Topic-Net Conversation Model

Min Peng[1(✉)], Dian Chen[1], Qianqian Xie[1], Yanchun Zhang[2], Hua Wang[2], Gang Hu[1], Wang Gao[1,3], and Yihan Zhang[4]

[1] Computer School, Wuhan University, Wuhan, China
{pengm,dchen,xieq,hoogang,gaowang}@whu.edu.cn
[2] Centre for Applied informatics, University of Victoria, Melbourne, Australia
{yanchun.zhang,hua.wang,}@vu.edu.au
[3] College of Sports Science and Technology, Wuhan Sports University, Wuhan, China
[4] Computer School, National University of Singapore, Singapore, Singapore
e0261914@u.nus.edu

Abstract. Most sequence-to-sequence neural conversation models have a ubiquitous problem that they tend to generate boring and safe responses with almost none useful information, such as "I don't know" or "I'm OK". In this paper, we study the response generation problem and propose a topic-net conversation model (TNCM) via incorporating topic information into the sequence-to-sequence framework. TNCM generates every response word using not only its word embedding hidden state but also the topic embedding, which guides the model to form more interesting and informative responses in conversation. The model increases the possibility of the topic word appearing in the response further via a mixed probabilistic model of two modes, namely the generate-mode and the topic-mode. Moreover, we improve the process of beam search during the test, which enhances the performance with better efficiency. Evaluation results on large scale dataset indicate that our model can significantly outperform state-of-the-art methods on response generation of the conversation system.

Keywords: Conversation model · Response generation
Sequence-to-sequence

1 Introduction

With the development of artificial intelligence technology, natural language processing (NLP) is widely used in many fields, such as data anonymisation [21–23], electronic business [8], access control [7,26] and bioinformatics [27]. Conversation systems, also known as dialogue systems and sometimes chatbots, aim at generating relevant and fluent responses in free-form natural language, which is a challenging task in AI and NLP. Conversation systems can be divided into the goal-driven systems [31] and open-domain chatbots. The former such as technical support services goals to help people to complete a specific task, while the latter focuses on talking like human chit-chat in the open domains [16,25] such

H. Hacid et al. (Eds.): WISE 2018, LNCS 11233, pp. 483–496, 2018.
https://doi.org/10.1007/978-3-030-02922-7_33

as language learning tools or computer game characters. Previous researches on conversation focused on goal-driven systems. Recently, with the large amount of conversation data available on the Internet, open-domain chatbots are drawing more and more attention in both academia and industry [28].

Sequence-to-sequence model (seq2seq) [1,24], as a data-driven method mapping between the two sequences of arbitrary length, has achieved remarkable success in various natural language processing (NLP) tasks [5], including but not limited to conversation systems [25] which is referred to the neural conversational model. Seq2seq is essentially an encoder-decoder model, in which the encoder first transforms the input sequence to a certain representation which can then transformed into the output sequence by the decoder. [5] These methods have become mainstream for capturing semantic and syntactic relationships between messages and responses in a scalable and end-to-end manner. However, in practice, the neural conversational model tends to generate trivial or non-committal responses, often involving in high-frequency general responses such as "I don't know" [18], which is boring and frustrating with almost none useful information.

Furthermore, in seq2seq model, during the test period, seq2seq model can only perceive the information of the partially-formed sequence that has been generated. However, the training process maximizes the generation probability of the fully-formed word sequence conditioned on the input sequence and the history of target words, which misses the information of the partially-formed sequence. The difference between the training and the testing process also involves in high-frequency general responses. In practice, during the test period, for lack of the information of the fully-formed word sequence, the commonly used method is that the response generation is accomplished by searching over output sequence greedily with beam-search. However, in this method, the problem of the high-frequency general responses still exists.

In this paper, we study the response generation problem of open-domain chatbots. Notably our goal is to generate responses which are more interesting, diverse and informative. We observe that the people often subconsciously extract the theme of the input message and then generate a follow-up response with the extracted theme. Inspired by this observation, we extract the topic of the input message to guide the generation of the response, thus increasing the diversity of responses and reducing the probability of high-frequency general responses. Moreover, the topic of the input message can be used as the extracted prior knowledge to increase the controllability.

