Computer Science and Information Systems 17(2):537–552 https://doi.org/10.2298/CSIS200131012Z

Variational Neural Decoder for Abstractive Text Summarization

Huan Zhao, Jie Cao, Mingquan Xu, and Jian Lu

College of Computer Science and Electronic Engineering, Hunan University,
Changsha, Hunan, P.R. China, 410000
{hzhao,jiecao,hb_xmq,jianlu}@hnu.edu.cn

Abstract. In the conventional sequence-to-sequence (seq2seq) model for abstractive summarization, the internal transformation structure of recurrent neural networks (RNNs) is completely determined. Therefore, the learned semantic information is far from enough to represent all semantic details and context dependencies, resulting in a redundant summary and poor consistency. In this paper, we propose a variational neural decoder text summarization model (VND). The model introduces a series of implicit variables by combining variational RNN and variational auto-encoder, which is used to capture complex semantic representation at each step of decoding. It includes a standard RNN layer and a variational RNN layer [5]. These two network layers respectively generate a deterministic hidden state and a random hidden state. We use these two RNN layers to establish the dependence between implicit variables between adjacent time steps. In this way, the model structure can better capture the complex semantics and the strong dependence between the adjacent time steps when outputting the summary, thereby improving the performance of generating the summary. The experimental results show that, on the text summary LCSTS and English Gigaword dataset, our model has a significant improvement over the baseline model.

Keywords: abstractive summarization, sequence-to-sequence, variational auto-encoder, variation neural inferer.

1. Introduction

Text summarization produces a brief summary of the core ideas of the source articles and is different from extractive text summarization ([4], [23],[30], [19]), which selects key sentences or key phrases in the original text to form a summary. Abstractive text summarization builds an internal semantic representation and then uses deep learning techniques to create a summary that is closer to what a human may generate. Most recent models for abstractive text summarization are based on the seq2seq framework with attention ([2],[21],[12],[29]). These seq2seq models consist of an encoder and a decoder; the encoder encodes input text into a semantic representation, and the decoder generates summaries from this representation.

With the development of deep learning, the neural networks-based encoder-decoder models are used in the sequence-to-sequence tasks, such as neural machine translation[28], speech recognition ([32],[33]) and text summarization ([3],[9],[25]). The Seq2seq framework for abstractive text summarization has recently achieved remarkable success and has become the dominant architecture. In the seq2seq framework, the semantic representation from the encoding end to the decoding end is learned in an implicit way, in which the internal transformation structure of recurrent neural networks (RNNs) is completely determined. [5]. Therefore, the learned semantic representations are poor at capturing all semantic details and dependencies [29],[14]. To address the insufficiency of semantic representations of abstractive text summarization, [20] proposed a generative model to capture the latent summary information, but they considered only a single latent variable for capturing the global semantics of each source text in their generative model, which led to limited representation ability. Furthermore, [14] introduced a seq2seq framework with a deep recurrent generative decoder that considered the recurrent dependencies in their generative model for capturing historical latent variable dependencies. Although this approach can obtain more latent structure information, in practice, long-term sequential recurrent dependencies can result in the loss of previous information and unnecessary noise; hence, this implementation may not be sufficient for capturing strong and complex semantic dependencies between adjacent target words at each time step of the decoding.

To tackle the problem, we present a variational neural decoder (VND) for abstractive text summarization that is more effective at forcing the decoder to make use of latent structure information. We introduce the variational autoencoder (VAE) [11],[24] process to the decoding process and use latent variables to model the complex potential distribution of text semantics at each time step. Drawing inspiration from the current success of the variational RNN (VRNN) [5], we incorporate latent variables into the RNN hidden state. By using latent variables, the VRNN can model the underlying semantics of source or target texts. Our decoding structure consists of variational neural inferers and two RNN layers: a standard RNN layer and a VRNN layer. A variational neural inferer is employed to address the intractable posterior inferer for the latent variables. The standard RNN layer generates a deterministic hidden state, which is employed to model long- and short-term dependencies. The stochastic latent hidden state based on the VRNN layer is used to capture complex and strong potential semantic distributions and is integrated into the summary generation softmax layer to improve the summary generation quality. Specifically, at each time step, we use the stochastic latent hidden state of the VRNN layer in the previous step as the input of the current RNN layer. This implementation integrates the dependencies between the latent variables in adjacent timesteps.

The main contributions of this paper are as follows: (1) we propose a VND model that efficiently captures the complex semantics and strong dependencies between neighboring target words for abstractive text summarization. (2) Experiments on the LCSTS dataset and English Gigaword for the text summarization task show that our proposed model significantly outperforms the baseline models.

2. Related Work

Automatic text summarization is one of the most active research in natural language processing. It produces a concise and smooth summary while preserving key information content and overall meaning [1]. Recently, an increasing number of researchers have employed a neural network framework to natural language processing. Sequence-to-sequence neural networks[29] have been applied to machine translation ([28],[17]), following their success in abstractive summarization ([2],[13]).

