Neural Dialogue Model with Retrieval Attention for Personalized Response Generation

Cong Xu[1, 2], Zhenqi Sun[2, 3], Qi Jia[2, 3], Dezheng Zhang[2, 3], Yonghong Xie[2, 3, *] and Alan Yang[4]

Abstract: With the success of new speech-based human-computer interfaces, there is a great need for effective and friendly dialogue agents that can communicate with people naturally and continuously. However, the lack of personality and consistency is one of critical problems in neural dialogue systems. In this paper, we aim to generate consistent response with fixed profile and background information for building a realistic dialogue system. Based on the encoder-decoder model, we propose a retrieval mechanism to deliver natural and fluent response with proper information from a profile database. Moreover, in order to improve the efficiency of training the dataset related to profile information, we adopt a method of pre-training and adjustment for general dataset and profile dataset. Our model is trained by social dialogue data from Weibo. According to both automatic and human evaluation metrics, the proposed model significantly outperforms standard encoder-decoder model and other improved models on providing the correct profile and high-quality responses.

Keywords: Dialogue system, LSTM, encoder-decoder model, attention mechanism.

1 Introduction

With the growing researches about open domain chatbot, consistency has become an important and intractable problem. The popular method of building a chatbot is to feed encoder-decoder model with a bunch of corpus [Shi, Yao, Chen et al. (2015)]. Due to the lack of external knowledge, this model only learns the information from the mass corpus. The current conversational agents are unable to deliver useful and coherent responses included specific character traits. And some commercial chatbots are capable of giving precise and designed response to character information by factitious template. Thus, most chatbots either look like the split personality or respond several fixed answers about their background and profile. We consider background and profile are the most essential parts

[1] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, 100083, China.

[2] Beijing Key Laboratory of Knowledge Engineering for Materials Science, University of Science and Technology Beijing, Beijing, 100083, China.

[3] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China.

[4] Amphenol AssembleTech, Houston, TX 77070, US.

[*] Corresponding Author: Yonghong Xie. Email: xie.yonghong@126.com.

CMC. doi:10.32604/cmc.2020.05239 www.techscience.com/cmc

of character or personality. An ideal chatbot should respond the post sentence about its name, age, hometown, etc. exactly and naturally. In this paper, we adopt an end-to-end generation model with a retrieval attention network for solving above difficulties. As we know, the generation model has the advantage of diversity and universality, and the method of retrieving response template can produce high quality answers about specific content. Therefore, the combination of these two methods can build a better open domain dialogue system featured with variety and consistency.

