Natural Language Generation by Hierarchical Decoding with Linguistic Patterns

Shang-Yu Su[†] **Kai-Ling Lo**[⋆] **Yi-Ting Yeh**[⋆] **Yun-Nung Chen**[⋆]
[⋆]Department of Computer Science and Information Engineering
[†]Department of Electrical Engineering
National Taiwan University
{r05921117,b04902010,b03902071}@ntu.edu.tw y.v.chen@ieee.org

Abstract

Natural language generation (NLG) is a critical component in spoken dialogue systems. Classic NLG can be divided into two phases: (1) sentence planning: deciding on the overall sentence structure, (2) surface realization: determining specific word forms and flattening the sentence structure into a string. Many simple NLG models are based on recurrent neural networks (RNN) and sequence-to-sequence (seq2seq) model, which basically contains a encoder-decoder structure; these NLG models generate sentences from scratch by jointly optimizing sentence planning and surface realization using a simple cross entropy loss training criterion. However, the simple encoder-decoder architecture usually suffers from generating complex and long sentences, because the decoder has to learn all grammar and diction knowledge. This paper introduces a hierarchical decoding NLG model based on linguistic patterns in different levels, and shows that the proposed method outperforms the traditional one with a smaller model size. Furthermore, the design of the hierarchical decoding is flexible and easily-extensible in various NLG systems[1].

1 Introduction

Spoken dialogue systems that can help users to solve complex tasks have become an emerging research topic in artificial intelligence and natural language processing areas (Wen et al., 2017; Bordes et al., 2017; Dhingra et al., 2017; Li et al., 2017). A typical dialogue system pipeline contains a speech recognizer, a natural language understanding component, a dialogue manager, and a natural language generator (NLG).

NLG is a critical component in a dialogue system, where its goal is to generate the natural language given the semantics provided by the dialogue manager. As the endpoint of interacting with users, the quality of generated sentences is crucial for user experience. The common and mostly adopted method is the rule-based (or template-based) method (Mirkovic and Cavedon, 2011), which can ensure the natural language quality and fluency. Considering that designing templates is time-consuming and the scalability issue, data-driven approaches have been investigated for open-domain NLG tasks.

Recent advances in recurrent neural network-based language model (RNNLM) (Mikolov et al., 2010, 2011) have demonstrated the capability of modeling long-term dependency by leveraging RNN structure. Previous work proposed an RNNLM-based NLG (Wen et al., 2015) that can be trained on any corpus of dialogue act-utterance pairs without any semantic alignment and hand-crafted features. Sequence-to-sequence (seq2seq) generators (Cho et al., 2014; Sutskever et al., 2014) further offer better results by leveraging encoder-decoder structure: previous model encoded syntax trees and dialogue acts into sequences (Dušek and Jurčíček, 2016) as inputs of attentional seq2seq model (Bahdanau et al., 2015). However, it is challenging to generate long and complex sentences by the simple encoder-decoder structure due to grammar complexity and lack of diction knowledge.

This paper proposes a hierarchical decoder leveraging linguistic patterns, where the decoding hierarchy is constructed in terms of part-of-speech (POS) tags. The original single decoding process is separated into a multi-level decoding hierarchy, where each decoding layer generates words associated with a specific POS set. The experiments show that our proposed method outperforms the

The first two authors have equal contributions.

[1]The source code is available at https://github.com/MiuLab/HNLG.

Proceedings of NAACL-HLT 2018, pages 61–66
New Orleans, Louisiana, June 1 - 6, 2018.