Specifically, [25] first proposed a convolution encoder and a recurrent decoder model for the abstractive sentence summarization task, which has achieved significant performance improvement over conventional methods, and provides the benchmark for the Gigaword dataset. [22] replaced the model with a full RNN seq2seq model and achieved outstanding performance. [17] proposed an attention mechanism, which greatly improved the performance of the seq2seq model on abstractive summarization. To address the unknown word problem, [21] proposed a generator-pointer model so that the decoder can generate words in source texts. [6] also solved this problem by integrating a copying mechanism into a seq2seq model. [27] propose an LSTM-CNN based seq2seq model that can construct new semtences by exploring more fine-grained fragments than sentences, namely, semantic phrases. [18] proposed a neural model to improve the semantic relevance between the source contents and the predicted summaries.

Some other work attempts to incorporate Variational auto-encoder for abstractive summarization. The variational auto-encoder is a popular probabilistic generative model ([5],[11]). These models utilize an neural inference model to approximate the intractable posterior, and optimize model parameters jointly with a reparameterized variational lower bound using the standard stochastic gradient technique. Due to its success in various tasks, this method has attracted increasing attention. Although seq2seq-oriented encoder-decoder framework has been developed and has widely used in abstractive text summarization, there are few research works incorporated variational anto-encoder into the text summary system. For example,[20] first proposed a generative model to capture the latent summary information based on the seq2seq framework. [26] presented an unsupervised approach to summarize sentences abstractively using a VAE. Furthermore, [5] extended the VAE into a recurrent framework for modeling complex semantic representations, which is called VRNN. [14] proposed a deep recurrent generative decoder to capture latent structure information.

Inspired by the successful application of variational auto-encoder in related works, we propose variational neural decoder for abstractive summarization. This paper proposes a variational neural decoder model, which introduces a series of continuous latent variables to capture the latent semantics of the content to improve the quality of the summary.

3. Background: Variational Autoencoder

The VAE [11],[24] is a recently introduced latent variable generative model, which combines Variational Inference with Deep Learning. In VAE, a generative network models an observed variable x as a continuous latent variable z, based on which the generate network reconstructs x. Then, the join distribution density function of the generated model is as follows:

$$p_\theta(x, z) = p_\theta(x|z)p_\theta(z), \tag{1}$$

where $p_\theta(z)$ is the probability density function of z prior distribution of latent variable, $p_\theta(x|z)$ is the conditional probability density function of x when z is known, and θ is the parameter of two density functions. In general, we assume that $p_\theta(z)$ and $p_\theta(x|z)$ are the standard standard Gaussian distribution that models the generation procedure, which is typically estimated via a deep nonlinear neural network.

Importantly, the VAE models the conditional distribution $p_\theta(x|z)$ as a highly flexible function approximator, which makes the inference of the posterior $p_\theta(z|x)$ intractable.

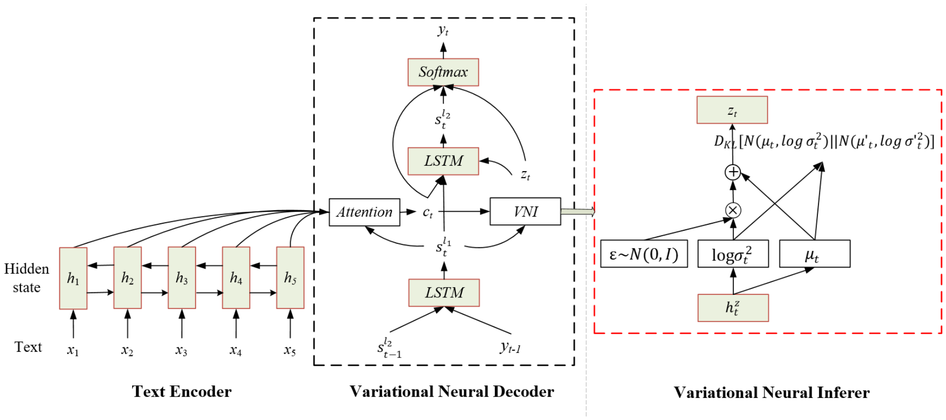

Fig. 1. The overview of the architecture of VAD model.

Thus VAE uses a variational approximation $q_\phi(z|x)$ of the posterior, which introduces the evidence lower bound:

$$\mathcal{L}_{VAE}(\theta, \phi, x) = -D_{KL}(q_\phi(z|x)||p_\theta(z)) + E_{q_\phi(z|x)}[\log p_\theta(x|z)] \leqslant \log p_\theta(x), \quad (2)$$

where θ and ϕ denote the parameters of the model and $D_{KL}(Q||P)$ is the Kullback-Leibler divergence between two distributions Q and P.

In [11], the approximate posterior $q_\phi(z|x)$ is a diagonal Gaussian $\mathcal{N}(\mu, diag(\sigma^2))$, whose mean μ and variance σ^2 are the output of a highly nonlinear function of x. The VAE training process maximizes the ELBO, which obtains the optimal parameter selection for the generative model $p_\theta(x|z)$ and inference model $q_\phi(z|x)$. Based on reparametrizing, we compute $z = \mu + \sigma \odot \epsilon$ and rewrite the equation as

$$E_{q_\phi(z|x)}[\log p_\theta(x|z))] = E_{p_\theta(\epsilon)}[\log p_\theta(z = \mu + \sigma \odot \epsilon)], \quad (3)$$

where ϵ is a vector of standard Gaussian variables. Then, the VAE model can be trained through a standard backpropagation technique for stochastic gradient descent.

