Multi-task Learning for Natural Language Generation in Task-Oriented Dialogue

Chenguang Zhu **Michael Zeng** **Xuedong Huang**

Microsoft Speech and Dialogue Research Group, Redmond, WA, USA
{chezhu, nzeng, xdh}@microsoft.com

Abstract

In task-oriented dialogues, Natural Language Generation (NLG) is the final and crucial step to produce user-facing system utterances. The result of NLG is directly related to the perceived quality and usability of a dialogue system. While most existing systems provide semantically correct responses given goals to present, they struggle to match the variation and fluency in the human language. In this paper, we propose a novel multi-task learning framework, NLG-LM, for natural language generation. In addition to generating high-quality responses conveying the required information, it also explicitly targets for naturalness in generated responses via an unconditioned language model. This can significantly improve the learning of style and variation in human language. Empirical results show that this multi-task learning framework outperforms previous models across multiple datasets. For example, it improves the previous best BLEU score on the E2E-NLG dataset by 2.2%, and on the Laptop dataset by 6.1%.

1 Introduction

Natural Language Generation (NLG) is the final procedure in the pipeline of task-oriented dialogues. As the result of NLG is directly facing users, its readability and informativeness have a direct impact on users' perception of the entire dialogue system. On one hand, the response must contain the desired information, referred to as *meaning representation* (MR), in order to provide or request a user's information. On the other hand, the system response needs to mimic the fluency and variation in human language to improve the user experience. To this end, there have been numerous studies on methods to generate natural responses for task-driven dialogues.

Early work primarily employ predefined rules or syntax (Cheyer and Guzzoni, 2014; Langkilde and Knight, 1998). Though these frameworks can provide adequate information, their lack of naturalness and variation in language make the response rather rigid. Moreover, these methods usually require non-trivial manual work to create templates, rendering them unscalable across domains.

Recently, corpus-based methods have gained considerable popularity in natural language generation (Wen et al., 2015a,b; Dušek and Jurčíček, 2016). With the increasing availability of rich dialogue task data, corpus-based frameworks design end-to-end trainable systems. With minimum human effort, these methods directly learn the pattern and styles of human responses from the data, while conveying the required task-specific meaning representation information. Furthermore, the booming of deep learning technology in natural language processing increases these models' capacity to generate sophisticated human-like responses. For instance, Dušek and Jurčíček (2016) employs sequence-to-sequence structure and attention mechanism to generate response tokens from the MR sequence. Wen et al. (2015b) uses a semantic control vector integrated into an LSTM to guide the response generation process. Li et al. (2015) uses maximum mutual information as objective function to generate diverse and appropriate responses. Wen et al. (2016) proposes data counterfeiting to reduce the complexity of transferring trained parameters across multiple domains. However, it still remains a challenge in task-oriented dialogue systems to generate truly natural utterance indistinguishable from a human's response.

On the other hand, language modeling is a technique typically employed to learn language patterns from text. It has been successfully used to generate natural and semantically sound utterances for text summarization, speech recognition and other NLP tasks (Roark et al., 2004; Rush

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 1261–1266,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

et al., 2015). As task-oriented dialogue datasets usually contain rich human responses, leveraging language modeling has a great potential to boost an NLG model's capacity to mimic human language.

Due to recent successes of multi-task learning in NLP (Collobert et al., 2011; Xu et al., 2018), we propose a multi-task scheme to tackle natural language generation in task-oriented dialogues. For the NLG task, we employ a sequence-to-sequence framework. The decoder uses an attention mechanism to carry over information from the encoder on an MR sequence. Therefore, the NLG task generates response conditioned on input MR.

The primary contribution of our work is to incorporate a language modeling task on human-generated responses as an *unconditioned* complementary process that brings in more language-related elements, without the intervention of required MR information. Furthermore, the unsupervised nature of language modeling means we do not need additional labelled data. Thus, under multi-task learning framework, we simultaneously train the NLG and language modeling tasks. To facilitate multi-task learning, we carry out language modeling task in decoder and it partially shares parameters from the NLG task.

To evaluate the effectiveness of our model, NLG-LM, we conduct evaluation on 5 task-oriented dialogue NLG tasks: E2E-NLG (Novikova et al., 2017), TV, Laptop, Hotel and the Restaurant datasets (Wen et al., 2015b, 2016). NLG-LM achieves new state-of-the-art results on all 5 datasets. For example, it outperforms Slug (Juraska et al., 2018), the best model in E2E-NLG competition, by 2.2% in BLEU score. Ablation studies show that the introduction of language modeling task during training can improve the result by 2.4% in BLEU score on average.