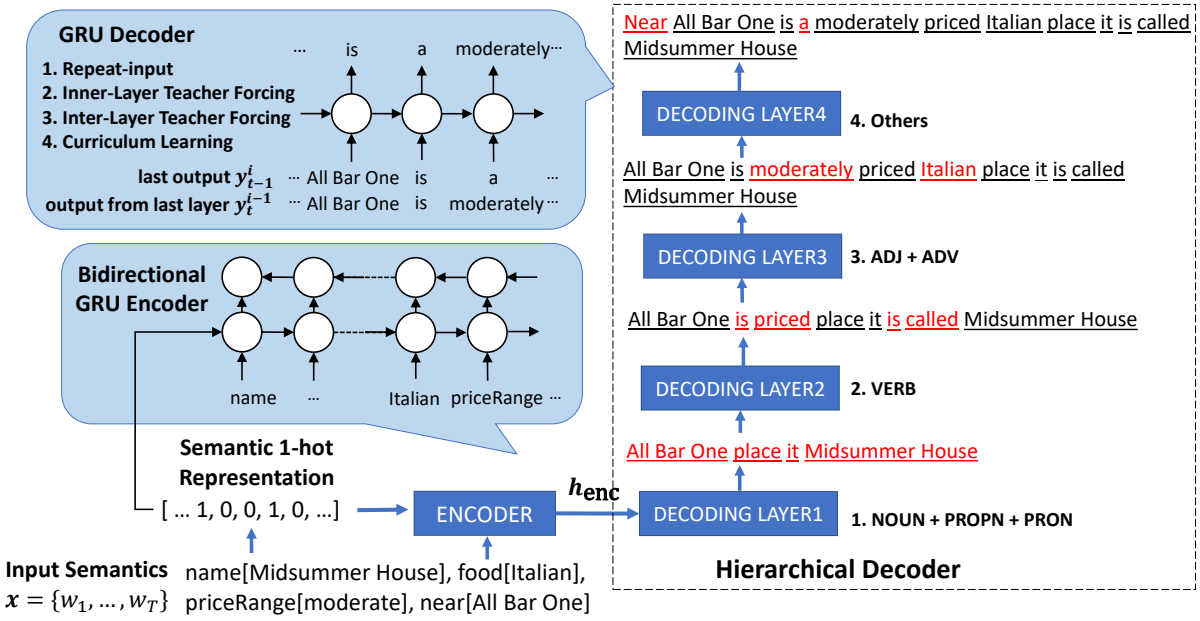

Figure 1: The framework of the proposed semantically conditioned NLG model.

classic seq2seq model with less parameters. In addition, our proposed model allows other word-level or sentence-level characteristics to be further leveraged for better generalization.

2 The Proposed Approach

The framework of the proposed semantically conditioned NLG model is illustrated in Figure 1, where the model architecture is based on an encoder-decoder (seq2seq) design (Cho et al., 2014; Sutskever et al., 2014). In the seq2seq architecture, a typical generation process includes encoding and decoding phases: First, the given semantic representation sequence $\mathbf{x} = \{w_t\}_1^T$ is fed into a RNN-based encoder to capture the temporal dependency and project the input to a latent feature space, and encoded into 1-hot semantic representation as the initial state of the encoder in order to maintain the temporal-independent condition as shown in the left-bottom of Figure 1. The recurrent unit of the encoder is bidirectional gated recurrent unit (GRU) (Cho et al., 2014),

$$\mathbf{h}_{\text{enc}} = \text{BiGRU}(\mathbf{x}). \quad (1)$$

Then the encoded semantic vector, $\mathbf{h}_{\text{enc}}$, flows into an RNN-based decoder as the initial state to generate word sequences by an RNN model shown in the left-top component of the figure.

2.1 Hierarchical Decoder

Despite the intuitive and elegant design of the seq2seq model, it is difficult to generate long, complex, and decent sequences by such encoder-decoder structure, because a single decoder is not capable of learning all diction, grammar, and other related linguistic knowledge. Some prior work applied additional technique such as reranker to select a better result among multiple generated sequences (Wen et al., 2015; Dušek and Jurčíček, 2016). However, the issue still remains unsolved in NLG community.

Therefore, we propose a hierarchical decoder to address the above issue, where the core idea is to separate the decoding process and learn different types of patterns instead of learning all relevant knowledge together. The hierarchical decoder is composed of several decoding layers, each of which is only responsible for learning a portion of the related knowledge. Namely, the linguistic knowledge can be incorporated into the decoding process and divided into several subsets.