4. Proposed Model

4.1. Overview

Our model is based on the seq2seq model with attention. The seq2seq model can compress source texts $x = \{x_1, x_2, ..., x_M\}_N$ into a continuous vector representation with an encoder, and then the decoder generates the summary text $y = \{y_1, y_2, ..., y_T\}_N$. As shown in Figure 1, VND model mainly contains three neural network based components: the text encoder for encoding text sentences, a variational neural decoder that generate a summary, a variational neural inferer for the posterior and the prior distributions.

4.2. Text Encoder

The text encoder builds meaningful representations of the source sentences. In our model, we use bidirectional long short-term memory ($LSTM$) [8] to encode the source text sequence x from both directions and compute the hidden states for each word, which produces the final hidden state $h = \{h_1, h_2, ..., h_M\}_N$ from the source text x :

$$\overrightarrow{h_i} = LSTM(\overrightarrow{h_{i-1}}, x_i, \overrightarrow{C_{i-1}}), \tag{4}$$

$$\overleftarrow{h_i} = LSTM(\overleftarrow{h_{i+1}}, x_i, \overleftarrow{C_{i+1}}), \tag{5}$$

$$h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}], \tag{6}$$

where $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ are the forward and the backward hidden outputs, respectively, x_i is the input at the i-th time step, N is the number of samples, and C_{i-1} refers to the cell state in the $LSTM$ layer.

The transition equations of $LSTM$ are defined as follows:

$$I_i = \sigma(W_I x_i + U_I h_{i-1} + b_I) \tag{7}$$

$$F_i = \sigma(W_f x_i + U_f h_{i-1} + b_f) \tag{8}$$

$$O_i = \sigma(W_o x_i + U_o h_{i-1} + b_o) \tag{9}$$

$$C_i = F_i \odot C_i + I_i \odot tanh(W_c x_i + U_c h_{i-1} + b_c) \tag{10}$$

$$H_i = O_i \odot tanh(C_i) \tag{11}$$

where $\odot$ stands for element-wise multiplication, σ is the sigmoid function, all $W \in \mathbb{R}^{d \times l}$ and $W \in \mathbb{U}^{d \times d}$ are weight matrices, all $b \in \mathbb{R}^d$ are bias vector.

4.3. Variational Neural Inferer

In order to integrate the stochastic latent variables into our decoder model, we use the variational neural inference to generate the latent variables at each time step of the decoding structure. As described in the background section, the key of variational models is to model the distributions related to latent variables. The variational neural inferer can be divided into two parts: the posterior and the prior distributions. Both posterior and prior distributions are assumed to be multivariate Gaussian distributions with diagonal covariance, but they introduce different parameters. As determined by the ELBO equation, the parameters of the prior are computed by the prior network, which takes the source sentence x and previously generated words $y_{<t}$ as the input. The posterior parameters are also determined from both the source sentence x and previously generated words $y_{<t}$.

Inspired by some ideas in previous works [11],[24]. We use variational neural inference to generate latent variables z in the model, as shown in part variational neural inferer of the Fig. 1, we focus on the use of neural network to simulate the posterior $q_\phi(z_t|x, y_{<t})$ and the prior $p_\theta(z_t|x, y_{<t})$. In this subsection, we use a similar network architecture to that proposed in variational recurrent neural machine translation (VRNMT) [28].

Neural Posterior. Following the standard VAE, we use neural networks for a better approximation in our model. The equation of $q_\phi(z_t|x, y_{<t})$ can be expressed as

$$q_\phi(z_t|x, y_{<t}) = \mathcal{N}(z_t; \mu_t(x, y_{<t}), \sigma_t(x, y_{<t})^2 I), \tag{12}$$

where the μ_t and σ_t denote the variational mean and standard derivation, respectively, which are computed via a neural network based on the observed variables x and $y_{<t}$.

Starting from the VAE, the key to estimating z_t is to calculate the μ_t and σ_t. First, we perform a nonlinear transformation that projects the word embedding y_{t-1}, the deterministic hidden states $s_t^{l_1}$ and the attention content c_t onto our concerned latent semantic space:

$$h_t^z = g(W_z[y_{t-1}; s_t^{l_1}; c_t] + b_z). \tag{13}$$

Then, the above-mentioned Gaussian parameters μ_t and $log\sigma_t^2$ are calculated through linear regression:

$$\mu_t = W_\mu h_t^z + b_\mu, \tag{14}$$

$$\log \sigma_t^2 = W_\sigma h_t^z + b_\sigma, \tag{15}$$

where W_z, $W_\mu$and W_σ comprise the parameter matrix and b_z, b_μ and b_σ are bias terms. $g(\cdot)$ refers to a nonlinear function. Then, to obtain a representation for the latent variable z_t using reparameterization [24], the latent variables can be expressed as:

$$z_t = \mu_t + \log \sigma_t^2 \odot \epsilon, \epsilon \sim \mathcal{N}(0, I). \tag{16}$$

Intuitively, this reparameterization reduces the gap between $q_\phi(z_t|x, y_{<t})$ and $p_\theta(z_t)$. In other words, it connects these two neural networks. This is important since it enables the stochastic gradient optimization via standard backpropagation.

