special neural cell. The assigned dialogue acts are represented in one-hot form, and are fed into dialogue acts cells, which rely on a decreasing mechanism on dialogue acts information to avoid repetition. The formula for this semantically conditioned LSTM is as following:

$$i_t = \sigma(W_{wi}w_t + W_{hi}h_{t-1}) \quad (1)$$

$$f_t = \sigma(W_{wf}w_t + W_{hf}h_{t-1}) \quad (2)$$

$$o_t = \sigma(W_{wo}w_t + W_{ho}h_{t-1}) \quad (3)$$

$$\hat{c}_t = \tanh(W_{wc}w_t + W_{hc}h_{t-1}) \quad (4)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \hat{c}_t + \tanh(W_{dc}d_t) \quad (5)$$

$$h_t = o_t \otimes \tanh(c_t) \quad (6)$$

With its additional third term, only formula (5) of cell value c_t differs from traditional LSTM. Term d_t serves as the dialogue act one-hot vector, and is derived from the following formula:

$$r_t = \sigma(W_{wr}w_t + \sum_l \alpha_l W^l_{hr} h^l_{t-1}) \quad (7)$$

$$d_t = r_t \otimes d_{t-1} \quad (8)$$

Wen et al. (2015b) term the mechanism based on (7) and (8) a *dialogue act cell* (DA cell). Vector r_t, known as the reading gate, is determined by the input of the current time step and the hidden state of the past generation history, and is multiplied element-wise with the dialogue act vector d_t to either retain or discard its information in future generation time steps.

The monotonically decreasing value of the dialogue act vector is intended to reduce repetition. However, the design provides an insufficient guarantee on avoiding repetition, as the model provides no direct link between the dialogue act generation possibility and the value of d_t; thus repeated keywords continue to remain a problem in practice.

2.2 Sequence GAN

The original generative adversarial network (GAN) is ill-suited to text generation given the discrete nature of text. In particular, the changing-signal guidance from the discriminator does not correspond to discrete dictionary tokens (Yu et al., 2017). Furthermore, the rewards can only be given to entire sequences when the whole generation is finished, making it impossible to estimate the value of a specific token in the generation step. Sequence GAN introduces a policy gradient (Sutton et al., 1999) as well as a rollout mechanism to help the discriminator pass its scores to the generator.

Given a current and incompletely generated response $Y_{1:t} = [y_1, y_2, y_3, \ldots, y_t]$, where t is the current time step of generation and y_t is the token generated at the current step, a reward is to be given to the current token y_t. However, these rewards can be estimated only once the entire sequence has been generated. To account for this, the generator must "roll out" the complete responses at every current step. For example, if we roll out starting from time step t, the complete utterance can be generated using Monte Carlo search as

$$Y^n_{1:T} \in MC^G(Y_{1:t}; N) \quad (9)$$

where MC denotes Monte Carlo search, G denotes the generator, and N denotes the assigned repeating turn for searching. The incomplete responses are completed after the rollout and then judged by the discriminator, which assigns reward scores to the rollouted responses. Rollout is accomplished using N Monte Carlo searches, and the rewards are averaged to serve as the expected utility for the incomplete utterance generated at time step t:

$$V(Y_{1:t}) = \frac{1}{N}\sum_{n=1}^{N} D_\phi(Y^n_{1:T}) \quad (10)$$

where $D_\phi(Y^n_{1:T})$ denotes the score assigned by the discriminator.

2.3 Conditional GAN

Unconditioned GAN loses control on generating the intended type of data. By giving conditions for

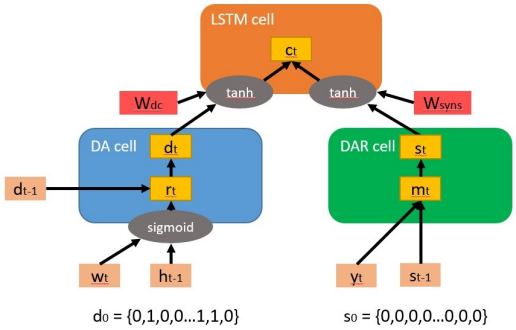

Figure 1: Proposed semantically controlled and recorded LSTM (SCR LSTM) cell.