In this paper, we use part-of-speech (POS) tags as the additional linguistic features to construct the hierarchy, where POS tags of the words in the target sentence are separated into several subsets and each layer is responsible for decoding the words associated with a specific set of POS patterns. An example is shown in the right part of Figure 1, where the first layer at the bottom is in charge of learning to decode nouns, pronouns, and

proper nouns, and the second layer is in charge of verbs, and so on. Our approach is also intuitive from the viewpoint of how humans learn to speak; for example, infants first learn to say the keywords which are often nouns. When an infant says "*Daddy, toilet.*", it actually means "*Daddy, I want to go to the toilet.*". Along with the growth of the age, children learn more grammars and vocabulary and then start adding verbs to the sentences, further adding adverbs, and so on. This process of how humans learn to speak is the core motivation of our proposed method.

In the hierarchical decoder, the initial state of each GRU-based decoding layer i is the extracted feature $\mathbf{h}_{\text{enc}}$ from the encoder, and the input at every step is the last predicted token $\mathbf{y}^i_{t-1}$ concatenated with the output from the previous layer $\mathbf{y}^{i-1}_t$,

$$\mathbf{h}^i_t, \mathbf{o}^i_t = \text{GRU}^i_{\text{dec}}(\mathbf{y}^i_{t-1}, \mathbf{y}^{i-1}_t \mid \mathbf{h}_{\text{enc}}, \mathbf{h}^i_{t-1}), \quad (2)$$

$$\mathbf{y}^i_t = \texttt{argmax}(\mathbf{o}_t), \quad (3)$$

where $\mathbf{h}^i_t$ is the t-th hidden state of the i-th GRU decoding layer and $\mathbf{y}^i_t$ is the t-th outputted word in the i-th layer. The cross entropy loss is used for optimization.

2.2 Inner- and Inter-Layer Teacher Forcing

Teacher forcing (Williams and Zipser, 1989) is a strategy for training RNN that uses model output from a prior time step as an input, and it works by using the expected output at the current time step $\hat{\mathbf{y}}_t$ as the input at the next time step, rather than the output generated by the network. In our proposed framework, an input of a decoder contains not only the output from the last step but one from the last decoding layer. Therefore, we design two types of teacher forcing techniques – inner-layer and inter-layer.

Inner-layer teacher forcing is the classic teacher forcing strategy:

$$\mathbf{h}^i_t, \mathbf{o}^i_t = \text{GRU}^i_{\text{dec}}(\hat{\mathbf{y}}^i_{t-1}, \mathbf{y}^{i-1}_t \mid \mathbf{h}_{\text{enc}}, \mathbf{h}^i_{t-1}). \quad (4)$$

Inter-layer teacher forcing uses the labels instead of the actual output tokens of the last layer:

$$\mathbf{h}^i_t, \mathbf{o}^i_t = \text{GRU}^i_{\text{dec}}(\mathbf{y}^i_{t-1}, \hat{\mathbf{y}}^{i-1}_t \mid \mathbf{h}_{\text{enc}}, \mathbf{h}^i_{t-1}). \quad (5)$$

The teacher forcing techniques can also be triggered only with a certain probability, which is known as the schedule sampling approach (Bengio et al., 2015). In our experiments, the schedule sampling approach is also adopted.

2.3 Repeat-Input Mechanism

The concept of our proposed method is to hierarchically generate the sequence, gradually adding words associated with different linguistic patterns. Therefore, the generated sequences from the decoders become longer as the generating process proceeds to the higher decoding layers, and the sequence generated by a upper layer should contain the words predicted by the lower layers. In order to ensure the output sequences with the constraints, we design a strategy that repeats the outputs from the last layer as inputs until the current decoding layer outputs the same token, so-called repeat-input mechanism. This approach offers at least two merits: (1) Repeating inputs tells the decoder that the repeated tokens are important to encourage the decoder to generate them. (2) If the expected output sequence of a layer is much shorter than the one of the next layer, the large difference in length becomes a critical issue of the hierarchical decoder, because the output sequence of a layer will be fed into the next layer. With the repeat-input mechanism, the impact of length difference can be mitigated.