Neural Prior. The neural model for the prior $p_\theta(z_t|x, y_{<t})$ is the same as that for the posterior $q_\phi(z_t|x, y_{<t})$. Here, we model the prior $p_\theta(z_t|x, y_{<t})$ as

$$p_\theta(z_t|x, y_{<t}) = \mathcal{N}(z_t; \mu_t'(x, y_{<t}), \sigma_t'(x, y_{<t})^2 I). \tag{17}$$

To obtain a representation for latent variable z_t, we first use the same method to employ the latent semantic space:

$$h_t^{z\prime} = g(W_z'[y_{t-1}; s_t^{l_1}; c_t] + b_z) \tag{18}$$

Then, Gaussian parameters μ_t' and $\log \sigma_t'^2$ in the model are computed by:

$$\mu_t' = W_\mu' h_t^{z\prime} + b_\mu', \tag{19}$$

$$\log \sigma_t'^2 = W_\sigma' h_t^{z\prime} + b_\sigma', \tag{20}$$

where W_z', W_μ' and W_σ' comprise the parameter matrix and b_z', b_μ' and b_σ' are bias terms. Different from the posterior, we directly set z_t as μ_t', as implemented in [31].

Finally, we integrate the latent variable z_t into decoding our model to enhance the summary generation results, which are described in detail in the following subsection.

Variational Lower Bound. As in the conventional VAE, we learn the generative and inference models jointly by maximizing the variational lower bound with respect to their parameters. We apply ELBO at each t-th time step, and based on factorizations (12) and (17), we have the accumulative ELBO as follows:

$$\mathcal{L}_{VAE} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T} D_{KL}(q_\phi(z_t^{(n)}|x^{(n)}, y_{<t}^{(n)})||p_\theta(z_t^{(n)}|x^{(n)}, y_{<t}^{(n)}))). \tag{21}$$

In the model training, $\mathcal{L}_{VAE}$ is part of the objective function.

4.4. Variational Neural Decoder

The function of the decoder is to generate a series of summary words. As shown in the VND section of Figure 1, at each decoding time step t, the decoder consists of two RNN layers.

The first layer is a standard RNN layer. Given the previously generated word embedding y_{t-1} and the previous stochastic latent hidden state $s_{t-1}^{l_2}$, the standard RNN layer is used to calculate the deterministic hidden state $s_t^{l_1}$ at the t-th time step:

$$s_t^{l_1} = LSTM_1([y_{t-1}; s_{t-1}^{l_2}], C_{t-1}^{l_2}), \tag{22}$$

where the superscript l denotes the decoder LSTM layer.

Next, we apply the dot attention mechanism [17] to obtain the content vector. Then, the deterministic hidden state $s_t^{l_1}$ and the encoder output h_i at each time step t of the process are computed as the attention weight $\alpha_{t,i}$ and the current content vector c_t, respectively:

$$c_t = \sum_{i=1}^{m} \alpha_{t,j} h_i, \tag{23}$$

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{n} \exp(e_{t,j})}, \tag{24}$$

$$e_{t,i} = {s_t^{l_1}}^{\mathrm{T}} W_e h_i, \tag{25}$$

where W_e is the weight parameter and $\alpha_{t,i}$ is the i-th word of the attention mechanism that assigns weight at time step t.

The second layer is the variational RNN layer. In [5], the author extended the VAE into a recurrent framework for modeling complex semantic representations, which was called VRNN. In VRNN, at each t-th time step, the transition function (LSTM cell) computes the next hidden state based on the previous hidden state and the sampled latent random variables. Inspired by [5], we first combine the deterministic hidden state $s_t^{l_1}$, the current content vector c_t, and the latent variable z_t (implemented in subsection 3.3) to construct a new latent semantics vector $\hat{s}_t$:

$$\hat{s}_t = W_c c_t + W_s s_t^{l_1} + W_z z_t + b_s, \tag{26}$$

where W_y, W_c, W_s, and W_z are the weight matrices and b_s is the bias term. Then, a VRNN layer is used with the latent semantics vector $\hat{s}_t$ to compute the stochastic latent hidden state $s_t^{l_2}$:

$$s_t^{l_2} = LSTM_2(\hat{s}_t, C_t^{l_1}). \tag{27}$$

Finally, to precisely generate summaries, the softmax layer is introduced to generate the target word y_t based on the latent variable z_t, the current context vector c_t and the stochastic latent hidden state $s_t^{l_2}$. We compute the probability distribution over the target word y_t:

$$p(y_t|z_t, y_{<t}, x) = softmax(W_v \hat{u}_t + b_v), \tag{28}$$

$$\hat{u}_t = g(W_d[z_t; s_t^{l_2}; c_t] + b_d), \tag{29}$$

where W_v and W_d comprise the parameter matrix of the output layer, b_v and b_d are the bias terms, and $g(\cdot)$ is the nonlinear activation function.