GAN to depend on (Mirza and Osindero, 2014), it is possible to guide the generation process. Conditional GAN extends the original GAN by providing extra information y for both the generator and the discriminator. The generator conditions on y, whereas the discriminator judges whether the generated data is suitable based on the relatedness of the generated results and the extra information y. Thus, the formula for conditional GAN extends the original GAN with y to become

$$\min_G \max_D V(D,G) = E_{x\sim P_{data}(x)}[\log D(x \mid \mathbf{y})] + E_{z\sim P_z(z)}[\log(1 - D(G(z \mid \mathbf{y})))] \quad (11)$$

where z denotes the generated data and x denotes the training data.

3 Methods

3.1 Semantically Controlled and Recorded LSTM Cell

Extended from (Wen et al. 2015b), we introduce the semantically controlled and recorded LSTM (SCR LSTM) cell, which provides semantic guidance and a recording mechanism, as shown in Figure 1. It integrates a DA cell with a synonym act and a special recording cell which we propose to provide a mechanism to record the existence of the desired semantics. We term this a *dialog act record cell* (DAR cell).

3.2 DA Cell with Synonym Act

The DA cell (Wen et al. 2015b) is integrated in our SCR LSTM cell, but here we slightly change the definition of *act*. We define an act as an element (synonym) of the assigned synset; we expect that just one of the acts (synonyms) will be used in the generation. A one-hot vector is used to encode this synonym act, denoted by d_t, where each element corresponds to a word in the vocabulary, and it is assigned 1 if the corresponding word belongs to the assigned synset. For example, in Figure 2, given an assigned synset - synset$_i$, there are three vocabulary elements (synonyms)—'tribe', 'group', and 'clan'—thus the vocabulary-size d_t is represented as $[0..0, 1, 0..0, 1, 0..0, 1, 0..0]$, in which the three 1s refer to 'tribe', 'group', and 'clan', respectively. Value d_t is fed into the DA cell, which relies on a decreasing mechanism for dialogue act information to prevent repetition, as shown in (7) and (8).

Note that although the three synonyms are all indexed in d_t, this does not mean that all three synonyms (dialogue acts) are to appear in the response. Instead, we expect only one of these to appear, in fact to appear exactly once, in contrast to (Wen et al. 2015b). However, the DA cell merely decays the influence of the assigned dialogue acts, and does not directly address this concern in the objectives; thus repeated keywords still remain a problem in practice, as we verify experimentally. To address this shortcoming, we propose the dialog act record cell (DAR cell) in concert with the DA cell.

3.3 DAR Cell with Synset Act

With the DAR cell we seek to provide a mechanism to record the occurrence of the desired semantics[1] to ensure that they are indeed included in the generated responses. At every generation time step we use a one-hot vector s_t, the dimension of which is the total number of synsets, to record whether the assigned synsets appear. Each element of s_t indicates whether the synset appears or not. For generation, s_t is initialized as $[0..0]$. Once an element of the assigned synset appears during generation, the synset's corresponding element in s_t is changed to 1. We develop a special gate called an $MGate$ to realize this function, which is formally presented in Algorithm 1.

Figure 2 illustrates the overall structure. Given an assigned synset - synset$_i$, there are three vocabulary elements (synonyms): 'tribe', 'group', and 'clan'. As generation proceeds, at the second time step, as 'tribe' is generated, s_t is updated from [0..0,0,0..0] to [0..0,1,0..0], in which 1 refers to synset$_i$'s current status. The updated s_t informs the model that synset$_i$ has already appeared, instructing it to not generate any element of synset$_i$ afterward.

[1] In our model, the desired semantics can be multiple synsets or a single synset. All of our experiments are based on a single synset.