2.4 Curriculum Learning

The proposed hierarchical decoder consists of several decoding layers, the expected output sequences of upper layers are longer than the ones in the lower layers. The framework is suitable for applying the curriculum learning (Elman, 1993), in which core concept is that a curriculum of progressively harder tasks could significantly accelerate a networks training. The training procedure is to train each decoding layer for some epochs from the bottommost layer to the topmost one.

3 Experiments

3.1 Setup

The experiments are conducted using the E2E NLG challenge dataset (Novikova et al., 2017)[2], which is a crowd-sourced dataset of 50k instances in the restaurant domain. The input is the semantic frame containing specific slots and corresponding values, and the output is the natural language containing the given semantics as shown in Figure 1.

To prepare the labels of each layer within the hierarchical structure of the proposed method,

[2] http://www.macs.hw.ac.uk/InteractionLab/E2E/

	NLG Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
(a)	Sequence-to-Sequence Model	**44.7**	51.6	19.5	40.6
(b)	+ Hierarchical Decoder	41.1	60.2	31.4	46.2
(c)	+ Hierarchical Decoder, Repeat-Input	41.2	60.5	33.8	48.6
(d)	+ Hierarchical Decoder, Curriculum Learning	40.9	62.9	34.5	50.1
(e)	+ All	44.1	**67.3**	**38.0**	**53.8**
(f)	(e) with High Inner-Layer TF Prob.	36.9	58.5	31.3	45.9
(g)	(e) with High Inter-Layer TF Prob.	42.5	**67.3**	**38.7**	53.3
(h)	(e) with High Inner- and Inter-Layer TF Prob.	41.7	64.5	36.6	52.0

Table 1: The NLG performance reported on BLEU, ROUGE-1, ROUGE-2, and ROUGE-L of models (%).

we utilize spaCy toolkit to perform POS tagging for the target word sequences. Some properties such as names of restaurants are delexicalized (for example, replaced with symbols "RESTAURANT_NAME") to avoid data sparsity. We assign the words with specific POS tags for each decoding layer: **nouns**, **proper nouns**, and **pronouns** for the first layer, **verbs** for the second layer, **adjectives** and **adverbs** for the third layer, and **others** for the forth layer. Note that the hierarchies with more than four levels are also applicable, the proposed hierarchical decoder is a general and easily-extensible concept.

The experimental results are shown in Table 1. Row (a) is the simple seq2seq model as the baseline. The probability of activating inter-layer and inner-layer teacher forcing is set to 0.5 in the rows (a)-(e); to evaluate the impact of teacher forcing, the probability is set to 0.9 (rows (f)-(h)). The probability of teacher forcing is attenuated every epoch, the decay ratio is 0.9. We perform 20 training epochs without early stop; when the curriculum learning approach is applied, only the first layer is trained during first five epochs, the second decoder layer starts to be trained at the sixth epoch, and so on. To evaluate the quality of the generated sequences regarding both precision and recall, the evaluation metrics include BLEU and ROUGE (1, 2, L) scores.

3.2 Results and Analysis

To fairly examine the effectiveness of our proposed approaches, we control the size of the proposed model to be smaller. The baseline seq2seq decoder has 400-dim hidden layer, and the models with the proposed hierarchical decoder (rows (b)-(h)) have four 100-dim decoding layers. Table 1 shows that simply introducing the hierarchical decoding technique without increment of parameters (row (b)) to separate the generation process into several phases achieves significant improvement in ROUGE scores, 16.7% in ROUGE-1, 61% in ROUGE-2, and 13.8% in ROUGE-L respectively. Applying the proposed repeat-input mechanism (row (c)) and the curriculum learning strategy (row (d)) both offer considerable improvement. Combining all the proposed techniques (row (e)) yields the best performance in ROUGE scores with nearly the same performance in BLEU and achieves 30.4%, 94.8%, and 32.5% improvement in ROUGE-1, ROUGE-2, and ROUGE-L respectively, demonstrating the effectiveness of the proposed approach.

