Deep Recurrent Generative Decoder for Abstractive Text Summarization[*]

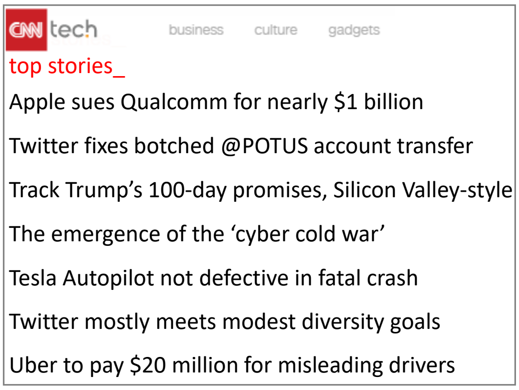
Piji Li[†] **Wai Lam**[†] **Lidong Bing**[‡] **Zihao Wang**[†]
[†]Key Laboratory on High Confidence Software Technologies (Sub-Lab, CUHK),
Ministry of Education, China
[†]Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong
[‡]AI Lab, Tencent Inc., Shenzhen, China
[†]{pjli, wlam, zhwang}@se.cuhk.edu.hk, [‡]lyndonbing@tencent.com

Abstract

We propose a new framework for abstractive text summarization based on a sequence-to-sequence oriented encoder-decoder model equipped with a deep recurrent generative decoder (DRGN). Latent structure information implied in the target summaries is learned based on a recurrent latent random model for improving the summarization quality. Neural variational inference is employed to address the intractable posterior inference for the recurrent latent variables. Abstractive summaries are generated based on both the generative latent variables and the discriminative deterministic states. Extensive experiments on some benchmark datasets in different languages show that DRGN achieves improvements over the state-of-the-art methods.

1 Introduction

Automatic summarization is the process of automatically generating a summary that retains the most important content of the original text document (Edmundson, 1969; Luhn, 1958; Nenkova and McKeown, 2012). Different from the common extraction-based and compression-based methods, abstraction-based methods aim at constructing new sentences as summaries, thus they require a deeper understanding of the text and the capability of generating new sentences, which provide an obvious advantage in improving the focus of a summary, reducing the redundancy, and keeping a good compression rate (Bing et al., 2015; Rush et al., 2015; Nallapati et al., 2016).



top stories_

Apple sues Qualcomm for nearly $1 billion

Twitter fixes botched @POTUS account transfer

Track Trump's 100-day promises, Silicon Valley-style

The emergence of the 'cyber cold war'

Tesla Autopilot not defective in fatal crash

Twitter mostly meets modest diversity goals

Uber to pay $20 million for misleading drivers

Figure 1: Headlines of the top stories from the channel "Technology" of CNN.

Some previous research works show that human-written summaries are more abstractive (Jing and McKeown, 2000). Moreover, our investigation reveals that people may naturally follow some inherent structures when they write the abstractive summaries. To illustrate this observation, we show some examples in Figure 1, which are some top story summaries or headlines from the channel "Technology" of CNN. After analyzing the summaries carefully, we can find some common structures from them, such as "**What**", "**What-Happened**" , "**Who Action What**", etc. For example, the summary "*Apple sues Qualcomm for nearly $1 billion*" can be structuralized as "Who (Apple) Action (sues) What (Qualcomm)". Similarly, the summaries "*[Twitter] [fixes] [botched @POTUS account transfer]*", "*[Uber] [to pay] [$20 million] for misleading drivers*", and "*[Bipartisan bill] aims to [reform] [H-1B visa system]*" also follow the structure of "Who Action What". The summary "*The emergence of the 'cyber cold war'*" matches with the structure of "What", and the summary "*St. Louis' public library computers hacked*" follows

[*] The work described in this paper is supported by a grant from the Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14203414).

the structure of “What-Happened”.

Intuitively, if we can incorporate the latent structure information of summaries into the abstractive summarization model, it will improve the quality of the generated summaries. However, very few existing works specifically consider the latent structure information of summaries in their summarization models. Although a very popular neural network based sequence-to-sequence (seq2seq) framework has been proposed to tackle the abstractive summarization problem (Lopyrev, 2015; Rush et al., 2015; Nallapati et al., 2016), the calculation of the internal decoding states is entirely deterministic. The deterministic transformations in these discriminative models lead to limitations on the representation ability of the latent structure information. Miao and Blunsom (2016) extended the seq2seq framework and proposed a generative model to capture the latent summary information, but they did not consider the recurrent dependencies in their generative model leading to limited representation ability.