We propose a topic-net conversation model (TNCM). TNCM is based on the sequence-to-sequence framework. It contains the topic generation model and the topic-net seq2seq model. The topic generation model is a convolutional neural network to obtain the topic words. As for the topic-net seq2seq model, it represents the input message as the hidden vector and extracts the topic embedding in encoder. In decoder, the model generates every response word using not only its word embedding hidden state but also the topic embedding. Furthermore, the model increases the possibility of the topic word appearing in the response

based on a mixed probabilistic model of two modes, namely the generate-mode and the topic-mode, that the former is the traditional probabilistic model and the latter picks words from the embedding of the topic. This mechanism makes the word in the response not only related to the input message but also to the topic information of the message and further increases the possibility that the topic word appears in the response. Moreover, to solve the problem of the difference between the training and the testing process, we improve the process of beam search during the test stage, which enhances the performance with better efficiency. Evaluation results on large scale test data indicate that our model can significantly outperform state-of-the-art methods for response generation of the conversation system.

Our contributions in this paper include: (1) a proposal of topic-net conversation model that naturally incorporates topic information into the encoder-decoder structure; (2) a proposal of a method of beam search optimization which enhance the performance with better efficiency; (3) an empirical verification of the effectiveness of topic-net conversation model for response generation.

The rest of this paper is arranged as follows. We survey the related literature in Sect. 2. In Sect. 3, we provide background on sequence-to-sequence model and attention mechanism. Section 4 gives the detail of our model. We report experimental results in Sects. 5 and 6 concludes the paper.

2 Related Work

The early traditional dialogue system relied on heuristic rules for response generation even there was a statistical component, which was inefficient and could only generate very limited responses [15,30].

[17] deemed response generation as a statistical machine translation (SMT) problem. It inspired attempts to extend neural language models in SMT to response generation [24], which [24] present a data-driven approach to generating responses to Twitter status posts, based on phrase-based SMT. [20] improved [17] with a seq2seq model and represented the utterances in previous turns as a context vector and incorporate the context vector into response generation, which became the mainstream method in this area [19,25], that [19] propose Neural Responding Machine (NRM), a neural network-based response generator for short-text conversation. However, the neural conversational model tends to generate high-frequency general responses. To solve this issue, various methods have been proposed. [10] propose using Maximum Mutual Information (MMI) as the objective function in neural models. [11] present persona-based models for handling the issue of speaker consistency in neural response generation. There were also attempts to model complex conversational structures [18] or to seek better optimization strategies [12,13].

There have been studies that introduce the topic information to the conversation system. [28] combined the topic with the tensor network, which the message vector, the response vector, and the two topic vectors are fed to neural tensors to calculate a matching score, but the message-response matching is limited to

data sets and non-scalable. [29] leveraged the topic information obtained from a pre-trained LDA model in response generation by a joint attention mechanism and a biased generation probability. However, the probabilistic topic is not suitable for seq2seq model and the topic is only introduced between the encoder and the decoder.

3 Background: Sequence-to-Sequence Model and Attention Mechanism

Before introducing our model, let us first briefly review the sequence-to-sequence model and the attention mechanism.

3.1 Sequence-to-Sequence Model

In seq2seq, given an input sequence (message) $X = \{x_1, x_2, \ldots, x_{N_x}\}$ and the output sequence (response) $Y = \{y_1, y_2, \ldots, y_{N_y}\}$, the model can be expressed in a probabilistic view as maximizing the generation probability of observing the Y conditioned on X : $p\left(y_1, y_2, \ldots, y_{N_y} \mid x_1, x_2, \ldots, x_{N_x}\right)$. Seq2seq is essentially an encoder-decoder model.

The encoder first transforms X to a context vector c through a recurrent neural network (RNN), i.e.

$$h_t = f\left(x_t, h_{t-1}\right); c = \phi\left(\{h_1, \ldots, h_T\}\right) \tag{1}$$

where $\{h_t\}$ is the RNN hidden state at time t, f is the dynamics non-linear function, and ϕ summarizes the hidden states. In practice, it is found that gated RNN alternatives such as LSTM [6] or GRU [2] often perform much better than vanilla ones [5]. In this work, the source sentence is encoded with a Bi-directional RNN, making each hidden state h_t aware of the contextual information from both ends.

