LSTM ENCODER–DECODER FOR DIALOGUE RESPONSE GENERATION

Zhenlong Yu, Caixia Yuan, Xiaojie Wang and Guohua Yang

School of Computer Science, Beijing University of Posts and Telecommunications,
Beijing 100876, China

ABSTRACT

This paper presents a dialogue response generator based on long short term memory (LSTM) neural networks for the SLG (Spoken Language Generation) pilot task of DSTC5 [1]. We first encode the input containing different number of semantic units as fixed-length semantic vector with a LSTM encoder. Then we decode the semantic vector with a variant of LSTM and generate corresponding text. In order to produce more flexible and context-aware response, we incorporate the historical dialogue acts when generating current utterance. Our experiments on DSTC5 data validate that the proposed LSTM-based generator significantly improves the quality of the generated responses compared to the baseline. Furthermore, it also yields comparable results to a state-of-the-art generator when evaluated on the same dataset.

Index Terms— NLG, LSTM, encoder, decoder

1. INTRODUCTION

The natural language generation (NLG) in spoken dialogue systems (SDSs) refers to the task of automatically generating natural language response from dialogue acts produced by a dialogue manager, which plays an important part in SDSs and has a significant impact on a user's experience of the system. The SLG pilot task of DSTC5 [1] aims to generate spoken response for a tour SDS. Its input consists of several speech acts and semantic tags, as illustrated in Table 1. Speech act denotes the general role of the utterance in the current dialog flow, such as question, response and follow. Semantic tag indicates information that we should convey in the current utterance, for example, city name, price and time, which is specified through mentions with verbal values (e.g., "birthday", "August twenty sixth" in Table 1). The output of the SLG pilot task is corresponding spoken expression. Unlike most NLG tasks with a set of flat semantic slots as input, this task takes input with nested semantic structures and the number of semantic symbols varies for different inputs. In order to represent the complicated input semantics, we first exploit a LSTM encoder which maps each input to a fixed-length semantic vector. The semantic vector is the hidden state obtained after the last semantic unit has been processed. In this way, we regard the input containing different number of semantic units as a variable-length sequence and get a semantic vector from the LSTM encoder. Then we decode it with a variant of LSTM which takes the input semantic vector and produces a probability distribution over the tokens in the generated text. The proposed approach is in essence a sequence-to-sequence model [2], which has been successfully used for neural machine translation, parsing and image captioning [3-5]. Inspired by the work of [6], we apply skip connections to the network to reduce the number of processing steps between the bottom network and the top network and therefore mitigate the vanishing gradient problem. Unlike the work of [6-8] which ignores the dialogue histories when generating current utterance, we treat hidden layer vector of the decoder of last turn as historical information of the dialog and deliver it into the encoder of next turn. Our extensive experiments validate the effectiveness of such historical information.

Table 1 An example of SLG input and output

Input	"speech_act": [{
	"attributes": ["EXPLAIN", "WHEN"],
	"act": "FOL"
	}],
	"semantic_tags": [{
	"attributes": {"cat": "EVENT"},
	"main": "det",
	"mention": "birthday"},
	"semantic_tags": [{
	"attributes": {"rel": "NONE", "cat":
	"DATE"},
	"main": "time",
	"mention": "August twenty sixth"
	}]

Output	That's actually his birthday, August twenty sixth.

This paper is structured as follows: In section 2, we briefly review the related work, then in section 3 we describe in detail the network architecture, Section 4 presents

experiments and evaluation of the proposed approach, section 5 concludes our work.