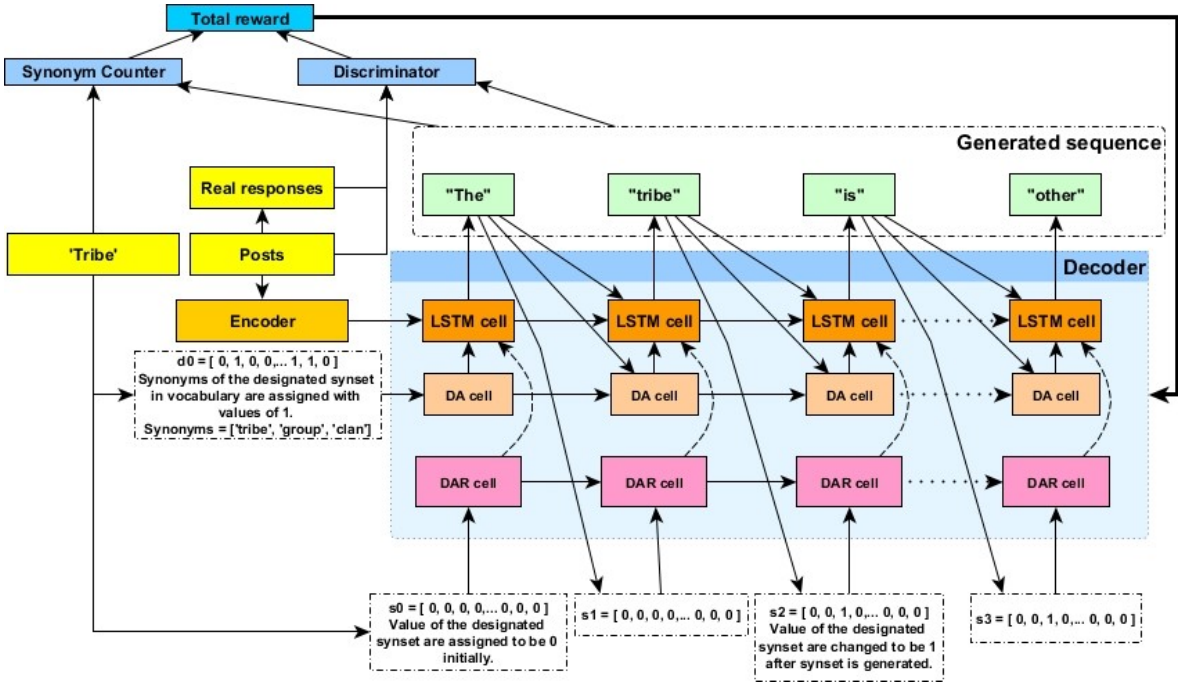

Figure 2: Overall view of the model. The decoder incorporates synset information through additional DAR cells, which retain or discard synset information in every generation step based on whether the assigned synset has appeared or not.

Algorithm 1 M Gate Algorithm.

Input: s_{t-1} and y_{t-1} (y refers to generated token)
Output: s_t

1: $s_t = s_{t-1}$
2: **for** each synset$_i \in$ assigned synset **do**
3: **if** $y_{t-1} \in$ synset$_i$ **then**
4: $s_t(i) = 1$ //$s_t(i)$ refers to i-th element of s_t
5: **end if**
6: **end for**

The SCR LSTM cell value c_t integrating the DA cell and DAR cell is

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \hat{c}_t + \tanh(W_{dc} d_t) + \tanh(W_{syns} s_t) \quad (12)$$

where $W_{dc} d_t$ and $W_{syns}\ s_t$ are the outputs of the DA and DAR cell, respectively, and $W_{dc} \in R^{h \times d_1}$, $W_{syns} \in R^{h \times d_2}$, $d_t \in \{0,1\}^{d_1}$, and $s_t \in \{0,1\}^{d_2}$. Value d_1 denotes the vocabulary size, d_2 denotes the total number of synsets, and h denotes the dimension of hidden states in the decoder.

To both prevent repetition and ensure the desired semantics in the generated responses, we use reinforcement learning to penalize our model for violations. The reward is

$$C_{syn} = 1 - |\text{semantic occurence} - 1| \quad (13)$$

where semantic occurrence is an integer that records the current occurrence (frequency) of the elements of the assigned synset at every time step of the generation. Thus we expect that when the generation is finished, semantic occurrence will be exactly 1 instead of a number greater than 1, indicating repetition of the desired semantics, or 0, indicating the absence of the desired semantics.

Thus only a semantic occurrence of 1 results in the highest value of 1 for C^{syn}; a semantic occurrence of 0 and a semantic occurrence greater than 1 cause C^{syn} to be less than or equal to 0.

Although this reward encourages appropriate appearances of the assigned synsets in the response, it could cause the model ignore other critical requirements for a response, including fluency, relevance to the posts, and information. To account for this, we add a conditional SeqGAN to provide another reward D_ϕ , which is the result of its discriminator, seeking to ensure that the generated responses approximate true human responses and are related to the given posts.