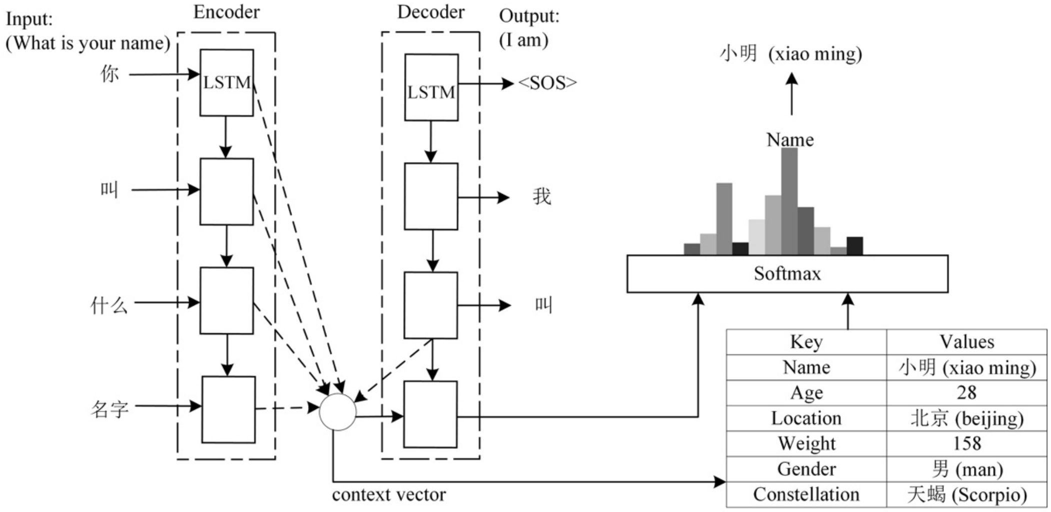
In previous works, researchers dealt with personality of data-driven dialogue system in two ways. One is treating coherent personality as a speaker-specific conversation style. The representative work is that Walker et al. [Walker, Litman, Kamm et al. (1997)] argued that linguistic style is a key aspect of character and showed how identity and linguistic style represent personality during interacting with an agent. Li et al. [Li, Galley, Brock- ett et al. (2016b)] proposed user embeddings for incorporating interaction patterns into an Encoder-Decoder model. These personal vectors lead the model to behave like a specific person during the conversation with real person. Another way to generate personalized response is endowing a chatbot with the background, profile and more clues that can outline the personality of the character. The outstanding work in this way is that Qian et al. [(Qian, Huang, Zhao et al. (2018)] addressed the problem of generating coherent responses to a pre-specified chatbot profile by three components, a profile detector, bidirectional decoder and a position detector. Inspirited by their scenario, we also define personality as profile and identity such as including name, age, gender, weight, hometown and constellation. While we propose an end to end model with retrieval attention network for generating the information-rich responses included the precise profile and identity information. Unlike the previous approaches which learn to encode and decode without extra information for responses generation, the proposed approach fully adopts the advantages of the encoder-decoder approaches and learns to generate personalized responses without too many complicated components.

The main contributions of this paper are two-fold: 1. We incorporate a retrieval attention component into an encoder-decoder framework, which allows the agent to acquire the exact useful information from a key-value profile database. 2. We propose a training approach for improving the performance of our model on generating personalized responses. The first step is to pre-train the standard encoder-decoder model on general dataset and the second step is to adjust the pre-trained model by adding a retrieval attention and find-tune on the profile dataset.

The rest of the paper is organized as follow. Section 2 briefly reviews the LSTM based encoder-decoder model and proposes our model; Section 3 demonstrates the training process in detail; In Section 4, we discuss the experiment setting and results; The final Section 5 shows the conclusion and future direction.

2 Model

Recently, sequence to sequence [Sutskever, Vinyals and Le (2014)] or encoder-decoder model [Cho, Van Merriënboer, Gulcehre et al. (2014)] has been the dominant method in neural language processing, especially in neural translation, image caption and open domain dialogue. We add an extra attention network [Bahdanau, Cho and Bengio (2014)] which is capable of retrieving the key-value profile database into conventional encoder-decoder framework, which is inspired by recent work on key-value memory networks [Miller, Fisch, Dodge et al. (2016)] and key-value retrieval networks [Eric, Krishnan, Charette et al. (2017)]. The proposed model is an end-to-end trainable system based on the recurrent neural network (RNN) [Zaremba, Sutskever and Vinyals (2014)]. The model given an input sentence X tries to generate an output sentence Y that can maximize the conditional probability. For the purpose of inserting the profile information into the output sentence, we combine the encoder-decoder model with a retrieval attention network as Fig. 1. Each component of our model is described in detail below.



Key	Values
Name	小明 (xiao ming)
Age	28
Location	北京 (beijing)
Weight	158
Gender	男 (man)
Constellation	天蝎 (Scorpio)

Figure 1: The framework of the proposed model

2.1 Encoder-decoder

Encoder and decoder parts both consist of the RNN units which typically are replaced by the LSTM [Hochreiter and Schmidhuber (1997)] or GRU [Chung, Gulcehre, Cho et al. (2014)]. The encoder layer reads the source sentence and transforms it into a context vector which can capture the semantic information of the whole sentence. First of all, the words of 1-hot representation are converted to embedding vectors x_t by the word embedding layer, where x_t is a d-dimensional word vector. Then we can define a sentence with length of T as $X = \{x_1; x_2; \cdots; x_T\}$.