The decoder then estimates the generation probability of Y with the context vector c as input, through the following dynamics and prediction model:

$$s_t = f\left(y_{t-1}, s_{t-1}, c\right); p\left(y_t | y_{<t}, X\right) = g\left(y_{t-1}, s_t, c\right) \tag{2}$$

where $\{s_t\}$ are the RNN hidden state at time t, and $\{y_{<t}\}$ denoting the history $\{y_1, \ldots, y_{t-1}\}$.

3.2 Attention Mechanism

The traditional seq2seq model takes input as a complete sequence X and compresses all information into a fixed-length vector. In practice, however, different words in Y could be semantically related to different parts of X. To address this issue, the attention mechanism was introduced to seq2seq [1]. It allows seq2seq model to inspect all the information in the input sequence X, then generate the Y according to the current word and context. Each y_i in Y corresponds to a

dynamically changing context vector c_i, which is a weighted average of all hidden states $\{h_t\}$ of the encoder, i.e.

$$c_t = \sum_{\tau=1}^{T} \alpha_{t\tau} h_\tau; \alpha_{t\tau} = \frac{e^{\eta(s_{t-1}, h_\tau)}}{\sum_{\tau'} e^{\eta(s_{t-1}, h_{\tau'})}} \tag{3}$$

where η is usually implemented as a multi-layer perceptron (MLP) with tanh as the activation function.

4 Topic-Net Conversation Model

Let X denotes an input message $X = \{x_1, x_2, \ldots, x_{N_x}\}$, where N_x denotes the number of words in X, and K denotes the topic representation of the message X. Let Y denotes a sequence in response to the message X, where $Y = \{y_1, y_2, \ldots, y_{N_y}\}$ and N_y is the length of the response. Our goal is to learn a generation model to generate response candidates for X with the topic representation K.

To learn the model, we need to deal with two questions: (1) how to obtain topic words? (2) how to perform learning? In the following sections, we first present our method on topic word generation, then we give details of our seq2seq model.

4.1 Topic Generation Model

Our topic generation model is based on neural language model TDLM [9]. TDLM is a topically driven neural language model with a convolutional neural network topic model, which is concise and efficient. We do some improvements on the model to make the topic more suitable for the conversational model, as shown in Fig. 1.

We assume that the input message is $X = \{x_1, x_2, \ldots, x_i, \ldots, x_{N_x}\}$, where $x_i \in \mathbb{R}^e$ is the e-dimensional word vector for the i-th word in the message. Then we can use a number of convolutional filters to process the word vectors. Let $W_v \in \mathbb{R}^{eh}$ be a convolutional filter which is applied to a window of m words to generate a feature. A feature u_i for a window of words $x_{i:i+m-1}$ is given as follows:

$$u_i = \delta\left(W_v^T x_{i:i+m-1} + b_v\right) \tag{4}$$

where b_v is a bias term and δ is the activation function, which is generally the Relu function. Then we apply a max-pooling operation, yielding the message vector d:

$$d = \max_i u_i \tag{5}$$

The topic vectors are stored in two lookup tables $A \in \mathbb{R}^{k \times a}$ (input vector) and $B \in \mathbb{R}^{k \times b}$ (output vector), where k is the number of topics, a and b are the dimensions of the topic vectors. The attention vector p can be written as