2. RELATED WORK

Data-driven approach for natural language generation mainly follows two streams of research. The one is grammar based NLG which formulates the natural language generation as a grammar derivation process and takes the leaves from the parsing tree as the generated texts. In [9] Konstas and Lapata proposed a PCFG-based generator. Their CFG (context free grammar) combines content selection ("what to say") and surface realization ("how to say"). Although such grammar-based approach can control the expression of semantics well, and ensure that the generated sentences are always grammatical, it takes a lot of time and efforts to define deliberate rules and it is likely to limit their scalability to new domains. The second stream of research has focused on sequence modeling approach for NLG, with special attention to especially LSTM [10] based approach, which can capture information from context. Mei et al. [11] first encodes semantic via an LSTM-based recurrent neural network, then utilizes a novel coarse-to-fine aligner to identify the small subset of semantic to talk about, and finally employs a decoder to generate free-form descriptions. SC-LSTM [6] is a variant of LSTM, which by jointly optimizing sentence planning and surface realization using a simple cross entropy training criterion and strengthen its expression for semantic by adding filter. On the basis of the SC-LSTM, Wen et al. propose that candidates can be reranked with CNN to keep consistent semantic between input and output [8]. Wen et al compare SC-LSTM with the work of [11] and show that SC-LSTM gets better performances on multi-domain NLG task [7]. Our approach is mainly based on the recent work on neural network based NLG. In order to represent the complicated input with nested semantic structures and richer contents, we encode it with LSTM. Furthermore, we focus on models which can exploit dialog histories to generate fluent, more human-like utterances for spoken dialogue systems.

3. MODEL

Our approach works like a language model based on the sequence to sequence framework described in [2, 14], which predicts the next word of the utterance given the previously generated word sequence and the input dialogue acts. The utterance generation process is divided into encoding and decoding phrases.

3.1. Encoding

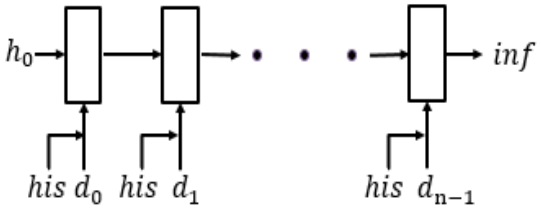

Fig. 1 LSTM encoder

As mentioned above, input contains several nested speech acts and semantic tags. We treat each speech act and semantic tag as one semantic unit of input. Each semantic unit is expressed as one hot vector. Then we get corresponding semantic unit vector by concatenating all one hot vectors in a semantic unit. And we add a flag in semantic unit vector to distinguish speech act and semantic tag. Finally, we pad semantic unit vector to a fixed length d_t. In this way, an input is converted to a set with elements as fixed-length semantic unit vectors $\{d_t | t = 0,1 \ldots n-1\}$, and n refers to the number of semantic units of input (i.e., the total number of speech acts and semantic tags).

Besides the input semantic symbols, our model takes the historical information into consideration when encoding the model input. We treat hidden layer vector h_f of the decoder at last time step as historical vector his implying dialog histories. Therefore, the input of LSTM encoder is the splice of d_t and his, and the output is semantic vector $\inf$ which is the hidden layer vector at last time step. The encoding model is illustrated in Figure 1 and defined by the following formulas.

$$x_t = [his, d_t] \tag{1}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1}) \tag{2}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1}) \tag{3}$$

$$g_t = \Phi(W_g x_t + U_g h_{t-1}) \tag{4}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1}) \tag{5}$$

$$s_t = i_t \bullet g_t + f_t \bullet s_{t-1} \tag{6}$$

$$h_t = o_t \bullet \Phi(s_t) \tag{7}$$

σ refers to sigmoid function, Φ refers to tanh function.