The discriminator not only distinguishes machine-generated utterances from human-generated utterances but also distinguishes

post-independent from post-dependent utterances. D_ϕ derives its score by projecting the concatenated final hidden states of two LSTM sequence-to-sequence networks to a 2-dimensional vector followed by softmax. The first LSTM network is given posts as encoder inputs and responses as decoder inputs, whereas the second network switches posts and responses. Therefore, the discriminator model can be formulated as

$$D_\phi(p,q) = \text{softmax}\left(W^D[h1^{final}_{p|q}; h2^{final}_{q|p}]\right) \quad (14)$$

where p denotes post, q denotes response, W^D denotes the projection matrix, and h_1 and h_2 denote two sequence-to-sequence networks respectively. During training, a third of the training batches are pairs composed of posts with their correlated human responses, another third is composed of pairs of posts with an uncorrelated human response, and the final third is pairs of posts with a generated response. Only the first third is labeled *true*; the other two-thirds are labeled *false*.

For every generation step, the expected utility V is given by both the semantic occurrence and the discriminator, calculated using Monte Carlo search as

$$V(p, Y_{1:t}) = \frac{1}{N}\sum_{n=1}^{N} D_\phi(p, Y^n_{1:T}) + C^{syn}(Y^n_{1:T}),$$
$$Y^n_{1:T} \in MC^G(Y_{1:t}, N) \quad (15)$$

where the notation p denotes the post, $Y_{1:t} = [y_1, y_2, y_3, \ldots, y_t]$ denotes the generated sequence, and G denotes the generator. N is the number of turns in the Monte Carlo search, here set to 5. The utility is then applied in the REINFORCE algorithm (Williams, 1992) as

$$\nabla J(\theta) \approx$$
$$\sum_t (V(p, Y_{1:t}) - b)\nabla \log p(y_t \mid x, Y_{1:t-1}) \quad (16)$$

where b denotes the baseline value to reduce the variance of the utility.

4 Evaluation

4.1 Dataset

Conversation data from Weibo was used for training and evaluation. The training data is composed of 570k post-response pairs with 3360 synonym sets, and the testing data is composed of 2k post-response pairs with 1731 synonym sets.

Here we established two datasets, as shown in Figure 3. In the first dataset, we attempted to eliminate interference from semantic selection and focus mainly on the effect of the model. Therefore, we fetched the assigned semantics from human response by randomly selecting one synset from the human response. In the first dataset, we used both human evaluation and automatic evaluation to analyze the efficacy of our model. Thereafter

Algorithm 2 Training Algorithm.

Input: (post, response) pairs with assigned synsets

```
Initialize generator and discriminator
Pre-train generator G using maximum likelihood estimation
repeat
    Generator G generates response Y_{1:T} given post and assigned synset
    for t ∈ range(T) do
        for n ∈ range(N) do // N is turns of MC search
            s_t ← M-Gate(y_t^n, s_{t-1})
            Roll out Y_{1:t}^n to full sentence Y_{1:T}^n
        end for
        Calculated the expected utility of Y_{1:t} by equation(15)
    end for
    Update generator G
    Update discriminator D
until reinforcement learning converges
```

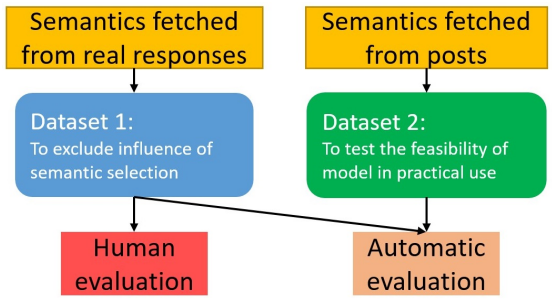

Figure 3: The two datasets in experiment.

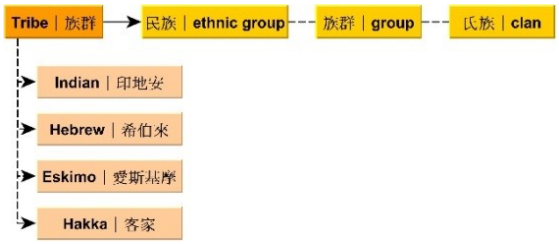

Figure 4: Structure of E-HowNet.

we switched to the second dataset, where the assigned semantics are simply fetched from posts by randomly selecting one synset from a post. We analyze the feasibility of our model in practical use. Automatic evaluations are also performed for the second dataset.