$$p = \gamma\left(Ad\right) \tag{6}$$

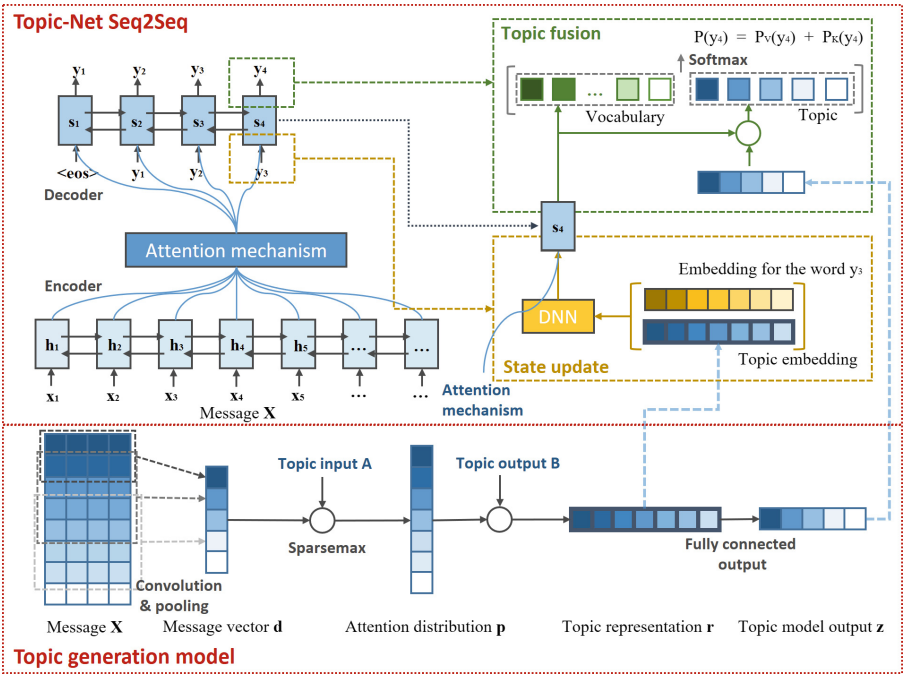

Fig. 1. The structure of the topic-net conversation model.

where $p \in \mathbb{R}^k$ and γ is usually the softmax function. But here we use sparsemax function [14], which is similar to the traditional softmax and outputs sparse probabilities.

Then, we calculate the topic representation r by

$$r = B^T p \tag{7}$$

where $r \in$. Intuitively, r is a weighted mean of topic vectors, with the weighting given by the attention p.

At last, r is connected to a dense layer to generate the topic word, and the model is optimized by using categorical cross-entropy loss.

In addition, considering that the conversation sequences are generally short, we introduce a penalty item similarity penalty to ensure that the similarity between topics is small enough. The output vector $B \in \mathbb{R}^{k \times b}$ determines the similarity of the topic representation. A row in B represents a topic. Therefore, we normalize B by

$$\tilde{B}_{ij} = \frac{B_{ij}}{\| B_i \|^2} \tag{8}$$

The cosine similarity of the topic representations is $\tilde{B}\tilde{B}^2$. Therefore, we can penalize the similarity by

$$Penality = \beta max\left(\tilde{B}\tilde{B}^T - E\right)^2 \tag{9}$$

where E is the identity matrix; and β is the penalty parameter.

4.2 Topic-Net Seq2Seq Model

As illustrated in Fig. 1, topic-net seq2seq model is built on the sequence-to sequence framework, which is still an encoder-decoder model.

In encoder, as same as [1], we use a bi-directional RNN to convert the input sequence X to a series of hidden states $\{h_t\}$ with equal length, specifically each hidden state h_t corresponding to word x_t. Then these hidden states $\{h_t\}$ are transformed to context vectors.

In decoder, a RNN predicts the target sequence with the context vector, which is similar with [1]. But, there are two important differences: the state update module and the topic fusion module. The state update module integrates the topic into the hidden states $\{s_t\}$. The topic fusion module predicts words based on a mixed probabilistic model of two modes, namely the generate-mode and the topic-mode, where the former is the traditional probabilistic model and the latter picks words from the embedding of the topic. It increases the possibility that the topic word appears in the response.

State Update. The state update module optimizes the update process of the hidden states $\{s_t\}$ with the topic. Specifically, it updates each decoding state s_t with the previous state s_{t-1}, the previous symbol y_{t-1}, the context vector c_{t-1} and the topic embedding r, so the Eq. (2) can be rewritten as

$$s_t = f\left(y_{t-1}, s_{t-1}, c, r\right) \tag{10}$$

where r is the topic representation extracted by the topic generation model. Compared to the traditional Eq. (2), the state update module makes each word generated in the response not only relevant to the input message, but also to the topic information.

Topic Fusion. In the traditional decoder, the words of the response y_t is predicted from the vocabulary $V = \{v_1, v_2, \ldots, v_{N_v}\}$. In addition, in the topic fusion module as in Fig. 2, we have another set of topic words K, for all the words extracted by the topic generation model $K = \{k_1, k_2, \ldots, k_{N_k}\}$, where K may contain words not in V.