2 Problem Description

In task-oriented dialogues, the natural language generation (NLG) process is to produce system utterances as natural language, given system-generated meaning representations from previous steps in the pipeline. Each MR is a slot-value pair, where the slot indicates the category of the information to convey and the value represents the content. For example, *(area, city south)* is a meaning representation and the corresponding utterance should indicate city south as area information.

In addition to meaning representation, *dialogue acts* (DA) are given to differentiate between different types of system actions. Typical examples of dialogue acts include *inform*, *request* and *confirm*. For a given meaning representation, the NLG process should generate different utterances for different dialogue acts. For instance, *confirm* dialogue act usually leads to system response starting with "Let me confirm" or "Correct me if I'm wrong".

In task-oriented dialogues, NLG is framed as a supervised learning problem. Given training data $\{(d_i, r_i, u_i)\}$, where d_i is the dialogue act, $r_i = \{(s_1, v_1), (s_2, v_2), ..., (s_k, v_k)\}$ is the set of meaning representations, and u_i is a sample utterance generated by human labellers, the goal is to generate utterance u given a new pair of dialogue act d and meaning representations r.

As certain types of meaning representation contain entities like location names and product types that are usually proper nouns, we use the *delexicalization* technique to replace values with a special slot token, ⟨slot name⟩, during training and generation. The ultimate response is obtained via a reversal *lexicalization* process to replace slot tokens with their corresponding values.

3 Model

3.1 The NLG task

We approach the NLG problem using the sequence-to-sequence method (Sutskever et al., 2014). Compared with SC-LSTM (Wen et al., 2015b), this method does not need to create additional one-hot MR vector, and can be much more easily extended across different domains with varying meaning representations.

We first concatenate dialogue act d and meaning representations r as a single input sequence $\mathcal{I}$ with m tokens. The output sequence $\mathcal{O}$ is similarly obtained from the given utterance u, with n tokens. Both sequences are delexicalized. We put special sentence tokens ⟨BOS⟩ and ⟨EOS⟩ around each sequence.

The goal is to generate output tokens one at a time, given previously predicted tokens and the input sequence. This can be modeled as maximizing the conditional probability distribution:

$$p(w_1, ..., w_n|\mathcal{I}) = \prod_{t=1}^{n} p(w_t|w_1, ..., w_{t-1}; \mathcal{I}) \quad (1)$$

To do this, we employ the encoder-decoder method.

Encoder. We train a dictionary $\mathcal{D}$ to map each token to a fixed-length vector of dimension d. The input embedding sequence then goes into a layer of bidirectional RNN to produce contextualized embeddings. We use GRU (Cho et al., 2014) as the RNN unit and sum up the forward and backward RNN outputs. The output of the encoder is denoted by $(\boldsymbol{u}_1, ..., \boldsymbol{u}_m) \in \mathbb{R}^{d_h \times m}$, where d_h is the RNN's output dimension and m is its input sequence length.

Decoder. The decoder employs an RNN with an attention mechanism to generate tokens one at a time. It starts with the beginning-of-sentence token and uses the final hidden state from encoder RNN as the initial hidden state. In the t-th step, we use the same dictionary $\mathcal{D}$ from the encoder to map the t-th output token into vector $\boldsymbol{s}_t$ and apply dropout. Then, given the previous hidden state $\boldsymbol{h}_{t-1}$, the decoder first computes attention weights over encoder outputs:

$$\boldsymbol{v}_i = [\boldsymbol{h}_{t-1}; \boldsymbol{u}_i] \in \mathbb{R}^{2d_h} \quad (2)$$

$$\boldsymbol{e}_i = \text{softmax}(\boldsymbol{W}_1 \boldsymbol{v}_i) \in \mathbb{R}^{d_h} \quad (3)$$

$$\alpha_i = \text{ReLU}(\boldsymbol{b}^T \boldsymbol{e}_i) \in \mathbb{R} \quad (4)$$

Here, $\boldsymbol{W}_1 \in \mathbb{R}^{d_h \times 2d_h}$ and $\boldsymbol{b} \in \mathbb{R}^{d_h}$ are parameters. The weights $\{\alpha_i\}_{i=1}^m$ are then applied to encoder outputs to obtain the context vector $\boldsymbol{c} = \sum_{1 \le i \le m} \alpha_i \boldsymbol{u}_i$.