3.2. Decoding

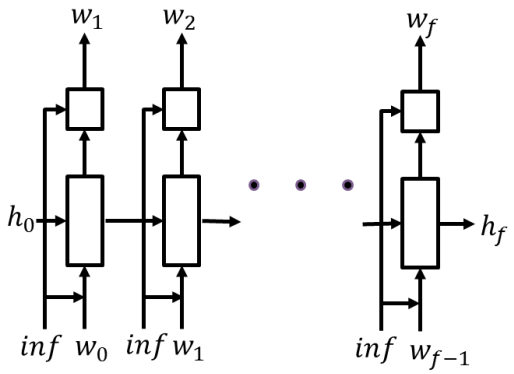

Fig. 2 LSTM decoder

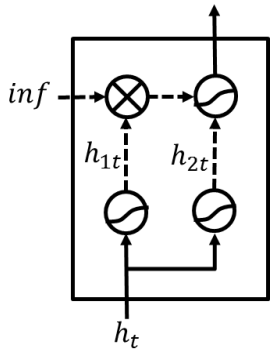

Fig. 3 The inner structure of LSTM decoder

As observed in [6], the standard LSTM may generate inconsistent semantic text in NLG task. To eliminate this problem, we enter semantic vector inf not only at each time step but also before and after LSTM unit which is similar to highway network [12]. In other perspective, semantic vector inf before LSTM unit can make network plan sentence structure at a high level, while semantic vector inf after LSTM unit can make network plan words and other details at a low level. The overall structure of the network is shown in Fig 2. The first layer receives the splice of semantic vector inf and word vector w_t at time step t. The second layer receives the hidden layer vector h_t and the semantic vector inf. We apply network structure that is similar to max out [13] to the LSTM unit of the second layer which is shown in Fig 3. The activation functions operate different linear transformation of h_t, and output h_{1t} and h_{2t}. According to the information of h_{1t}, a part of semantic vector is focused by attention like mechanism and thus determines which part should be expressed. Max activation function gets the filtered vector and h_{2t} and outputs a vector which will be entered softmax layer to predict next words. Decoding process is defined by equations (8) -(17).

$$x_t^{'} = [\mathrm{inf}, w_t] \tag{8}$$

$$i_t^{'} = \sigma(W_i^{'} x_t^{'} + U_i^{'} h_{t-1}^{'}) \tag{9}$$

$$f_t^{'} = \sigma(W_f^{'} x_t^{'} + U_f^{'} h_{t-1}^{'}) \tag{10}$$

$$g_t^{'} = \Phi(W_g^{'} x_t^{'} + U_g^{'} h_{t-1}^{'}) \tag{11}$$

$$o_t^{'} = \sigma(W_o^{'} x_t^{'} + U_o^{'} h_{t-1}^{'}) \tag{12}$$

$$s_t^{'} = i_t^{'} \bullet g_t^{'} + f_t^{'} \bullet s_{t-1}^{'} \tag{13}$$

$$h_t^{'} = o_t^{'} \bullet \Phi(s_t^{'}) \tag{14}$$

$$h_{1t} = \Phi(W_{h1} h_t^{'}) \tag{15}$$

$$h_{2t} = \Phi(W_{h2} h_t^{'}) \tag{16}$$

$$P(w_{t+1} \mid w_t, w_{t-1}, \ldots, w_0, \mathrm{inf}) = soft\max(W_{soft} \max(h_{1t} \bullet \mathrm{inf}, h_{2t})) \tag{17}$$

In a word, our LSTM decoder applies attention and max out [13] to globally control which part of the input should be expressed in a single time step. Such structure is essentially different to that of SC-LSTM, which controls the number of semantic vectors to be expressed in a single time step by means of tuning hyper parameters of additional penalty in cost function [6].

3.3. Cross-language SLG

Unlike previously reported work [5, 6], the SLG task of DSTC 5 is cross-language. In training set, the semantic input and text output are in English language with their Chinese translations and alignment information. In develop and test set, the input and output are in Chinese. To deal with cross-language data, we first replace mentions in English (e.g., “Singapore”) in the training set with their Chinese translations (“新加坡”) through looking up the offered translation and alignment information. In this way, the training set is transformed into Chinese language without additional translation model.