In the topic fusion module, every word in response is predicted based on a mixed probabilistic model of two modes, namely the generate-mode for the word in V and the topic-mode for the word in K. Given the hidden state s_t at time t, we define the mixture generation probability of the response word y_t as

$$p\left(y_t|y_{t-1}, s_{t-1}, c_t, K\right) = p_V\left(y_t|y_{t-1}, s_{t-1}, c_t\right) + p_K\left(y_t|y_{t-1}, s_{t-1}, c_t, K\right) \tag{11}$$

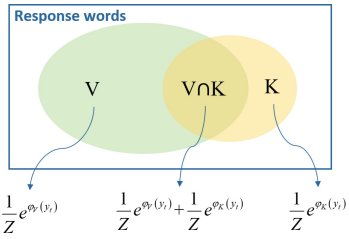

Fig. 2. The illustration of the decoding probability $p(y_t|\cdot)$ in the topic fusion module. V is the vocabulary of the predicted words. And K the vocabulary of the topic words.

where p_V is the generate-mode and p_K is the topic-mode. The probability of the word y_t in these two modes are defined by

$$\begin{aligned} p_V\left(y_t|y_{t-1}, s_t, c_t\right) &= \begin{cases} \frac{1}{Z} e^{\varphi_V(y_t)}, & y_t \in V \\ 0, & y_t \notin V \end{cases} \\ p_K\left(y_t|y_{t-1}, s_t, c_t, K\right) &= \begin{cases} \frac{1}{Z} \sum_{j:k_j=y_t} e^{\varphi_K(y_t)}, & y_t \in K \\ 0, & y_t \notin K \end{cases} \end{aligned} \tag{12}$$

where φ_V and φ_K are the measure functions for the generate-mode and the topic-mode and Z is the normalization term $Z = \sum_{v \in V} e^{\varphi_V(v)} + \sum_{k \in K} e^{\varphi_K(k)}$. These two modes are basically competing through one softmax function. Therefore, they share the same normalization term.

As shown in Fig. 2, the measure functions of the generate-mode and topic-mode are calculated as follow.

For the generate-mode, the measure function is the same as [1]:

$$\varphi_V\left(y_t = v_i\right) = v_i^T W_o s_t \tag{13}$$

where $W_o \in \mathbb{R}^{(N+1) \times d_s}$ and v_i is the one-hot indicator vector for the response word in V.

For the topic-mode, the measure function is defined by

$$\varphi_K\left(y_t = k_i\right) = \delta\left(k_i^T W_c\right) s_t \tag{14}$$

where $W_c \in \mathbb{R}^{d_h \times d_s}$, k_i is the one-hot indicator vector for the response word in K and δ is a non-linear activation function, which is the role of mapping k_i to the semantic space of s_t.

With the topic fusion, the generation probability of response word is partial to the topic words. For the non-topic words, only the generate-mode can be activated, and the measure functions of topic-mode does not work. For the topic words, topic-mode further increases the possibility of the topic words appearing in responses.

4.3 Beam-Search Optimization

As illustrated in Fig. 3, to solve the problem of the difference between the training and the testing process, we reconstruct the response sequence of the train

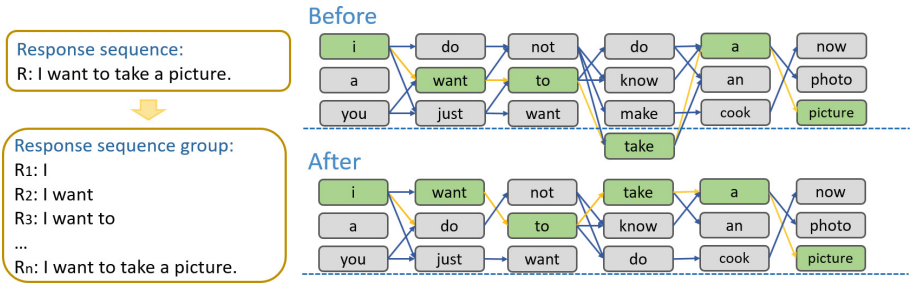

Fig. 3. Beam-search optimization.The left is the schematic diagram of the sequence group. The right is the optimization of the argmax. Right top: possible $y_{1:t}$ formed in training with a beam of size $K = 3$ and with target sequence y1:6 = "i want to take a picture". Note that in time-step $t = 4$ the predicted prefixes involve in margin violations. Right bottom: after grouping, margin violations rarely appear.

dataset, which makes the training process perceive the information of partially-formed sequences.