To tackle the above mentioned problems, we design a new framework based on sequence-to-sequence oriented encoder-decoder model equipped with a latent structure modeling component. We employ Variational Auto-Encoders (VAEs) (Kingma and Welling, 2013; Rezende et al., 2014) as the base model for our generative framework which can handle the inference problem associated with complex generative modeling. However, the standard framework of VAEs is not designed for sequence modeling related tasks. Inspired by (Chung et al., 2015), we add historical dependencies on the latent variables of VAEs and propose a deep recurrent generative decoder (DRGD) for latent structure modeling. Then the standard discriminative deterministic decoder and the recurrent generative decoder are integrated into a unified decoding framework. The target summaries will be decoded based on both the discriminative deterministic variables and the generative latent structural information. All the neural parameters are learned by back-propagation in an end-to-end training paradigm.

The main contributions of our framework are summarized as follows: (1) We propose a sequence-to-sequence oriented encoder-decoder model equipped with a deep recurrent generative decoder (DRGD) to model and learn the latent structure information implied in the target summaries of the training data. Neural variational inference is employed to address the intractable posterior inference for the recurrent latent variables. (2) Both the generative latent structural information and the discriminative deterministic variables are jointly considered in the generation process of the abstractive summaries. (3) Experimental results on some benchmark datasets in different languages show that our framework achieves better performance than the state-of-the-art models.

2 Related Works

Automatic summarization is the process of automatically generating a summary that retains the most important content of the original text document (Nenkova and McKeown, 2012). Traditionally, the summarization methods can be classified into three categories: extraction-based methods (Erkan and Radev, 2004; Goldstein et al., 2000; Wan et al., 2007; Min et al., 2012; Nallapati et al., 2017; Cheng and Lapata, 2016; Cao et al., 2016; Song et al., 2017), compression-based methods (Li et al., 2013; Wang et al., 2013; Li et al., 2015, 2017), and abstraction-based methods. In fact, previous investigations show that human-written summaries are more abstractive (Barzilay and McKeown, 2005; Bing et al., 2015). Abstraction-based approaches can generate new sentences based on the facts from different source sentences. Barzilay and McKeown (2005) employed sentence fusion to generate a new sentence. Bing et al. (2015) proposed a more fine-grained fusion framework, where new sentences are generated by selecting and merging salient phrases. These methods can be regarded as a kind of indirect abstractive summarization, and complicated constraints are used to guarantee the linguistic quality.

Recently, some researchers employ neural network based framework to tackle the abstractive summarization problem. Rush et al. (2015) proposed a neural network based model with local attention modeling, which is trained on the Gigaword corpus, but combined with an additional log-linear extractive summarization model with handcrafted features. Gu et al. (2016) integrated a copying mechanism into a seq2seq framework to improve the quality of the generated summaries. Chen et al. (2016) proposed a new attention mechanism that not only considers the important source segments, but also distracts them in the decoding step in order to better grasp the overall meaning of input documents. Nallapati et al. (2016) utilized a trick to control the vocabulary size to improve the training efficiency. The calculations in these methods are all deterministic and the representation ability is limited. Miao and Blunsom (2016) extended the seq2seq framework and proposed a generative model to capture the latent summary information, but they do not consider the recurrent dependencies in their generative model leading to limited representation ability.

Some research works employ topic models to capture the latent information from source documents or sentences. Wang et al. (2009) proposed a new Bayesian sentence-based topic model by making use of both the term-document and term-sentence associations to improve the performance of sentence selection. Celikyilmaz and Hakkani-Tur (2010) estimated scores for sentences based on their latent characteristics using a hierarchical topic model, and trained a regression model to extract sentences. However, they only use the latent topic information to conduct the sentence salience estimation for extractive summarization. In contrast, our purpose is to model and learn the latent structure information from the target summaries and use it to enhance the performance of abstractive summarization.