4.5. Objective Function

The objective function of our model consists of two terms. The first objective is the variational lower bound $\mathcal{L}_{VAE}$ in Equation 21. This term is the KL divergence between two Gaussian distributions, which can be computed and differentiated without estimation [11]. The second objective is the maximum likelihood estimation of the generated summaries. Given the latent variable z_t at each time-step and source text x, the models generate a summary $\tilde{y}$. The learning process is to minimize the negative log-likelihood between the generated summary $\tilde{y}$ and the reference y:

$$\mathcal{L}_{Seq} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}\{p(y_t^{(n)}|\tilde{y}_{<t}^{(n)}, x^{(n)}, z_t^{(n)})\}. \tag{30}$$

Finally, the objective function, which need to be minimized, is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{VAE} + \mathcal{L}_{Seq}. \tag{31}$$

5. Experiments

5.1. Datasets

LCSTS is a large-scale Chinese short text summarization dataset constructed by [9]. The dataset was collected from a famous Chinese social media website called Sina Weibo and consists of more than 2.4 M text-summary pairs. It is split into three parts, with 2,400,591 pairs in PART I, 10,666 pairs in PART II and 1,106 pairs in PART III. All text-summary pairs in PART II and PART II are manually annotated with relevant scores ranging from 1 to 5. We reserve only pairs with scores of no less than 3, leaving 8,685 pairs in PART II and 725 in PART III. In our experiments, we selected PART I as the training set, PART II as the validation set, and PART III as the test set.

English Gigaword is a sentence summarization dataset based on Annotated Gigawords [22], a dataset consisting of sentence pairs representing the first sentence of collected news articles and their corresponding headlines. We use the dataset preprocessed by [25], which contains 3.8M training pairs, and 189K validation pairs and 2K test pairs were randomly selected. In the data processed directly with [25], we find that there are abnormal data pairs in the corpus, e.g., some unreadable or incomprehensible pairs. Therefore, we reprocessed the data set, therein retaining 3.2M training pairs, a 16K-pair validation set, and a 1,520-pair test set.

5.2. Evaluation Metrics

We employ the recall-oriented understudy for gisting evaluation (ROUGE) score[15] as our evaluation metric with standard options. ROUGE measures the quality of a summary by computing overlapping lexical units, such as unigram, bigram, trigram, and longest common subsequence (LCS). Following previous work [9], our evaluation metrics in the experimental results are the F1 scores of ROUGE: ROUGR-1 (unigram), ROUGE-2 (bigram) and ROUGE-L (LCS).

5.3. Experimental Settings

For the LCSTS dataset, to avoid the effect of Chinese word segmentation error, both our encoder and decoder input texts are Chinese characters. The vocabularies are extracted from training sets, and the source texts and the summaries do not share the same vocabularies. We prune resource and target vocabularies of 8K and 5K words, respectively.

For the English Gigaword dataset , we prune the resource and target vocabularies of 40K and 30K words, respectively. The input word embedding is initialized randomly and learned during the optimization process.

In both experiments, we set the word embedding size and the hidden size to 512, and the number of LSTM layers is 3. The batch size is set to 64, and we do not use dropout on this dataset. We implement the beam search and set the beam size to 10. We use the Adam optimizer [10] to learn the model parameters with the default setting $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-8}$. The learning rate is halved every epoch. Our experiments are implemented using PyTorch.

5.4. Baselines

To evaluate the performance of our variational neural decoder (**VND**) model, we compare our results with the results of the baselines and state-of-the-art methods on the LCSTS dataset and Gigaword dataset.

– **ABS** and **ABS+** [25] are the Seq2seq model with attention mechanism and handcrafted features, which are trained on Gigaword to produce summaries.
– **RNN** and **RNN-context**: [9] are the neural network framework, where RNN+context has the attention mechanism.
– **RNN+distract** [3] employ a new attention mechanism by distracting the historical attention in the decoding steps.
– **DRGD** [14] is a deep recurrent generative decoder model, combining the decoder with a variational auto-encoder.
– **ASC+FSC**$_1$ [20] uses a generative method to model the latent summary variables. The generative model first draws a latent summary sentence from a background language model and subsequently draws the observed sentence conditioned on this latent summary.
– **ARL** [7] is based on the seq2seq framework, which applies the adversarial reinforcement learning strategy to bridge the gap between the generated summary and the human summary.
– **CGU** [16] is a seq2seq model with a convolutional gated unit for global encoding.
– **Seq2seq** is our implementation of the attention mechanism-based seq2seq model, which has a similar setting as our model for comparison.

5.5. Results Analysis

Results on LCSTS Corpus. The summary ROUGE-F1 score comparison of different models on the LCSTS dataset is shown in Table 1. The experimental results show that our VND model has a significantly improved score in each ROUGE compared to the base model.First, we compare the model with the seq2seq baseline. The results show that

Table 1. Comparison with other models on the LCSTS test set. **R-1**, **R-2** and **R-L** denote ROUGE-1, ROUGE-2 and ROUGE-L,respectively.