The context vector $\boldsymbol{c}$ and embedded vector $\boldsymbol{s}_t$ are then concatenated and sent into decoder GRU with output $\boldsymbol{g} \in \mathbb{R}^{d_h}$ and a new hidden state $\boldsymbol{h}_t \in \mathbb{R}^{d_h}$.

To generate the next token, we reuse the dictionary $\mathcal{D}$ with its transposed weights $\boldsymbol{W}_{\mathcal{D}}$[1]. We again integrate the context vector to fuse in contextual information:

$$\boldsymbol{o} = \boldsymbol{W}_2[\boldsymbol{g}; \boldsymbol{c}] \in \mathbb{R}^d \quad (5)$$

$$\boldsymbol{p}_t = \boldsymbol{W}_{\mathcal{D}} \boldsymbol{o} \in \mathbb{R}^{|V|} \quad (6)$$

where $\boldsymbol{W}_2 \in \mathbb{R}^{d \times 2d_h}$ is a parametrized matrix. $\boldsymbol{p}_t$ is the probability distribution of the next token over all tokens in dictionary.

The loss function is cross entropy. Suppose the one-hot vector for the ground-truth at t-th step is $\boldsymbol{y}_t$, then the loss function for each training sample sequence pair is:

$$\mathcal{L}^{NLG}(\theta) = -\sum_{t=1}^{n} \boldsymbol{y}_t^T \log(\boldsymbol{p}_t) \quad (7)$$

3.2 Coupling with Language Model

The encoder-decoder approach above incorporates the information from dialogue act and meaning representation at each step via attention. However, due to this mechanism, the generated utterance inevitably relies to a great extent on the input sequence, focusing less on the fluency and variation of human language, which is as important as conveying the required information in task-oriented dialogues.

On the other hand, language modeling is typically used to characterize the naturalness of words, phrases and sentences. A well-trained language model can assign natural and semantically sound utterance higher scores than rigid and unnatural sentences. In deep learning, the language model task is often solved by a recurrent neural network. However, instead of depending on an input sequence like in Equation (1), the probability of the next token in language models only relies on preceding words:

$$p^{LM}(w_1, ..., w_n) = \prod_{t=1}^{n} p^{LM}(w_t | w_1, ..., w_{t-1}) \quad (8)$$

We propose that by integrating language modelling into the NLG process as an additional objective, the generated sentences will better approximate the styles and variation in human response.

To do this, we add another GRU unit, GRU^{LM}, to the decoder, that has its own hidden state $\boldsymbol{h}_{t-1}^{LM}$ and takes the embedded vector $\boldsymbol{s}_t$ as input. The output is $\boldsymbol{g}^{LM}$ and the new hidden state is $\boldsymbol{h}_t^{LM}$. The probability distribution of next token in language model is:

$$\boldsymbol{o}^{LM} = \boldsymbol{W}_2[: d_h]\boldsymbol{g}^{LM} \in \mathbb{R}^d \quad (9)$$

$$\boldsymbol{p}_t^{LM} = \boldsymbol{W}_{\mathcal{D}} \boldsymbol{o}^{LM} \in \mathbb{R}^{|V|} \quad (10)$$

where $\boldsymbol{W}_2[: d_h]$ are the first d_h columns of $\boldsymbol{W}_2$. As we can see, the context $\boldsymbol{c}$ does not affect the probability computation for language modeling. The loss function of language model is:

$$\mathcal{L}^{LM}(\theta) = -\sum_{t=1}^{n} \boldsymbol{y}_t^T \log(\boldsymbol{p}_t^{LM}) \quad (11)$$

[1] Suppose $\mathcal{D} \in \mathbb{R}^{|V| \times d}$, its transposed weights $\boldsymbol{W}_{\mathcal{D}} \in \mathbb{R}^{d \times |V|}$.