3 Framework Description

3.1 Overview

As shown in Figure 2, the basic framework of our approach is a neural network based encoder-decoder framework for sequence-to-sequence learning. The input is a variable-length sequence $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ representing the source text. The word embedding $\mathbf{x}_t$ is initialized randomly and learned during the optimization process. The output is also a sequence $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$, which represents the generated abstractive summaries. Gated Recurrent Unit (GRU) (Cho et al., 2014) is employed as the basic sequence modeling component for the encoder and the decoder. For latent structure modeling, we add historical dependencies on the latent variables of Variational Auto-Encoders (VAEs) and propose a deep recurrent generative decoder (DRGD) to distill the complex latent structures implied in the target summaries of the training data. Finally, the abstractive summaries will be decoded out based on both the discriminative deterministic variables H and the generative latent structural information Z.

3.2 Recurrent Generative Decoder

Assume that we have obtained the source text representation $\mathbf{h}^e \in \mathbb{R}^{k_h}$. The purpose of the decoder is to translate this source code $\mathbf{h}^e$ into a series of hidden states $\{\mathbf{h}^d_1, \mathbf{h}^d_2, \dots, \mathbf{h}^d_n\}$, and then revert these hidden states to an actual word sequence and generate the summary.

For standard recurrent decoders, at each time step t, the hidden state $\mathbf{h}^d_t \in \mathbb{R}^{k_h}$ is calculated using the dependent input symbol $\mathbf{y}_{t-1} \in \mathbb{R}^{k_w}$ and the previous hidden state $\mathbf{h}^d_{t-1}$:

$$\mathbf{h}^d_t = f(\mathbf{y}_{t-1}, \mathbf{h}^d_{t-1}) \tag{1}$$

where $f(\cdot)$ is a recurrent neural network such as vanilla RNN, Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), and Gated Recurrent Unit (GRU) (Cho et al., 2014). No matter which one we use for $f(\cdot)$, the common transformation operation is as follows:

$$\mathbf{h}^d_t = g(\mathbf{W}^d_{yh}\mathbf{y}_{t-1} + \mathbf{W}^d_{hh}\mathbf{h}^d_{t-1} + \mathbf{b}^d_h) \tag{2}$$

where $\mathbf{W}^d_{yh} \in \mathbb{R}^{k_h \times k_w}$ and $\mathbf{W}^d_{hh} \in \mathbb{R}^{k_h \times k_h}$ are the linear transformation matrices. $\mathbf{b}^d_h$ is the bias. k_h is the dimension of the hidden layers, and k_w is the dimension of the word embeddings. $g(\cdot)$ is the non-linear activation function. From Equation 2, we can see that all the transformations are deterministic, which leads to a deterministic recurrent hidden state h^d_t. From our investigations, we find that the representational power of such deterministic variables are limited. Some more complex latent structures in the target summaries, such as the high-level syntactic features and latent topics, cannot be modeled effectively by the deterministic operations and variables.

Recently, a generative model called Variational Auto-Encoders (VAEs) (Kingma and Welling, 2013; Rezende et al., 2014) shows strong capability in modeling latent random variables and improves the performance of tasks in different fields such as sentence generation (Bowman et al., 2016) and image generation (Gregor et al., 2015). However, the standard VAEs is not designed for modeling sequence directly. Inspired by (Chung et al., 2015), we extend the standard VAEs by

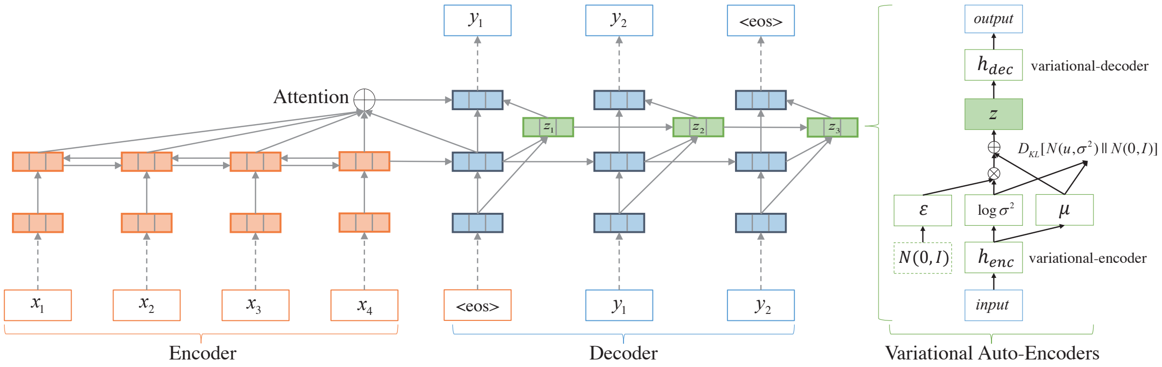


Figure 2: Our deep recurrent generative decoder (DRGD) for latent structure modeling.