The synonyms of an assigned synset are retrieved from E-hownet (Ma and Shih, 2018; Chen et al. 2005), a structured Chinese knowledge net. The structure of E-hownet is shown in Figure 4. The synonyms of an assigned word are at the same level of the word, whereas meanings of a word are inferior to the word.

For the experiments here we fetch only the synonyms. Note that our model is not confined to E-hownet; other synonym datasets could be used for our proposed model as well.

4.2 Baselines

SEQ2SEQ

The Sequence-to-sequence model (Sutskever et al., 2014) with an attention mechanism (Bahdanau et al., 2014) is implemented without auxiliary keywords.

Hierarchical Gated Fusion Unit (HGFU)

HGFU (Yao et al., 2017) incorporates assigned keywords into sequence generation. We replace the keyword input with the synset to focus the comparison on the model design and ensure a fair comparison.

Semantically conditioned LSTM (SC-LSTM)

Wen et al. (2015b) use dialogue acts cells to generate utterances that contain the assigned dialogue acts. Here we replace the dialogue acts with synsets for comparison. In addition, for a full comparison, we implement SC-LSTM with over-generation, as suggested by Wen et al. 2015a, generating 20 sequences and selecting the top-scoring one.

4.3 Proposed method

SCR-LSTM + RL

This approach extends the former method using an RL mechanism and an additional DAR cell to record whether the synonym set has already been generated in previous generation steps (Section 3.1).

The proposed methods and baselines all leverage beam search with a beam size of 5 to generate appropriate responses. Only the top-scored sequences are selected for further evaluation.

Methods	**Average score**
SEQ2SEQ	0.99
HGFU	1.19
SC-LSTM (over generation)	1.06
SCR-LSTM+RL	1.23

Table 1: Scores of different models from human evaluation

Situation	**Percentage**
HGFU win	24.65%
Tie	42.71%
SCR-LSTM + RL win	32.64%

Table 2: Comparison between HGFU and SCR-LSTM + reinforcement learning

Methods	**Repetition percentage**	**Non-appearing percentage**
HGFU	22.99%	4.00%
SC-LSTM (over generation)	3.05%	5.55%
SCR-LSTM+RL	2.10%	0.25%

Table 3: Percentage of synonym repetition and non-appearing when semantics are fetched from responses

Methods	**Repetition percentage**	**Non-appearing percentage**
HGFU	31.03%	4.05%
SC-LSTM (over generation)	2.90%	5.55%
SC-LSTM (MLE)	13.10%	1.25%
SCR-LSTM+RL (w/o discriminator)	2.20%	0.50%
SCR-LSTM+RL (w/ discriminator)	4.15%	0.70%

Table 4: Percentage of synonym repetition and non-appearing when semantics are fetched from posts

Methods	**Uni -gram**	**Bi -gram**	**Tri -gram**
SEQ2SEQ	3.20%	10.81%	19.15%
HGFU	7.11%	23.99%	39.87%
SC-LSTM (over generation)	15.02%	61.47%	89.85%
SC-LSTM (MLE)	7.60%	22.33%	36.53%
SCR-LSTM+RL (w/o discriminator)	8.73%	25.10%	39.80%
SCR-LSTM+RL (w/ discriminator)	8.91%	26.17%	42.21%

Table 5: Unigram, bigram and trigram variations of different methods

Methods	Example 1	Example 2
	Post: 好...我抱着一丝期待等答案公布啊！ **Post:** OK...I am anticipating the answer! **Synset:** 希望、盼、盼望、期待、期盼 (anticipate)	**Post:** 因为我不会画，所以就被丑化啦 **Post:** I am defamed because I can't draw **Synset:** 没法、不能、无法、不会、不可 (cannot)
SEQ2SEQ	好的，谢谢你的支持 Ok, thank you for your support	我画的是你画的 I drew what you drew
HGFU	你说的是什么? What are you saying?	不会吧 No way
SC-LSTM (over generation)	不用期待了吗? Not looking forward to it?	人生才不会造成吧? Life wouldn't cause that, would it?
SCR-LSTM (MLE)	好的，期待！ Okay, looking forward to it!	不会吧，我也很喜欢 No, I like it too
SCR-LSTM + RL (w/o discriminator)	期待中！ Looking forward to it!	不会吧 No way
SCR-LSTM + RL (w/ discriminator)	**好的，敬请期待** **OK, stay tuned**	**不会吧，我也觉得挺好看的** **No, I think it's pretty good too**