We chose LSTM cells as recurrent units instead of vanilla RNN cells, which can avoid problems of vanishing and exploding gradients in long-term dependencies. Since the Bi-directional LSTM (Bi-LSTM) [Hakkani Tür, Tür, Celikyilmaz et al. (2016)] has the ability of capturing the sequential information of input sentence forwardly and backwardly, we adopt Bi-LSTM for encoding the source sentence from both directions. The forward LSTM encodes the source sentence into hidden state $\overrightarrow{h}_t$ for each time step t by:

$$i_t = \sigma\left(W_i \cdot \left[\overrightarrow{h}_{t-1}, e(x_t)\right]\right) \tag{1}$$

$$f_t = \sigma\left(W_f \cdot \left[\overrightarrow{h}_{t-1}, e(x_t)\right]\right) \tag{2}$$

$$o_t = \sigma\left(W_o \cdot \left[\overrightarrow{h}_{t-1}, e(x_t)\right]\right) \tag{3}$$

$$\tilde{C}_t = \tanh\left(W_l \cdot \left[\overrightarrow{h}_{t-1}, e(x_t)\right]\right) \tag{4}$$

where i_t, f_t, o_t denote the input gate, forget gate and output gate respectively, W_i, W_f, W_o are the weights of corresponding gate, and $sigma$ denotes the sigmoid function. The variable $\tilde{C}_t$ is used to compute hidden state $\overrightarrow{h}_t$.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_{t-1} \tag{5}$$

$$\overrightarrow{h}_t = o_t \cdot \tanh\,(C_t) \tag{6}$$

Then the backward hidden state $\overleftarrow{h}_t$ is calculated in the similar way. We concatenate the forward and backward hidden states as encoder hidden state h_i.

$$h_i = \left[\overrightarrow{h}_i, \overleftarrow{h}_i\right] \tag{7}$$

The conventional encoder-decoder model considers the final encoder hidden state h_T as the context vector c_t. However, Bahdanau et al. [Bahdanau, Cho and Bengio (2014)] proposed attention mechanism, which allows a model to automatically search for parts of a source sentence that are relevant to predict a target word. In this paper, we utilize Bahdanau attention to dynamically compute the vector c_t for every hidden state as follow:

$$c_t = \textstyle\sum_{j=1}^{N} a_{i,j}\, h_j \tag{8}$$

and

$$a_{i,j} = \frac{exp\ (score(s_{i-1}, h_i))}{\sum_{k=1}^{N} exp\ (score(s_{i-1}, h_k))} \tag{9}$$

$$score(s_i, h_i) = v^T \tanh\,(W_\alpha[s_i; h_i]) \tag{10}$$

where s_i denotes the decoder hidden state, v and W_α are trainable parameters. Then the decoder layer receives the context vector c_i and last decoder hidden state s_{i-1} for computing the distribution over the all vocabulary.

$$o_t = LSTM(s_{t-1}, c_t, y_{t-1}) \tag{11}$$

$$y_t = \mathrm{Softmax}(o_t) \tag{12}$$

We chose the word which has the maximum probability as the predicted word y_t^* and the output sequence consists of predicted word of each step. The goal of training is to maximize the log-likelihood of the correct output sequence given the input sequence.

2.2 Retrieval attention on key-value profile

We treat the personality as some information about profile and background as shown in Tab. 1. Key-value paired database is designed manually and is retrieved by an attention network. We store every category of profile as key and store the specific profile as value, like $< name, Tom >$.