introducing the historical latent variable dependencies to make it be capable of modeling sequence data. Our proposed latent structure modeling framework can be viewed as a sequence generative model which can be divided into two parts: inference (variational-encoder) and generation (variational-decoder). As shown in the decoder component of Figure 2, the input of the original VAEs only contains the observed variable $\mathbf{y}_t$, and the variational-encoder can map it to a latent variable $\mathbf{z} \in \mathbb{R}^{k_z}$, which can be used to reconstruct the original input. For the task of summarization, in the sequence decoder component, the previous latent structure information needs to be considered for constructing more effective representations for the generation of the next state.

For the inference stage, the variational-encoder can map the observed variable $\mathbf{y}_{<t}$ and the previous latent structure information $\mathbf{z}_{<t}$ to the posterior probability distribution of the latent structure variable $p_\theta(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})$. It is obvious that this is a recurrent inference process in which $\mathbf{z}_t$ contains the historical dynamic latent structure information. Compared with the variational inference process $p_\theta(\mathbf{z}_t|\mathbf{y}_t)$ of the typical VAEs model, the recurrent framework can extract more complex and effective latent structure features implied in the sequence data.

For the generation process, based on the latent structure variable $\mathbf{z}_t$, the target word y_t at the time step t is drawn from a conditional probability distribution $p_\theta(\mathbf{y}_t|\mathbf{z}_t)$. The target is to maximize the probability of each generated summary $y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T\}$ based on the generation process according to:

$$p_\theta(y) = \prod_{t=1}^{T} \int p_\theta(\mathbf{y}_t|\mathbf{z}_t) p_\theta(\mathbf{z}_t) d\mathbf{z}_t \tag{3}$$

For the purpose of solving the intractable integral of the marginal likelihood as shown in Equation 3, a recognition model $q_\phi(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})$ is introduced as an approximation to the intractable true posterior $p_\theta(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})$. The recognition model parameters ϕ and the generative model parameters θ can be learned jointly. The aim is to reduce the Kulllback-Leibler divergence (KL) between $q_\phi(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})$ and $p_\theta(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})$:

$$\begin{aligned} & D_{KL}[q_\phi(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})\|p_\theta(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})] \\ &= \int_z q_\phi(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t}) \log \frac{q_\phi(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})}{p_\theta(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})} dz \\ &= \mathbb{E}_{q_\phi(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})}[\log q_\phi(\mathbf{z}_t|\cdot) - \log p_\theta(\mathbf{z}_t|\cdot)] \end{aligned}$$

where $\cdot$ denotes the conditional variables $\mathbf{y}_{<t}$ and $\mathbf{z}_{<t}$. Bayes rule is applied to $p_\theta(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})$, and we can extract $\log p_\theta(\mathbf{z})$ from the expectation, transfer the expectation term $\mathbb{E}_{q_\phi(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})}$ back to KL-divergence, and rearrange all the terms. Consequently the following holds:

$$\begin{aligned} & \log p_\theta(\mathbf{y}_{<t}) = \\ & D_{KL}[q_\phi(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})\|p_\theta(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})] \\ & + \mathbb{E}_{q_\phi(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})}[\log p_\theta(\mathbf{y}_{<t}|\mathbf{z}_t)] \\ & - D_{KL}[q_\phi(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})\|p_\theta(\mathbf{z}_t)] \end{aligned} \tag{4}$$

Let $\mathcal{L}(\theta, \phi; y)$ represent the last two terms from the right part of Equation 4:

$$\begin{aligned} & \mathcal{L}(\theta, \varphi; y) = \\ & \quad \mathbb{E}_{q_\phi(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})}\Big\{\sum\nolimits_{t=1}^{T} \log p_\theta(\mathbf{y}_t|\mathbf{z}_t) \\ & \quad - D_{KL}[q_\phi(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})\|p_\theta(\mathbf{z}_t)]\Big\} \end{aligned} \tag{5}$$

Since the first KL-divergence term of Equation 4 is non-negative, we have $\log p_\theta(\mathbf{y}_{<t}) \geq \mathcal{L}(\theta, \phi; y)$ meaning that $\mathcal{L}(\theta, \phi; y)$ is a lower bound (the objective to be maximized) on the marginal likelihood. In order to differentiate and optimize the

lower bound $\mathcal{L}(\theta, \phi; y)$, following the core idea of VAEs, we use a neural network framework for the probabilistic encoder $q_\phi(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t})$ for better approximation.