Models	**R-1**	**R-2**	**R-L**
RNN [9]	21.5	8.9	18.6
RNN-context [9]	29.9	17.4	27.2
RNN+distract [3]	35.2	22.6	32.5
DRGD [14]	37.0	24.2	34.2
ARL [7]	39.4	21.7	29.1
CGU [16]	39.4	26.9	36.5
Seq2seq (our impl.)	33.2	22.4	31.8
VND (our model)	**42.5**	**28.5**	**37.6**

the VND model is significantly improved over the Seq2seq model, with ROUGE-1 9.3% higher, ROUGE-2 6.1% higher, and ROUGE-L 5.8% higher. It proves that our model is effective. Then, we also compare the VND model with the remaining base models.

Next, compared with the DRGD model, our VND model exceeds 5.5% on ROUGE-1, 4.3% on ROUGE-2 and 3.4% on ROUGR-L. In the DRGD model, although the author uses a deep recursive generative model to capture the underlying semantics to improve the quality of the summary, its expressive power is limited and no significant improvement can be achieved. Finally, the ARL model uses adversarial reinforcement learning to optimize parameters on the basis of generating an adversarial network. It is currently the most cutting-edge summary generation model, but the VND model still has obvious advantages. The results show that random latent variables and two RNN layer components plus Seq2seq text summarization model can improve the accuracy of text summarization, which shows the effectiveness of variational neural decoder.

In order to further prove that our proposed model can capture more semantic details and enhance the dependency between the time parts before and after decoding to improve the quality of the summary. We give two summary examples of models randomly drawn from the test set. The original text, the original abstract, the abstract generated by the Seq2seq model and the abstract generated by the model in this chapter are shown in Table 2.

In the first example, the Seq2seq model generates a summary containing the phrases ”big bank” and ”stock market storm,” which is tedious and inaccurate. For comparison, the VND model refines such phrases into more concise and natural phrases. In addition, we can see that the abstract generated by the model is closer to the semantics of the reference abstract than the abstract of the Seq2seq model.

In the second example, the main idea of the original text is that China bans tobacco advertising and imposes fines of up to 30,000 yuan on movies and TV series. However, the summary generated by the seq2seq model only contains information about ”TV programs”. In addition, it also missed the most important information ”forbidden”, which makes the semantics very different from the original text, and is not coherent and sufficient. In contrast, the summary of our model is more coherent and streamlined. In addition, in the summary generated by the VND model, the number (30000) is the same as the number in the reference text, which means that the integration of random hidden variables into the decoder model can obtain richer semantics and can make the decoding

Table 2. Examples of the generated summaries on the test set of the LCSTS corpus, compared with that of the Seq2seq model and the reference.

Source(1): 央行今日将召集**大型商业银行和股份制银行**开会，以应对当前的债市风暴。消息人士表示，央行一方面旨在维稳银行间债券市场，另一方面很可能探讨以丙类户治理为重点的改革内容。此次债市风暴中，国家审计署扮演了至关重要的角色。 The central bank will convene **large commercial banks and joint-stock banks** today to deal with the current debt market turmoil. Sources said that on the one hand, the central bank aims to stabilize the inter-bank bond market, on the other hand, it is likely to explore the reform content focusing on the governance of class C households. The National Audit Office played a vital role in the debt market turmoil. **Reference:** 媒体称央行今日召集**银行开会**应对当前债市风暴 The media said that the Central Bank called **a meeting of banks** today to deal with the current debt market turmoil. **Seq2seq:** 央行召集**大型银行和股份制银行**应对债市风暴 The Central Bank calls **large banks and joint-stock banks** to deal with the storm of the bond market **VND:** 央行今日召集**银行开会**应对当前债市风暴 The Central Bank convened **a meeting of banks** today to deal with the current turmoil in the bond market
Source(2): 国务院法制办公布《公共场所控制吸烟条例(送审稿)》：禁止所有烟草广告、促销和赞助；没有设置室外吸烟点的视为全面禁止吸烟；违反《条例》规定，**电影电视剧**播放吸烟镜头最高罚3万；禁止通过自动售货机等任何方式向未成年人售烟。 The legislative affairs office of the state council announced the ”regulations on control of smoking in public places (submitted for review): ban all tobacco advertising, promotion and sponsorship; if there is no outdoor smoking point, it will be considered as a total ban on smoking. Those who violate the regulations can be fined up to 30,000 yuan for smoking in **movies and TV series**. Selling cigarettes to minors through vending machines or any other means is prohibited. **Reference:** 我国拟全面**禁止烟草广告影视剧**播吸烟最高罚3万 China intends to totally ban tobacco in advertisements, **movies and TV series**, The maximum penalty for smoking is 30,000 yuan. **Seq2seq:** 我国拟规定**电视剧**播放吸烟镜头最高罚#万 China intends to stipulate a maximum penalty of # for smoking scenes in **TV series**. **VND:** 我国拟规定**禁止电影电视剧**播放吸烟镜头最高罚3万 China intends to stipulate a maximum penalty of 30,000 yuan for smoking scenes in **movies and TV series**.

process adjacent The semantic dependence of the time step is stronger, which makes the generated summary coherent and of high quality.