Table 1: Key-value profile database

Key	name	age	location	weight	gender	constellation
Value	小明(xiaoming)	28	北京(beijing)	150	男(man)	天蝎(Scorpio)

The attention network computes the relevance score between input sentence and keys. In order to combine the relevance score and the logit score of all words together, we extend the relevance score vector l_t with m dimensions into o_t with $|V|+m$ dimensions by filling $|V|$ number of zero.

$$score(s_i, d_k) = u^T \tanh\left(W_\beta[s_i, d_k]\right) \tag{13}$$

$$l_t = [score(s_i, d_1), \cdots, score(s_i, d_k), \cdots], k \in [1,2,\cdots,m] \tag{14}$$

where m is the number of keys of the profile database.

$$o'_t = o_t + o'' \tag{15}$$

We can select a specific value in profile database or a word in the vocabulary to output in the decoder. For our purposes, the various specific profiles in target sentence of training dataset are replaced with canonicalized representation, such as name, age, location and so on. At testing time, if the decoder outputs the canonicalized representation, we convert it into the actual corresponding value through the database lookup. As the result of combining the vector of relevance scores with the vector of logit scores, the output of decoder is expanded to $|V|+m$ dimensions. We utilize softmax function to normalize the combination scores. Therefore, the output y_t of the proposed model is the probability distribution of the whole vocabulary and the keys of profile database.

$$y_t = \text{Softmax}(o'_t) \tag{16}$$

To address the problem of generating general and useless responses, we adopt the reranking method proposed by Li et al. [Li, Galley, Brockett et al. (2016a)]. During decoding time, the model generate the output sentence by using beam search [Jia, Gavves, Fernando et al. (2015)]. Then we rerank all candidate sentences by calculating a scoring function which is defined as follow.

$$S = \log p(Y|X) + \varphi \log p(X|Y) + \lambda|L| \tag{17}$$

where $\log p(Y|X)$ represents the probability of the generated response Y given the input sentence X. $|L|$ denotes the length of the generated sentence. $\log p(X|Y)$ is the probability of outputting X given input sentence Y through another standard encoder-decoder model. We feed $[X, Y]$ pairs to train this inverse encoder-decoder model. We follow Li et al. [Li, Galley, Brockett et al. (2016a)] by using the development set of MERT [Och (2003)] to optimize φ and λ.

3 Training process

The loss function is the log cross entropy between the output sequence of the above model and the target sequence. We adjust the parameters of all models by using Adam [Kingma and Ba (2015)]. To improve the model performance on generating response with profile information, we propose a two-step training approach, namely pre-train and adjustment. Firstly, we feed the standard encoder-decoder model with the dialogue pairs from which profile information has been removed. The purpose of this step is to train a generic version of the dialog model. Secondly, the pre-trained model that has added a retrieval attention receives the dialogue pairs about profile information to learn to generate the personalized responses.

4 Experiments and analysis

4.1 Dataset

To test the performance of the proposed model, we conduct an experiment with Weibo dataset [Qian, Huang, Zhao et al. (2018)]. This dataset concludes 9,697,651 post-response pairs from Weibo and 76,930 pairs labeled manually for profile keys (name, gender, age, city, weight, constellation); The testing dataset has 3000 pairs labeled the corresponding profile key. We replace the phrases related to profile with tags, such as <Name>, <Age> and so on. Unseen tokens in the training/testing set are categorized as <UNK> and <EOS> is appended to the end of each input sentence.

4.2 Setting

We compare our model (E2D-RA) with standard Encoder-Decoder model (E2D) with greedy search, Encoder-Decoder model with beam search (E2D-BS) and Encoder-Decoder model with a profile classifier (E2D-PC); The decoder of E2D-PC model receives a classification sign from the profile classifier which is trained separately by profile dataset.