Table 6: Examples of different models

Methods	Example 1	Example 2
	Post: 是反话啊，很明显...哈哈 **Post:** It's ironic, obviously ... haha **Synset:** 向、是 (is)	**Post:** 哈哈，每道菜我都要品尝! **Post:** Haha, I will try every dish! **Synset:** 品尝、尝 (taste)
SCR-LSTM + RL (w/o discriminator)	是的，我也很喜欢他的 Yes, I like him too	尝了吧! Taste it!
SCR-LSTM + RL (w discriminator)	**是啊，我也觉得很搞笑** **Yeah, I also find it funny**	**欢迎您来品尝!** **You are welcome to have a taste!**

Table 7: Examples SCR-LSTM with and without discriminator

Methods	SCR-LSTM (w/ discriminator)
Example 1	**Post:** 抚州娃发来贺电，南昌新年好 **Post:** Greetings from the baby in Fuzhou, happy new year in Nanchang **Synset:** 娃、小子、孩子、孩儿 (kids, baby) **Response:** 谢谢!孩子们! **Response:** Thanks, kids!
Example 2	**Post:** 讲什么的，育儿? **Post:** What is it about? Raising child? **Synset:** 说、说话、讲、讲话(say) **Response:** 你说的是什么 **Response:** What do you mean?

Table 8: Synsets help to extend semantics

4.4 Results and analysis

Human evaluation

Since automatic metrics such as the BLEU score or perplexity are not suitable in evaluating dialogue generation (Shang et al., 2015), we used human judgments instead. The criteria of human evaluation are referenced from Shang et al. (2015) with three levels: unsuitable, neutral, and suitable. To be judged 'suitable', the response must be clearly correlated to the post and must be natural. For 'neutral', the response can be suitable in a specific scenario. The response is considered 'unsuitable' if it does not fit in any scenario provided by the post. Scores of 0, 1, and 2 were given for the three levels respectively. Four methods for comparison were evaluated, with 230 generated responses each. Every generated response was evaluated by three people using Amazon Turk. As mentioned above, the semantics for this part of data were fetched from real human responses.

Table 1 demonstrates that SCR-LSTM + RL receives the highest score and HGFU ranks second. To further compare the two methods, 96 posts and generated responses from the two methods were compared directly, with ties allowed. Table 2 shows that the proposed method still outperforms HGFU.

Also note that the proposed model outperforms SEQ2SEQ, which does not rely on extra semantic guidance, demonstrating that semantic guidance plays an important role in generating meaningful and related sequences given the post.

Automatic evaluation

To further evaluate the effect of the proposed model, we implemented automatic evaluations. We also calculate the percentage of semantic repetition and non-appearance. Table 3 shows that when semantics are fetched from human responses, SCR-LSTM + RL generates sequences with the least semantic repetition and absence. For dataset 1, both human evaluation and automatic evaluation prove that with semantic selection, the proposed model generates natural responses with the assigned semantics appearing only once.

To further evaluate the feasibility of our model in practical use, we shift to dataset 2, in which semantics are fetched from posts. We evaluate the effect of reinforcement learning and the discriminator, respectively, using three methods: SCR-LSTM trained with maximum-likelihood-estimation without RL (SCR-LSTM MLE), SCR-LSTM trained with synset occurrences during reinforcement learning but without the discriminator (SCR-LSTM + RL w/o discriminator), and SCR-LSTM trained with synset occurrences and the discriminator (SCR-LSTM + RL w/ discriminator), respectively.

We implement as automatic methods the percentage of semantic repetition and that of non-appearance. Table 4 shows that SCR-LSTM + RL both with and without discriminator methods generate less semantic repetition and absence than SCR-LSTM MLE. This shows that reinforcement learning with the target of single-appearance semantics has achieved its goal. SCR-LSTM+RL without the discriminator, which is trained using only synset occurrences as a reward, reduces semantic repetition and absence even more, resulting in the best performance in Table 4. In addition, SCR-LSTM MLE also results in significantly less semantic repetition and fewer absences than HGFU, showing that the proposed SCR-LSTM design alone is enough to induce the desired semantics to appear just once.