3.3 Abstractive Summary Generation

We also design a neural network based framework to conduct the variational inference and generation for the recurrent generative decoder component similar to some design in previous works (Kingma and Welling, 2013; Rezende et al., 2014; Gregor et al., 2015). The encoder component and the decoder component are integrated into a unified abstractive summarization framework. Considering that GRU has comparable performance but with less parameters and more efficient computation, we employ GRU as the basic recurrent model which updates the variables according to the following operations:

$$\begin{aligned}
\mathbf{r}_t &= \sigma(\mathbf{W}_{xr}\mathbf{x}_t + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{b}_r) \\
\mathbf{z}_t &= \sigma(\mathbf{W}_{xz}\mathbf{x}_t + \mathbf{W}_{hz}\mathbf{h}_{t-1} + \mathbf{b}_z) \\
\mathbf{g}_t &= \tanh(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \\
\mathbf{h}_t &= \mathbf{z}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \odot \mathbf{g}_t
\end{aligned}$$

where $\mathbf{r}_t$ is the reset gate, $\mathbf{z}_t$ is the update gate. $\odot$ denotes the element-wise multiplication. $tanh$ is the hyperbolic tangent activation function.

As shown in the left block of Figure 2, the encoder is designed based on bidirectional recurrent neural networks. Let $\mathbf{x}_t$ be the word embedding vector of the t-th word in the source sequence. GRU maps $\mathbf{x}_t$ and the previous hidden state $\mathbf{h}_{t-1}$ to the current hidden state $\mathbf{h}_t$ in feed-forward direction and back-forward direction respectively:

$$\begin{aligned}
\overrightarrow{\mathbf{h}}_t &= GRU(x_t, \overrightarrow{\mathbf{h}}_{t-1}) \\
\overleftarrow{\mathbf{h}}_t &= GRU(x_t, \overleftarrow{\mathbf{h}}_{t-1})
\end{aligned} \qquad (6)$$

Then the final hidden state $\mathbf{h}^e_t \in \mathbb{R}^{2k_h}$ is concatenated using the hidden states from the two directions: $\mathbf{h}^e_t = \overrightarrow{\mathbf{h}}_t || \overleftarrow{\mathbf{h}}$. As shown in the middle block of Figure 2, the decoder consists of two components: discriminative deterministic decoding and generative latent structure modeling.

The discriminative deterministic decoding is an improved attention modeling based recurrent sequence decoder. The first hidden state $\mathbf{h}^d_1$ is initialized using the average of all the source input states: $\mathbf{h}^d_1 = \frac{1}{T^e}\sum_{t=1}^{T^e}\mathbf{h}^e_t$, where $\mathbf{h}^e_t$ is the source input hidden state. T^e is the input sequence length. The deterministic decoder hidden state $\mathbf{h}^d_t$ is calculated using two layers of GRUs. On the first layer, the hidden state is calculated only using the current input word embedding $\mathbf{y}_{t-1}$ and the previous hidden state $\mathbf{h}^{d_1}_{t-1}$:

$$\mathbf{h}^{d_1}_t = GRU_1(\mathbf{y}_{t-1}, \mathbf{h}^{d_1}_{t-1}) \qquad (7)$$

where the superscript d_1 denotes the first decoder GRU layer. Then the attention weights at the time step t are calculated based on the relationship of $\mathbf{h}^{d_1}_t$ and all the source hidden states $\{\mathbf{h}^e_t\}$. Let $a_{i,j}$ be the attention weight between $\mathbf{h}^{d_1}_i$ and $\mathbf{h}^e_j$, which can be calculated using the following formulation:

$$\begin{aligned}
a_{i,j} &= \frac{\exp(e_{i,j})}{\sum_{j'=1}^{T^e}\exp(e_{i,j'})} \\
e_{i,j} &= \mathbf{v}^T \tanh(\mathbf{W}^d_{hh}\mathbf{h}^{d_1}_i + \mathbf{W}^e_{hh}\mathbf{h}^e_j + \mathbf{b}_a)
\end{aligned}$$