In addition to the Chinese data set, we also provide several examples of randomly generated summary on the English English Gigaword data set. Coherent and more descriptive. We believe that this is mainly because the variational neural decoder can pay

Table 3. ROUGE scores of different models on the English Gigawords dataset.

Models	**R-1**	**R-2**	**R-L**
ABS [25]	29.6	11.3	26.4
ABS+ [25]	29.8	11.9	27.0
ASC+FSC$_1$ [20]	34.2	15.9	31.9
DRGD [14]	36.3	17.6	33.6
CGU [16]	36.3	18.0	33.8
Seq2seq (our impl.)	33.5	16.8	31.4
VND (our model)	**39.0**	**19.4**	**35.3**

attention to the potential semantic relationship between the source text and the target sentence. The Seq2seq model cannot obtain more semantics, so it is easy to produce too general phrases and does not necessarily involve the original text.

Table 4. Examples of the generated summaries on the test set of the English Gigawords corpus, compared with that of the Seq2seq model and the reference.

Source(1): jordan’s crown prince hassan ibn talal arrived tuesday for his first visit to jerusalem and was to pay his condolences to the widow of assassinated prime minister yitzhak rabin.
Reference: jordan’s crown prince makes first visit to jerusalem
Seq2seq: jordan’s crown prince arrives in jerusalem
VND: jordan’s crown prince arrives for first visit to jerusalem

Source(2): a consortium led by us investment bank goldman sachs thursday increased its takeover offer of associated british ports holdings the biggest port operator in britain after being threatened with possible rival bid.
Reference: goldman sachs increases bid for ab ports
Seq2seq: goldman sachs ups takeover offer of ab british ports
VND: goldman sachs ups increases offer of ab ports

Source(3): the german government and red cross have decided to give ###,### dollars in humanitarian aid to victims of the earthquake which devastated unk in northwest iran the embassy here announced wednesday.
Reference: germany gives ###,### dollars in aid for iran quake victims
Seq2seq: germany to give ###,### dollars in humanitarian aid to iran
VND: germany to give ###,### dollars in aid to iran quake victims

Source(4): global banking giant hsbc said on monday that its pre tax profits had risen in the third quarter despite loan write offs in the united states rising to #.# billion dollars lrb #.# billion euros rrb.
Reference: hsbc says profits rise despite rising us bad debts
Seq2seq: hsbc says profits up despite us loan write offs
VND: hsbc says profits rise despite us loan write offs

Results on Gigaword Corpus. Table 3 presents the test ROUGE F1 score on the Gigaword corpus. Our **VND** model still outperforms all baseline models and achieves F1 scores of 39.0, 19.4 and 35.3 for ROUGE 1, 2, and L. Comparing with the ABS model, our model performs significantly better by an 9.4 ROUGE-1 F1 score. The ASC+FSC$_1$ and DRGD models has the highest scores, because both of which incorporates the latent variables into the abstractive summarization model, but in different ways. Compared with these two models, our model can still be better, perhaps the design of our model structure allows for a better enhancement of the summarization performance. In CGU model, the author uses a a convolutional gated unit for global encoding to get a high score. In future work, we will try to incorporates this component to our model to improve the quality of the summary.

In addition to ROUGE scores, we also provide several randomly selected examples of generated summaries in Table 4. It can observe that the summaries generated by our model are more coherent and descriptive than those generated by the seq2seq model. We suspect that the main reason for this phenomenon lies in variational neural decoder , the seq2seq model only concerns about the hidden semantics relation behind the source text and target sentence, so it tends to generate phrases which are too general to necessarily refer to the source text.

In the experiment, we train and test our model performance on Chinese and English datasets, and the results on both datasets are better than the baseline model. This proves that our model has good generality in different languages, and also proves that our model can indeed capture complex semantics and strong dependencies through a latent variable that explicitly models the underlying semantics of the source texts to improve the quality of the generated summary.

6. Conclusion

This paper focuses on the problem of redundant and incoherent abstract information in the current field of text summary generation. We propose a new text summarization model based on variational neural decoder. This model combines the advantages of the variational recurrent neural network and the variational decoder to learn complex semantics, and uses a double-layer recurrent network to enhance the strong dependency information between the time before and after the summary output to improve the quality of the summary generation. We evaluate our proposed model on the LCSTS and English Gigaword datasets. The experimental results show that our model outperforms state-of-the-art models and prove that our model captures more strong and complex dependencies to ensure that the generated summary has higher quality.

Acknowledgments. This work was supported by the National Key R&D Program of China No. 2018YFC0831800 and the National Science Foundation of China under Grant 61772188.