Our model consists of a 256 units Bi-LSTM encoder and a 256 units LSTM decoder. Word embeddings are set as 128 dimensions and randomly initialized. We apply dropout [Zarem- ba, Sutskever and Vinyals (2014)] on decoder layer to regularize the network and set the dropout rate as 0.1. When training our model, we adopt mini-batch method with batch size of 128 for pre-training and adjustment steps. The optimization algorithm is Adam and the threshold of gradient clipping is set to 10. Our code is implemented in Pytorch and run on a single GPU device GTX 1080Ti. We adopt accuracy, Blue score Papineni et al. [Papineni, Roukos, Ward et al. (2002)] and human evaluation to evaluate all results.

4.3 Experiment results

The results listed in Tab. 2 are the accuracy of correct classification on all test data and each category of profile data. We see the standard encoder-decoder model has the lowest accuracy score for most categories of profile dataset. And the beam search slightly contributes to improve the result of classification. Moreover, a profile classifier is beneficial to generate correct personalized responses and the E2D-PC model has the highest score on the class of location. Our model with retrieval attention clearly outperforms on all test data and three categories of profile data. Finally, we combine above two methods to decode responses, which lead to obtain the best accuracy for classifying name and gender categories. Moreover, we can see that the accuracy of the two steps training method is often much higher than those of the one step training method in Tab. 2. The experiment shows the effectiveness and accuracy of the proposed model for outputting responses with a proper information.

Table 2: Accuracy (%) of dialogue models for full testing dataset and each categories of profile dataset

Model	Full	Name	Age	Location	Weight	Gender	Constellation
E2D	17.9	12.7	25.3	17.1	10.2	27.3	9.4
E2D-BS	20.3	18.6	23.7	15.3	19.3	24.1	11.8
E2D-PC	28.6	39.4	34.2	33.7	27.6	32.7	23.3
E2D-RA (one step)	27.5	40.2	27.9	29.9	30.3	28.3	26.8
E2D-RA (two steps)	38.1	50.3	54.9	27.5	34.9	40.2	32.5
E2D-RA-PC	37	52.7	45.3	31	31.3	41.3	30.5

We report the performance using standard BLEU metric in Tab. 3. As we can see, our model also obtains the best BLEU score, which means the E2D-RA model has learnt better responses from the training dialogue pairs.

Table 3: BLUE score on Weibo dataset

Model	E2D	E2D-BS	E2D-PC	E2D-RA	E2D-RA-PC
BLEU	6.7	9.3	9.7	12.1	11.3

As far as we know, there is no a totally appropriate metric to evaluate the dialogue system. We conducted an experiment of human evaluation for comparing the realistic performance of above models. For this purpose, we randomly generated 100 distinct input sentences across the six categories of profile data. We presented the source sentences and the corresponding responses of every model on the screen for 4 volunteers. Every volunteer should sort the responses from the most relevant one to the less relevant one. We adopt 5-point mark scoring to assign a score to each model. Then we count the score of each model and report the statistical results in Tab. 4.

Table 4: Statistical results of score of each model by human evaluation

Model	Five	Four	Three	Two	One
E2D	3	5	30	142	220
E2D-BS	33	27	128	161	51
E2D-PC	69	47	193	57	34
E2D-RA	201	166	19	11	3
E2D-RA-PC	94	155	30	29	92

We found the proposed model has a high probability of getting five or four points by human evaluating. Tab. 5 illustrates some examples of responses delivered by the five models for the same input.

Table 5: Samples of responses on human evaluation data

Model	Post: 你的房子在哪里啊？(Where is your house?)
E2D	我不知道。(I don’t know.)
E2D-BS	联系房东吧。(Contact the landlord.)
E2D-PC	一个很久的房子。(A very old house.)
E2D-RA	我住在北京的楼房里。(I live in a building in Beijing.)
E2D-RA-PC	可以来北京找我玩。(You can come to Beijing to play with me.)
Models	Post: 你今年多大了？(How old are you?)
E2D	不小了吧。(Not too young.)
E2D-BS	16 了。(16.)
E2D-PC	你觉得呢。(What do you think?)
E2D-RA	我已经 28 岁了。(I am 28.)
E2D-RA-PC	今年 28。(28, this year)

We can clearly see that our model has the ability of generating a natural and fluent response with the accurate information. It is a small step to solve the consistency and personalization of neural dialogue system.