Specifically, for train dataset, we split each response sequence into a sequence group order by the word, as shown in Fig. 3. The sequence group contains the information of partially-formed sequences, which exposes partially-formed sequences to the training process. In practice, in order to simplify the experiment process, we directly weight each word according to word order in the loss function to achieve the effect of splitting. For each partially-formed sequence in sequence group, the loss can be written as

$$loss_p = -\sum_{t=1}^{N_R} \log p\left(y_t | y_{<t}, X\right) \tag{15}$$

where N_R is the sequence length of every partially-formed sequence. So, the loss of the sequence group is

$$\begin{aligned} loss &= -\sum_{N_R=1}^{N_R=N_y} \sum_{t=1}^{N_R} \log p\left(y_t | y_{<t}, X\right) \\ &= -\sum_{t=1}^{N_y} \left(N_y - i + 1\right) \log p\left(y_t | y_{<t}, X\right) \end{aligned} \tag{16}$$

Furthermore, we replace beam-search with argmax to enhance the performance with better efficiency. In traditional beam-search, the model is never exposed to its own errors during the test process, and so the inferred histories in the test do not resemble the training histories. In Fig. 3, the partially decoded words could not be included in the beam-size of the beam-search before the ending of the test process. However, after exposing partially-formed sequences to the training process, we discover that almost all the partially decoded words emerged in the beam-size of the beam-search have the relatively high scores. Therefore, to enhance efficiency, we substitute the beam-search with argmax, which narrows the search space and simplifies the test process.

5 Experiment

5.1 Experiment Setup

We train and evaluate the models on the Cornell Movie Dialog Corpus [3], which contains a total of 220,579 multi-turn dialogs between 10,292 pairs of movie characters, extracted from 617 original movie screenplays. During preprocessing, sentences with any non-Roman alphabet are removed and few standardizations are made via regular expressions such as mapping all valid numbers to <number>.

We consider the following models as baselines: **(1) bS2SA:** a bidirectional sequence-to-sequence model with attention; **(2) S2SA-MMI**: a sequence-to-sequence model using MMI as the objective function [10]; **(3) bS2SA-Topic Attention (bS2SA-TA):** a bS2SA with topic attention, which synthesizes topic vectors from a pre-trained LDA [29]; **(4) bS2SA-State update(bS2SA-Su):** a bS2SA with state update module, to verify the effectiveness of the topic fusion module of TNCM; **(5) bS2SA-State update & Topic fusion(bS2SA-ST):** a bS2SA with state update module and the topic fusion module, to verify the effectiveness of the beam-search optimization of TNCM.

The accurate evaluation of a non-goal-driven dialogue system is an open problem [4] but not the focus of the paper. We follow the existing work and employ the automatic evaluation metrics include perplexity and Distinct-1 & Distinct-2 [10].

Perplexity. For probabilistic language models word perplexity is a well-established performance metric and has been suggested for generative dialogue models previously [18]. Perplexity is defined by

$$PPL = \exp\left(-\frac{1}{N}\sum_{n=1}^{N}\log\left(p\left(Y_i\right)\right)\right)$$

Perplexity measures how well the model generates a response. It explicitly measures the model?s ability to account for the syntactic structure of the dialogue and the syntactic structure of each utterance [18]. A lower perplexity score indicates better generation performance.

Distinct-1 and Distinct-2. Following [10], we calculate Distinct-1 and Distinct-2. Distinct-1 and Distinct-2 are respectively the number of distinct unigrams and bigrams divided by total number of generated words. The two metrics measure how informative and diverse the generated responses are. High numbers and high ratios mean that there is much content in the generated responses, and high numbers further indicate that the generated responses are long [29].

In addition to automatic evaluation metrics, we also recruit human annotators to determine the quality of responses generated by different models [29]. Five volunteers with rich experience are invited to do this. The volunteers judge the quality of the response according to the following criteria: Good (+2): the response was not only relevant and natural, but also interesting and informative; Medium (+1): the response can be used as a reply to a message, but it is too common, such as “Yes, I see”, “Me too” and “I don’t know”; Bad (+0): the response cannot be used as a reply to the message.

Table 1. Results on automatic metrics.