Since mentions can be expressed by more than one word (e.g., "mention": "August twenty sixth" in Table 1 contains 3 words), it is difficult to denote all semantic inputs with a fixed-length vector. It means that we need a nested LSTM structure if we also encode mentions with LSTM, which makes training difficult. To deal with this problem, we create a mention dictionary which denotes different mention with a unique entry. We encode the entry of a mention instead of its verbal values. At same time, the output utterance is translated into Chinese utterance according to the offered translation in training data and the mention value in

Table 2 Results with different model parameters

system	GUIDE		TOURIST	
	AM-FM	BLEU	AM-FM	BLEU
Baseline	0.2201	0.3382	0.2747	0.4056
Generator(128)	0.3177	0.4374	0.3539	0.4839
Generator(200)	0.3247	**0.4445**	0.3561	**0.4882**
Generator(400)	0.3328	0.4352	0.3618	0.4876
Generator(400a)	**0.3354**	0.4380	**0.3652**	0.4851

Table 3 Result comparison with SC-LSTM

System		GUIDE		TOURIST	
		AM-FM	BLEU	AM-FM	BLEU
Generator(200)	with history	0.3247	**0.4445**	0.3561	**0.4882**
	without history	0.3407	0.4294	0.3681	0.4856
SC-LSTM(200)	with history	0.3240	0.4247	0.3642	0.4822
	without history	**0.3519**	0.4194	**0.3748**	0.4747

Table 4 Examples generated by our model

References	Predications
哦，就现在想来新加坡，哦？ (Oh, Do you want to come to Singapore now?)	新加坡。(Singapore.)
其中一个是麦里芝蓄水池。 (One of them is MacRitchie Reservoir.)	那个麦里芝蓄水池。 (That MacRitchie Reservoir.)
你要你想知道那个巴士路线吗？ (Do you want to know the bus route?)	巴士呀，对不对？ (Bus, right?)
呃这个甘榜格南呵， (Eh, Kampong Glam, ah.)	好的，那么你在甘榜格南。 (OK. You are in Kampong Glam)
狮城？(Singapore?)	狮城吗？(Singapore?)

utterance is also replaced with its entry. We need replace the mention entry with its corresponding mention value at the last step of the generation. As a result, the network only learns and predicts tokens referring to mentions instead of exact words.

The objective function was the cross entropy error between the predicted word distribution p_t and the actual word distribution y_t in the training corpus, which is defined as:

$$F(\theta) = \sum_t P_t^T \log(y_t) \qquad (18)$$

4. EXPERIMENT

4.1. Dataset

The dataset of DSTC 5 consists of GUIDE and TOURIST. For GUIDE utterance generation, the model gets TOURIST's input, TOURIST's output and GUIDE's input and need output GUIDE's utterances. For TOURIST utterance generation, the model generates TOURIST's utterances in accordance with TOURIST's input, GUIDE's output and GUIDE's input. There are 13k utterances on training data, 3k utterances on develop data, 12k utterances which small part of need to be generated on test data. Every speech act

contains 4 values and 21 attributes. Each semantic tag contains 8 main values and uncertain number of attributes.

4.2. Parameter setting

Our model is trained 30 iterations in the training set and the parameters that yield the best performance in develop set are chosen. We tested the input word vector respectively with dimension of 128, 200 and 400. Since the model with 400-dimensional word vector overfits easily, we conducted an experiment in 400-dimensional word vector with 20 iterations, which is marked 400a. The experiment was repeated 10 times and we calculated their averages to get a more rational analysis.

4.3. Evaluation

BLEU [15] score and AM-FM [16] metric are used for evaluation. BLEU is to measure geometric average of n-gram precision (for n = 1, 2, 3, 4) of the system generated utterance with respect to reference utterance. AM-FM is the weighted mean of (1) the cosine similarity between the system generated utterance and the reference utterance and (2) the normalized n-gram probability of the system generated utterance.

The baseline model is provided by DSTC5 [1]. It uses an example-based language generation approach using k-nearest neighbors algorithm on the vector space with the speech act and semantic tags features. For each input, the system finds the most similar instance in the English training set, and then outputs the top-1 hypothesis of its Chinese translations as the generated result.