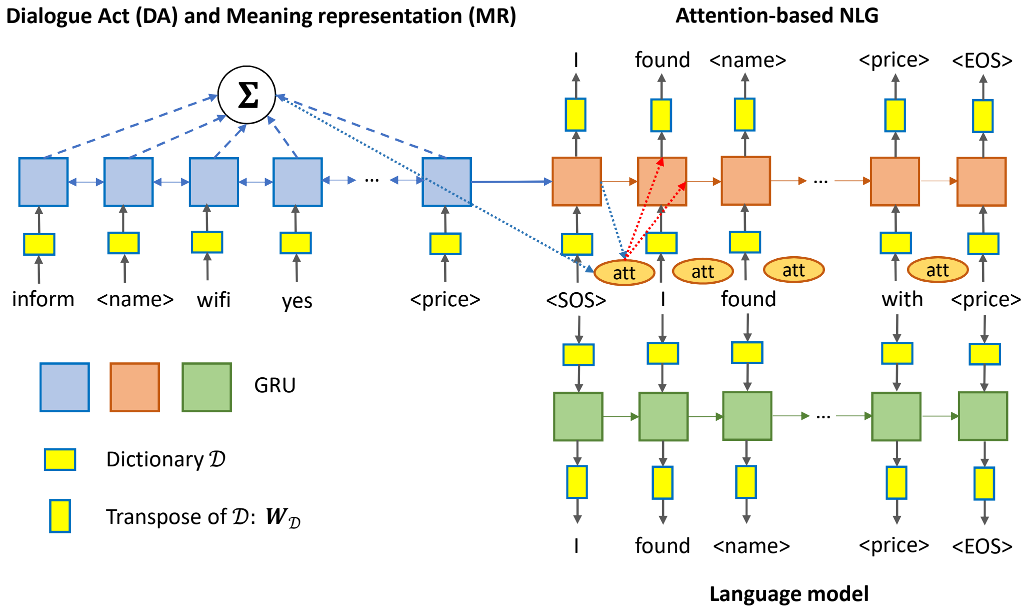

Figure 1: NLG-LM model structure. The encoder takes dialogue act (DA) and meaning representations (MR) as input. The decoder for NLG applies attention on encoder outputs, and the decoder for language modeling shares dictionary with NLG but uses a separate GRU unit.

Model	**E**	**TV**	**L**	**H**	**R**
TGen	0.659	/	/	/	/
Slug	0.662	0.529	0.524	/	/
SCLSTM	/	0.527	0.512	0.848	0.752
RALSTM	/	0.541	0.525	0.898	0.779
w/o LM	0.673	0.609	0.560	0.927	0.734
NLG-LM	**0.684**	**0.617**	**0.586**	**0.939**	**0.795**

Table 1: BLEU scores on E2E-NLG (**E**), TV, Laptop (**L**), Hotel (**H**) and Restaurant (**R**) *testset*. w/o LM is our model without language model task.

Model	**TGen**	**Slug**	**w/o LM**	**NLG-LM**
NIST	8.609	8.613	8.581	**8.626**

Table 2: NIST scores on E2E-NLG *testset*.

Finally, we linearly combine the two loss functions into a single multi-task loss function:

$$\mathcal{L}(\theta) = \mathcal{L}^{NLG}(\theta) + \alpha\mathcal{L}^{LM}(\theta) \qquad (12)$$

We depict our model structure in Figure 1. As shown, the dictionary $\mathcal{D}$ is shared between the NLG task and language modeling task.

4 Experiments

4.1 Datasets and settings

We evaluated the models on five datasets from different domains, covering restaurant booking, hotel booking and retail. The largest dataset is from the E2E-NLG task (Novikova et al., 2017), consisting of 51.2K MR-utterance pairs in the restaurant domain. We also use the four datasets from RNN-LG (Wen et al., 2016), including dialogue scenarios in TV retails, laptop retails, hotel-booking and restaurant-booking domains, with 14.1K, 26.5K, 8.7K and 8.5K samples respectively.

For fairness, we use the official evaluation scripts from E2E-NLG and RNN-LG (Wen et al., 2016) to assess models. We use the BLEU-4 (Papineni et al., 2002) and NIST (Przybocki et al., 2009) metrics.

Delexicalization. In the experiment, we do not delexicalize slots that have binary values or are inappropriate for verbatim substitution. For instance, in E2E-NLG datasets, we only delexicalize *name* and *near* slots. For TV dataset, we delexicalize all slots except *hasusbport*. In Laptop dataset, we delexicalize all slots except *isforbusinesscomputing* and *request*. In Hotel dataset, we delexicalize all slots except *acceptscreditcards*, *dogsallowed* and *hasinternet*. In Restaurant dataset, we delexicalize all slots except *kidsallowed* and *request*.