To further verify the impact of teacher forcing, the integrated models (row (e)) with high inter and inner-layer teacher forcing probability (rows (f)-(h)) are also evaluated. Note that when the teacher forcing is activated probabilistically, the strategies are also known as schedule sampling (Bengio et al., 2015). Row (f) shows that high probability of triggering inner-layer teacher forcing results in severe performance degradation, while models with high inter-layer teacher forcing probability (rows (g)-(h)) can avoid the harmful impact. The results are reasonable and reflects the potential issue of error propagation within the proposed hierarchical structure.

Note that the decoding process is a single-path forward generation without any heuristics and other mechanisms (like beam search and reranking), so the effectiveness of the proposed methods can be fairly verified. The experiments show that by considering linguistic patterns in hierarchical decoding, the proposed approaches can significantly improve NLG results with smaller models.

4 Conclusion

This paper proposes a seq2seq-based model with a hierarchical decoder that leverages various linguistic patterns and further designs several corre-

sponding training and inference techniques. The experimental results show that the models applying the proposed methods achieve significant improvement over the classic seq2seq model. By introducing additional word-level or sentence-level labels as features, the hierarchy of the decoder can be designed arbitrarily. Namely, the proposed hierarchical decoding concept is general and easily-extensible, with flexibility of being applied to many NLG systems.

Acknowledgements

We would like to thank reviewers for their insightful comments on the paper. The authors are supported by the Institute for Information Industry, Ministry of Science and Technology of Taiwan, Google Research, Microsoft Research, and MediaTek Inc..

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*. pages 1171–1179.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *Proceedings of ICLR*.

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. In *Proceedings of EMNLP*. pages 1724–1734.

Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2017. Towards end-to-end reinforcement learning of dialogue agents for information access. In *Proceedings of ACL*. pages 484–495.

Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of ACL*. pages 45–51.

Jeffrey L Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition* 48(1):71–99.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-end task-completion neural dialogue systems. In *Proceedings of IJCNLP*. pages 733–743.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of Interspeech*.

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Proceedings of ICASSP*. IEEE, pages 5528–5531.

Danilo Mirkovic and Lawrence Cavedon. 2011. Dialogue management using scripts. US Patent 8,041,570.

Jekaterina Novikova, Ondrej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of SIGDIAL*. pages 201–206.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*. pages 3104–3112.

Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. In *Proceedings of SIGDIAL*. pages 275–284.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of EACL*. pages 438–449.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation* 1(2):270–280.

A Dataset Detail

The experiments are conducted using the E2E NLG challenge dataset, which is a crowd-sourced dataset in the restaurant domain, the training set contains 42064 instances while there are 4673 instances in the validation (development) set. In our experiments, we use the validation set to test our models. In the E2E NLG Challenge dataset, the input is the semantics containing slots and their values, and the output is the corresponding natural language. For example, the slot-value pairs "`name[Bibimbap House], food[English], priceRange[moderate], area[riverside], near[Clare Hall]`" correspond to the target sentence

"*Bibimbap House is a moderately priced restaurant who's main cuisine is English food. You will find this local gem near Clare Hall in the Riverside area.*".

B Parameter Setting

We use mini-batch Adam as the optimizer with the batch size of 32 examples. The baseline seq2seq model (row (a)) sets the encoder's hidden layer size to 200 and the decoder's to 400. The size of the hidden layer in the encoder and the decoder layers of the models based on the proposed hierarchical decoder (rows (b)-(h)) are 200 and 100, respectively. Note that in this setting, the models applied the proposed methods will have less parameters than the baseline seq2seq model. In terms of the models utilized the basic RNN cell, the baseline seq2seq model (row (a)) has 640k parameters whereas the proposed models (rows (b)-(h)) have only 520k parameters.