References

1. Allahyari, M., Pouriyeh, S.A., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B., Kochut, K.: Text summarization techniques: A brief survey. CoRR abs/1707.02268 (2017)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR 2015 (2015)

3. Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H.: Distraction-based neural networks for document summarization. CoRR abs/1610.08462 (2016)
4. Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers (2016)
5. Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A.C., Bengio, Y.: A recurrent latent variable model for sequential data. In: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada. pp. 2980–2988 (2015)
6. Gu, J., Lu, Z., Li, H., Li, V.O.K.: Incorporating copying mechanism in sequence-to-sequence learning. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers (2016)
7. Hao, X., Cao, Y., Shang, Y., Liu, Y., Tan, J., Li, G.: Adversarial reinforcement learning for chinese text summarization. In: International Conference on Computational Science (2018)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735–1780 (1997)
9. Hu, B., Chen, Q., Zhu, F.: Lcsts: A large scale chinese short text summarization dataset. Computer Science pp. 2667–2671 (2015)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR abs/1412.6980 (2014)
11. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. CoRR abs/1312.6114 (2013)
12. Kouris, P., Alexandridis, G., Stafylopatis, A.: Abstractive text summarization based on deep learning and semantic content generalization. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers. pp. 5082–5092 (2019)
13. Li, J., Zhang, C., Chen, X., Cao, Y., Liao, P., Zhang, P.: Abstractive text summarization with multi-head attention. pp. 1–8 (07 2019)
14. Li, P., Lam, W., Bing, L., Wang, Z.: Deep recurrent generative decoder for abstractive text summarization. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017. pp. 2091–2100 (2017)
15. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out (2004)
16. Lin, J., Sun, X., Ma, S., Su, Q.: Global encoding for abstractive summarization. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers. pp. 163–169 (2018)
17. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015. pp. 1412–1421 (2015)
18. Ma, S., Sun, X., Xu, J., Wang, H., Li, W., Su, Q.: Improving semantic relevance for sequence-to-sequence learning of chinese social media text summarization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers. pp. 635–640 (2017)
19. Mehta, P., Majumder, P.: From Extractive to Abstractive Summarization: A Journey. Springer (2019)
20. Miao, Y., Blunsom, P.: Language as a latent variable: Discrete generative models for sentence compression. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016. pp. 319–328 (2016)
21. Nallapati, R., Zhou, B., dos Santos, C.N., Gülçehre, Ç., Xiang, B.: Abstractive text summarization using sequence-to-sequence rnns and beyond. In: Proceedings of the 20th SIGNLL

Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016. pp. 280–290 (2016)

22. Napoles, C., Gormley, M.R., Durme, B.V.: Annotated gigaword. In: Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction, AKBC-WEKEX@NAACL-HLT 2012, Montrèal, Canada, June 7-8, 2012. pp. 95–100 (2012)
23. Neto, J.L., Freitas, A.A., Kaestner, C.A.A.: Automatic text summarization using a machine learning approach. In: Advances in Artificial Intelligence, 16th Brazilian Symposium on Artificial Intelligence, SBIA 2002, Porto de Galinhas/Recife, Brazil, November 11-14, 2002, Proceedings. pp. 205–215 (2002)
24. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014. pp. 1278–1286 (2014)
25. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015. pp. 379–389 (2015)
26. Schumann, R.: Unsupervised abstractive sentence summarization using length controlled variational autoencoder. CoRR abs/1809.05233 (2018)
27. Song, S., Huang, H., Ruan, T.: Abstractive text summarization using LSTM-CNN based deep learning. Multimedia Tools Appl. 78(1), 857–875 (2019)
28. Su, J., Wu, S., Xiong, D., Lu, Y., Han, X., Zhang, B.: Variational recurrent neural machine translation. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018. pp. 5488–5495 (2018)
29. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. pp. 3104–3112 (2014)
30. Verma, R., Lee, D.: Extractive summarization: Limits, compression, generalized model and heuristics. Computacion Y Sistemas 21(4) (2017)
31. Zhang, B., Xiong, D., Su, J., Duan, H., Zhang, M.: Variational neural machine translation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016. pp. 521–530 (2016)
32. Zhao, H.: A retrieval algorithm of encrypted speech based on syllable-level perceptual hashing. Computer Science and Information Systems 14, 24–24 (2017)
33. Zhao, H., Wang, G., Xu, C., Yu, F.: Voice activity detection method based on multivalued coarse-graining lempel-ziv complexity. Computer Science and Information Systems. 8(3), 869–888 (2011)

Huan Zhao is a Professor at the College of Computer Science and Electronic Engineering, Hunan University. She obtained her B. Sc degree and M.S. degree in Computer Application Technology from Hunan University in 1989 and 2004, respectively, and completed her Ph.D. in Computer Science and Technology at the same school in 2010.

Her current research interests include speech information processing, embedded speech recognition and machine learning.

Jie Cao received his bachelor's degree in information and computing Science from Shenyang Ligong University, Shenyang, China, and his master's degree in computer technology from Hunan University, Changsha, China, in 2017.

His current research interests include natural language processing, abstractive summarization, dialog systems, machine learning, and deep learning.

Mingquan Xu received his bachelor's degree in network engineering from Wuhan Textile University, Shenyang, China, and his master's degree in computer technology from Hunan University, Changsha, China, in 2017.

His current research interests include natural language processing, named entity recognition, machine learning, and deep learning.

Jian Lu received his bachelor's degree in information and computing Science from Dalian University, Shenyang, China, and his master's degree in computer technology from Hunan University, Changsha, China, in 2017.

His current research interests include natural language processing, named entity recognition, dialog systems, machine learning, and deep learning.

Received: January 31, 2020; Accepted: June 1, 2020.