5 Conclusions and future work

In this work, we have presented a novel open domain dialogue model which is able to generate personalized response by retrieving a designed profile database. The proposed method endows the encoder-decoder model with the ability of responding the consistent information through a retrieval attention. In addition, we adopt a two-step training approach to improve the performance of delivering the proper profile. Our model outperforms competitive baseline model and other improved models on both automatic and human evaluation metrics. As for future work, we will investigate how to apply generative adversarial network [Yu, Zhang, Wang et al. (2017)] to improve the conversational systems on providing the most proper personalized and consistent responses.

Acknowledgement: This work is supported by the National Key Research and Development Program of China under Grant 2017YFB1002304. The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the funding agencies. We thank Weiyan Li for helping to revise the paper.

References

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

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F. et al. (2014): Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pp. 1724-1734.

Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. (2014): Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555.

Eric, M.; Krishnan, L.; Charette, F.; Manning, C. D. (2017): Key-value retrieval networks for task-oriented dialogue. *Proceedings of the 18th Annual SIGDial Meeting on Discourse and Dialogue*, pp. 37-49.

Hakkani Tür, D.; Tür, G.; Celikyilmaz, A.; Chen, Y. N.; Gao, J. et al. (2016): Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. *Interspeech*, pp. 715-719.

Hochreiter, S.; Schmidhuber, J. (1997): Long short-term memory. *Neural Computation*, vol. 9, no. 8, pp. 1735-1780.

Jia, X.; Gavves, E.; Fernando, B.; Tuytelaars, T. (2015): Guiding the long-short term memory model for image caption generation. *IEEE International Conference on Computer Vision*, pp. 2407-2415.

Kingma, D. P.; Ba, J. (2015): Adam: a method for stochastic optimization. *Proceedings of the International Conference on Learning Representations*, pp. 126-138.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; Dolan, B. (2016): A diversity-promoting objective function for neural conversation models. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110-119.

Li, J.; Galley, M.; Brockett, C.; Spithourakis, G. P.; Gao, J. et al. (2016): A persona-based neural conversation model. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 994-1003.

Miller, A. H.; Fisch, A.; Dodge, J.; Karimi, A. H.; Bordes, A. et al. (2016): Key-value memory networks for directly reading documents. *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pp. 1013-1025.

Och, F. J. (2003): Minimum error rate training in statistical machine translation. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, vol. 1, pp. 160-167.

Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. (2002): BLEU: a method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 311-318.

Qian, Q.; Huang, M.; Zhao, H.; Xu, J.; Zhu, X. (2018): Assigning personality/identity to a chatting machine for coherent conversation generation. *Proceedings of International Joint Conference on Artificial Intelligence and European Conference on Artificial Intelligence*, pp. 320-341.

Shi, Y.; Yao, K.; Chen, H.; Pan, Y.; Hwang, M. Y. et al. (2015): Contextual spoken language understanding using recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5271-5275.

Sutskever, I.; Vinyals, O.; Le, Q. V. (2014); Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, pp. 3104-3112.

Walker, M. A.; Litman, D. J.; Kamm, C. A.; Abella, A. (1997): PARADISE: a framework for evaluating spoken dialogue agents. *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics*, pp. 271-280.

Yu, L.; Zhang, W.; Wang, J.; Yu, Y. (2017): SeqGAN: sequence generative adversarial nets with policy gradient. *Proceedings of the Association for the Advance of Artificial Intelligence Conference*, pp. 2852-2858.

Zaremba, W.; Sutskever, I.; Vinyals, O. (2014): Recurrent neural network regularization. arXiv:1409.2329.