	E2E-NLG	TV	Laptop	Hotel	Restaurant
Dropout rate	0.4	0.2	0.2	0.3	0.3
Learning rate	0.005	0.001	0.001	0.005	0.001
Batch size	20	20	20	20	20
Dictionary dimension	100	50	50	200	50
RNN hidden size	512	128	512	256	512

Table 3: Hyperparameters of NLG-LM in experiments.

Baseline. Our baseline models include TGen (Dušek and Jurčíček, 2016), SC-LSTM (Wen et al., 2015b), RALSTM (Tran and Nguyen, 2017) and Slug (Juraska et al., 2018).

Training details. We use Adamax (Kingma and Ba, 2014) as the optimizer. We use teacher forcing, which means that during training, the decoder is always presented with the previous ground-truth token. The language modeling task is only used during training, and it uses utterances from the same batch as NLG task. The inference uses a beam search of width 10. We use the multi-task coefficient $\alpha = 0.5$ in all experiments. The hyperparameters were chosen on the dev set with early stopping, as shown in Table 3.

4.2 Result

We present our experimental results in Table 1 and 2. As shown, our model, NLG-LM, outperforms the baseline models in all 5 datasets. In E2E-NLG dataset, it achieves 2.2% higher BLEU score and 0.013 higher NIST score than Slug. In TV, Laptop, Hotel and Restaurant datasets, NLG-LM greatly improves previously best result by 7.6%, 6.1%, 4.1%, and 1.6%, respectively. We also ran our model without the language modeling task as an ablation study, denoted by *w/o LM*. As seen, language modeling can improve the result by 0.8% to 6.1%, or on average 2.4%, which demonstrates the effectiveness of multi-task learning.

In the appendix, we examined some predicted samples generated from our model, which shows that the addition of the language model makes the generated responses more natural and variable.

Efficiency. We compared the training time of NLG-LM with that of w/o LM. As shown in Table 4, the additional language model only introduces 23.6% more training time, since it does not involve expensive attention computation. Therefore, NLG-LM can offer more natural response generation with comparable efficiency.

	NLG-LM	w/o LM
Per-batch	618.87s	500.64s

Table 4: Running time per batch of NLG-LM and w/o LM on E2ENLG training set. The machine has an Intel Xeon CPU E5 and a Tesla v100 GPU.

5 Conclusions

In this paper, we propose a novel multi-task learning method, NLG-LM. It incorporates a language model task into the response generation process as an unconditioned complementary process to boost the naturalness of generated utterances. We fit both tasks into a sequence-to-sequence structure under a multi-task learning scheme. Empirical results show that NLG-LM significantly outperforms previous methods in 5 large-scale datasets with reasonable computational efficiency. Ablation studies show the effectiveness of using the language modeling task within a multi-task scheme.

Acknowledgement

We thank the anonymous reviewers for the insightful comments. We thank William Hinthorn for proof-reading our paper.

References

Adam Cheyer and Didier Guzzoni. 2014. Method and apparatus for building an intelligent automated assistant. US Patent 8,677,377.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *arXiv preprint arXiv:1606.05491*.

Juraj Juraska, Panagiotis Karagiannis, Kevin K Bowden, and Marilyn A Walker. 2018. A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. *arXiv preprint arXiv:1805.06553*.

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

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 704–710. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*.

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

Mark Przybocki, Kay Peterson, Sébastien Bronsart, and Gregory Sanders. 2009. The nist 2008 metrics for machine translation challengeoverview, methodology, metrics, and results. *Machine Translation*, 23(2-3):71–103.

Brian Roark, Murat Saraclar, and Michael Collins. 2004. Corrective language modeling for large vocabulary asr with the perceptron algorithm. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–749. IEEE.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Van-Khanh Tran and Le-Minh Nguyen. 2017. Natural language generation for spoken dialogue system using rnn encoder-decoder networks. *arXiv preprint arXiv:1706.00139*.

Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015a. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. *arXiv preprint arXiv:1508.01755*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. *arXiv preprint arXiv:1603.01232*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015b. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Yichong Xu, Xiaodong Liu, Yelong Shen, Jingjing Liu, and Jianfeng Gao. 2018. Multi-task learning for machine reading comprehension. *arXiv preprint arXiv:1809.06963*.