Model	Perplexity	Distinct-1	Distinct-2
bS2SA	87.24	.095	.199
S2SA-MMI	87.24	.148	.295
bS2SA-TA	81.62	.122	.273
bS2SA-Su	81.03	.160	.297
bS2SA-ST	74.85	**.172**	.382
TNCM	**73.96**	.170	**.383**

5.2 Evaluation Results

Table 1 shows the results of automatic metrics. It is clear that our model has achieved the best performance. On perplexity, bS2SA-State update, bS2SA-State update & Topic fusion and TNCM beat all the baselines, where TNCM is the best. S2SA-MMI is an after-processing mechanism on the responses generated by S2SA, therefore, following [29], we report the perplexity of S2SA to approximately represent the generation ability of S2SA-MMI. It is worth mentioning that bS2SA-TA and our model perform better than bS2SA and S2SA-MMI, which verifies our claim that the topic information does have an effect on conversation models. However, bS2SA-TA is worse than bS2SA-Su and bS2SA-ST; probably because bS2SA-TA leverages the LDA topic, where the probabilistic topic may not be suitable for seq2seq model and the topic information is only introduced between the encoder and decoder. The bS2SA-ST shows better performance than bS2SA-Su, indicating the importance of the topic fusion module. Compared to bS2SA-ST, the perplexity of TNCM reduces a little, suggesting that the beam-search optimization contributes to enriching the response in some extent. On distinct-1 and distinct-2, bS2SA-State update & Topic fusion outperforms all the baseline models, which further illustrates the state update module and the topic fusion module have positive effects. Note that bS2SA-ST performs better than TNCM in term of the number of distinct unigram, which results from the beam-search optimization narrowing the search space of the response words.

Table 2. Results on human evaluation.

Model	Good	Medium	Bad	Ave
bS2SA	20.8%	39.6%	39.6%	.812
S2SA-MMI	32.7%	36.5%	30.8%	1.019
bS2SA-TA	38.4%	29.6%	32.0%	1.064
bS2SA-Su	39.3%	28.1%	32.6%	1.067
TNCM	41.6%	28.3%	30.1%	**1.115**

Table 2 shows the results of human evaluation. Obviously, TNCM are better than all the baseline models. Compared with the S2SA-MMI, bS2SA-TA and bS2SA-Su all generates more "Good" responses, confirming the topic information takes effect on generating interesting and informative responses. However, generic responses ("Medium") of bS2SA-Su reduces and the "Bad" responses of bS2SA-Su increases. It is because that noise in the topic is brought to generation without the topic fusion module. This shows that the topic fusion module is indispensable.

Table 3. Results on the efficiency improvement of the beam-search optimization.

Model	BLEU	Speedup
bS2SA-ST	3.59	10.4x
TNCM	3.62	3.38x

Table 3 shows the results of the efficiency improvement of the beam-search optimization. Beam-search is applied to generating natural language of the response. Therefore, we use BLEU [4] to measure the quality of the natural language. On the one hand, it's obvious that TNCM performs a little bit better than the bS2SA-ST with regards to BLEU. On the other hand, our model has almost 3 times speedup in test inference. This suggests that the beam-search optimization enhances the effect with better efficiency.

6 Conclusions

In this paper, we study the response generation problem and propose a topic-net conversation model (TNCM) to incorporate topic information into the sequence-to-sequence framework. TNCM generates every response word using not only its word-embedding hidden state but also the embedding of the topic and increases the possibility that the topic word appears in the response. Moreover, we improve the process of beam search during the test stage, which enhances the effect with better efficiency. Experimental results show that TNCM can significantly outperform state-of-the-art models and generate more interesting and more informative responses.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)