Table 2 demonstrates results achieved by different model parameters. Overall, the model yields better performance on TOURIST than on GUIDE. An important implication of this observation is that our model favors to learn and predict shorter utterance. Generally, the larger dimension of the word embeddings is, the better the performance becomes. However, Generator(400) has a poorer performance likely due to the over-fitting problem. In order to compare our approach with previous related work, Table 3 summarizes results achieved by our approach (Generator(200)) and the work of [6] (SC-LSTM(200)). To draw a paralleled comparison, SC-LSTM(200) uses a single LSTM layer in decoder and uses the same encoder with Generator(200) due that SC-LSTM is unable to accept variable-length input. We can observe that our model gets better BLEU scores when compared with that of [6]. Although the historical information has not contributed to the results as much as expected, mainly due to its sparseness and incomplete model convergence, we still believe that dialogue history should be a useful hint to generate context-sensitive utterance.

Table 4 shows the results of our model on several examples. It is interesting to notice that our model remedies some odd utterances of the references in both grammatical fluency and semantic correctness.

.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed LSTM-based generator, which encodes semantic input with complicated structure and generates corresponding spoken utterance. When tested on the SLG pilot task of DSTC5, our model outperforms the provided baseline, and obtains results comparable to a state-of-the-art system.

In this challenge, the input with nested semantic structure calls for a complicated encoder, which makes our model too complex to reach complete convergence within reasonable length of time. The above factors lead to the result that our model biases to generate frequent words and learn the commonly used expressions. It is necessary to explore how to avoid the negatives of these factors.

ACKNOWLEDGMENTS

This work was partially supported by Natural Science Foundation of China (No. 61202248, No.61273365), Discipline Building Planning 111 Base Fund (No. B08004).

6. REFERENCES

[1] Kim, Seokhwan and D'Haro, Luis Fernando and Banchs, Rafael E. and Williams, Jason and Henderson, Matthew and Yoshino, Koichiro, The Fifth Dialog State Tracking Challenge. In Proceedings of the 2016 IEEE Workshop on Spoken Language Technology (SLT). 2016.

[2] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Advances in neural information processing systems. 2014: 3104-3112.

[3] Jean, S., Cho, K., Memisevic, R., and Bengio, Y. On using very large target vocabulary for neural machine translation. CoRR, abs/1412.2007, 2014.

[4] Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I.,and Hinton, G. Grammar as a foreign language. arXiv preprint arXiv:1412.7449, 2014a.

[5] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. Show and tell: A neural image caption generator. arXiv preprint arXiv:1411.4555, 2014b.

[6] Wen T H, Gasic M, Mrksic N, et al. Semantically conditioned lstm-based natural language generation for spoken dialogue systems[J]. arXiv preprint arXiv:1508.01745, 2015.

[7] Wen T H, Gašic M, Mrkšic N, et al. Toward Multi-domain Language Generation using Recurrent Neural Networks[J].

[8] Wen T H, Gasic M, Kim D, et al. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking[J]. arXiv preprint arXiv:1508.01755, 2015.

[9] Konstas I, Lapata M. A Global Model for Concept-to-Text Generation[J]. J. Artif. Intell. Res.(JAIR), 2013, 48: 305-346.

[10] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[11] Mei H, Bansal M, Walter M R. What to talk about and how? Selective Generation using LSTMs with Coarse-to-Fine Alignment[J]. arXiv preprint arXiv:1509.00838, 2015.

[12] Srivastava R K, Greff K, Schmidhuber J. Highway networks[J]. arXiv preprint arXiv:1505.00387, 2015.

[13] Goodfellow I J, Warde-Farley D, Mirza M, et al. Maxout networks[J]. ICML (3), 2013, 28: 1319-1327.

[14] Mikolov, T., Karafi át, M., Burget, L., Cernock`y, J., and Khudanpur, S. Recurrent neural network based language model. In *INTERSPEECH*, pp. 1045–1048, 2010.

[15] Papineni K, Roukos S, Ward T, et al. BLEU: a method for automatic evaluation of machine translation[C]//Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002: 311-318.

[16] Banchs R E, Li H. AM-FM: a semantic framework for translation quality assessment[C]//Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2. Association for Computational Linguistics, 2011: 153-158.