Another metric is the percentage of distinct unigrams, bigrams, and trigrams. Proposed by Li et al. 2016b, this quantifies the diversity of a generated sequence. This metric is calculated by counting the distinct unigrams, bigrams, and trigrams, and divided this by the total number of unigrams, bigrams, and trigrams respectively. Table 5 shows that SCR-LSTM + RL with the discriminator achieves higher distinct unigram, bigram, and trigram percentages than SCR-LSTM + RL without the discriminator. Thus the discriminator does help the reinforcement learn to generate more diverse responses. Note that the over-generation of SC-LSTM yields the highest diversity because the model generates words randomly and thus has a higher possibility to pick up non-frequent words. Table 6 contains examples from different models.

Case study: Effect of the discriminator

The effect of the discriminator is seen in Table 7, which compares SCR-LSTM + RL with and without the discriminator. In the first example, SCR-LSTM + RL w/o discriminator generates a sequence that is not correlated with the given post. SCR-LSTM + RL w/ discriminator generates a better sequence that is relevant to the post. For the second example, both methods generate relevant sequences to the post, but SCR-LSTM + RL w/o discriminator generates a sequence that is too short and not very informative while the LSTM + RL w/ discriminator generates a sequence that is more meaningful and diverse.

Case study: Semantic coverage

With the synset implementation we seek to extend the semantic coverage of the desired keywords. In Table 8, keywords from posts are not directly used when generating responses. Instead, the synonyms of the keywords are used as extra information during the generation process. This shows that a particular synonym may be used as semantic guidance in generated responses.

5 Conclusion

In this work, to develop an effective semantic control mechanism, we propose the SCR-LSTM model with reinforcement learning to ensure that the desired semantics appear once and do not repeat. We also present a conditional SeqGAN to help generate more coherent and informative responses. Results from both human and automatic evaluations show that the proposed models outperform other baselines and achieve the lowest repetition and absence percentages of the assigned synsets in the generated responses, proving that the proposed approach indeed produces high-quality responses under the desired semantic control. Also, we prove that SeqGAN is an essential part of enabling the model to generate more diverse and coherent responses.

The proposed model leverages synsets to serve as the semantic guidance. To investigate the feasibility of our model in practical use, in this work, the assigned synsets are simply fetched from posts. However, the selection or prediction of the desired

semantics is an interesting task that we leave to future study.

References

K. Cho, B.V. Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio. 2014. Learning phrase representations using rnn encoder- decoder for statistical machine translation. In *EMNLP*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. In *Neural Computation*, 9, pages 1735–1780.

Chen Keh-Jiann, Huang Shu-Ling, Shih Yueh-Yin, , and Chen Yi-Jun. 2005. Extended-hownet - a representational framework for concepts. In *OntoLex 2005 - Ontologies and Lexical Resources IJCNLP-05 Workshop*.

J. Li, M. Galley, C. Brockett, J. Gao, and W.B. Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *HLT- NAACL*.

J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. In *EMNLP*.

J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *EMNLP*.

Y. Luan, Y. Ji, and M. Ostendorf. 2016. Lstm based conversation models. In *CoRR*.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. In *CoRR*.

L. Shang, Z. Lu, and H. Li. 2015. Neural responding machine for short-text conversation. In *ACL*.

A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J. Nie, J. Gao, and W.B. Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *HLT-NAACL*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, page 3104–3112.

R.S. Sutton, D.A. McAllester, S.P. Singh, and Y. Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*.

O. Vinyals and Q.V. Le. 2015. A neural conversational model. In *CoRR*.

Ma Wei-Yun and Shih Yueh-Yin. 2018. Extended hownet 2.0 – an entity-relation common-sense representation model. In *LREC*.

T. Wen, M. Gasic, D. Kim, N. Mrksic, P. Su, D. Vandyke, and S.J. Young. 2015a. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *EMNLP*.

T. Wen, M. Gasic, D. Kim, N. Mrksic, P. Su, D. Vandyke, and S.J. Young. 2015b. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. In *SIGDIAL*.

Ronald J Williams. 1992. Simple statistical gradientfollowing algorithms for connectionist reinforcement learning. In *Machine Learning 8 (3-4)*, page 229–256.

Z. Xu, B. Liu, B. Wang, C. Sun, and X. Wang. 2016. Incorporating loose-structured knowledge into lstm with recall gate for conversation modeling. In *CoRR*.

L. Yao, Y. Zhang, Y. Feng, D. Zhao, , and R. Yan. 2017. Towards implicit content-introducing for generative short-text conversation systems. In *EMNLP*.

L. Yu, W. Zhang, J. Wang, and Y. Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*.