3. Danescu-Niculescu-Mizil, C., Lee, L.: Chameleons in imagined conversations: a new approach to understanding coordination of linguistic style in dialogs. In: Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics, pp. 76–87. Association for Computational Linguistics (2011)
4. Galley, M., Brockett, C., Sordoni, A., Ji, Y., Auli, M., Quirk, C., Mitchell, M., Gao, J., Dolan, B.: deltaBLEU: a discriminative metric for generation tasks with intrinsically diverse targets. arXiv preprint arXiv:1506.06863 (2015)
5. Gu, J., Lu, Z., Li, H., Li, V.O.: Incorporating copying mechanism in sequence-to-sequence learning. arXiv preprint arXiv:1603.06393 (2016)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
7. Kabir, M.E., Wang, H., Bertino, E.: A role-involved purpose-based access control model. Inf. Syst. Front. **14**(3), 809–822 (2012)
8. Khalil, F., Li, J., Wang, H.: An integrated model for next page access prediction. Int. J. Knowl. Web Intell. **1**(1–2), 48–80 (2009)
9. Lau, J.H., Baldwin, T., Cohn, T.: Topically driven neural language model. arXiv preprint arXiv:1704.08012 (2017)
10. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. arXiv preprint arXiv:1510.03055 (2015)
11. Li, J., Galley, M., Brockett, C., Spithourakis, G.P., Gao, J., Dolan, B.: A persona-based neural conversation model. arXiv preprint arXiv:1603.06155 (2016)
12. Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., Jurafsky, D.: Deep reinforcement learning for dialogue generation. arXiv preprint arXiv:1606.01541 (2016)
13. Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., Jurafsky, D.: Adversarial learning for neural dialogue generation. arXiv preprint arXiv:1701.06547 (2017)
14. Martins, A., Astudillo, R.: From Softmax to Sparsemax: a sparse model of attention and multi-label classification. In: International Conference on Machine Learning, pp. 1614–1623 (2016)
15. Nio, L., Sakti, S., Neubig, G., Toda, T., Adriani, M., Nakamura, S.: Developing non-goal dialog system based on examples of drama television. In: Mariani, J., Rosset, S., Garnier-Rizet, M., Devillers, L. (eds.) Natural Interaction with Robots, Knowbots and Smartphones. Springer, New York (2014)
16. Perez-Marin, D.: Conversational Agents and Natural Language Interaction: Techniques and Effective Practices: Techniques and Effective Practices. IGI Global (2011)
17. Ritter, A., Cherry, C., Dolan, W.B.: Data-driven response generation in social media. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 583–593. Association for Computational Linguistics (2011)
18. Serban, I.V., Sordoni, A., Bengio, Y., Courville, A.C., Pineau, J.: Building end-to-end dialogue systems using generative hierarchical neural network models. In: AAAI 2016, pp. 3776–3784 (2016)
19. Shang, L., Lu, Z., Li, H.: Neural responding machine for short-text conversation. arXiv preprint arXiv:1503.02364 (2015)
20. Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.Y., Gao, J., Dolan, B.: A neural network approach to context-sensitive generation of conversational responses. arXiv preprint arXiv:1506.06714 (2015)
21. Sun, X., Li, M., Wang, H., Plank, A.: An efficient hash-based algorithm for minimal k-anonymity. In: Proceedings of the Thirty-First Australasian Conference on Computer Science, vol. 74, pp. 101–107. Australian Computer Society, Inc. (2008)
22. Sun, X., Wang, H., Li, J., Zhang, Y.: Injecting purpose and trust into data anonymisation. Comput. Secur. **30**(5), 332–345 (2011)

23. Sun, X., Wang, H., Li, J., Zhang, Y.: Satisfying privacy requirements before data anonymization. Comput. J. **55**(4), 422–437 (2012)
24. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems, pp. 3104–3112 (2014)
25. Vinyals, O., Le, Q.: A neural conversational model. arXiv preprint arXiv:1506.05869 (2015)
26. Wang, H., Cao, J., Zhang, Y.: Ticket-based service access scheme for mobile users. In: Australian Computer Science Communications, vol. 24, pp. 285–292. Australian Computer Society, Inc. (2002)
27. Wang, H., Zhang, Y., et al.: Detection of motor imagery eeg signals employing naïve bayes based learning process. Measurement **86**, 148–158 (2016)
28. Wu, Y., Wu, W., Li, Z., Zhou, M.: Response selection with topic clues for retrieval-based chatbots. arXiv preprint arXiv:1605.00090 (2016)
29. Xing, C., Wu, W., Wu, Y., Liu, J., Huang, Y., Zhou, M., Ma, W.Y.: Topic aware neural response generation. In: AAAI 2017, pp. 3351–3357 (2017)
30. Young, S., Gašić, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., Yu, K.: The hidden information state model: a practical framework for pomdp-based spoken dialogue management. Comput. Speech Lang. **24**(2), 150–174 (2010)
31. Young, S., Gašić, M., Thomson, B., Williams, J.D.: POMDP-based statistical spoken dialog systems: a review. Proc. IEEE **101**(5), 1160–1179 (2013)