where $\mathbf{W}^d_{hh} \in \mathbb{R}^{k_h \times k_h}$, $\mathbf{W}^e_{hh} \in \mathbb{R}^{k_h \times 2k_h}$, $\mathbf{b}_a \in \mathbb{R}^{k_h}$, and $\mathbf{v} \in \mathbb{R}^{k_h}$. The attention context is obtained by the weighted linear combination of all the source hidden states:

$$\mathbf{c}_t = \sum_{j'=1}^{T^e} a_{t,j'}\mathbf{h}^e_{j'} \qquad (8)$$

The final deterministic hidden state $\mathbf{h}^{d_2}_t$ is the output of the second decoder GRU layer, jointly considering the word $\mathbf{y}_{t-1}$, the previous hidden state $\mathbf{h}^{d_2}_{t-1}$, and the attention context $\mathbf{c}_t$:

$$\mathbf{h}^{d_2}_t = GRU_2(\mathbf{y}_{t-1}, \mathbf{h}^{d_2}_{t-1}, \mathbf{c}_t) \qquad (9)$$

For the component of recurrent generative model, inspired by some ideas in previous works (Kingma and Welling, 2013; Rezende et al., 2014; Gregor et al., 2015), we assume that both the prior and posterior of the latent variables are Gaussian, i.e., $p_\theta(\mathbf{z}_t) = \mathcal{N}(0, \mathbf{I})$ and $q_\phi(\mathbf{z}_t|\mathbf{y}_{<t}, \mathbf{z}_{<t}) = \mathcal{N}(\mathbf{z}_t; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ denote the variational mean and standard deviation respectively, which can be calculated via a multilayer perceptron. Precisely, given the word embedding $\mathbf{y}_{t-1}$, the previous latent structure variable $\mathbf{z}_{t-1}$, and the previous deterministic hidden state $\mathbf{h}^d_{t-1}$, we first project it to a new hidden space:

$$\mathbf{h}^{e_z}_t = g(\mathbf{W}^{e_z}_{yh}\mathbf{y}_{t-1} + \mathbf{W}^{e_z}_{zh}\mathbf{z}_{t-1} + \mathbf{W}^{e_z}_{hh}\mathbf{h}^d_{t-1} + \mathbf{b}^{e_z}_h)$$

where $\mathbf{W}^{e_z}_{yh} \in \mathbb{R}^{k_h \times k_w}$, $\mathbf{W}^{e_z}_{zh} \in \mathbb{R}^{k_h \times k_z}$, $\mathbf{W}^{e_z}_{hh} \in \mathbb{R}^{k_h \times k_h}$, and $\mathbf{b}^{e_z}_h \in \mathbb{R}^{k_h}$. g is the sigmoid activation function: $\sigma(\mathbf{x}) = 1/(1 + e^{-\mathbf{x}})$. Then the

Gaussian parameters $\boldsymbol{\mu}_t \in \mathbb{R}^{k_z}$ and $\boldsymbol{\sigma}_t \in \mathbb{R}^{k_z}$ can be obtained via a linear transformation based on $\mathbf{h}_t^{e_z}$:

$$\begin{aligned} \boldsymbol{\mu}_t &= \mathbf{W}_{h\mu}^{e_z}\mathbf{h}_t^{e_z} + \mathbf{b}_{\mu}^{e_z} \\ \log(\boldsymbol{\sigma}_t^2) &= \mathbf{W}_{h\sigma}\mathbf{h}_t^{e_z} + \mathbf{b}_{\sigma}^{e_z} \end{aligned} \tag{10}$$

The latent structure variable $\mathbf{z}_t \in \mathbb{R}^{k_z}$ can be calculated using the reparameterization trick:

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \;\; \mathbf{z}_t = \boldsymbol{\mu}_t + \boldsymbol{\sigma}_t \otimes \varepsilon \tag{11}$$

where $\boldsymbol{\varepsilon} \in \mathbb{R}^{k_z}$ is an auxiliary noise variable. The process of inference for finding $\mathbf{z}_t$ based on neural networks can be teated as a variational encoding process.

To generate summaries precisely, we first integrate the recurrent generative decoding component with the discriminative deterministic decoding component, and map the latent structure variable $\mathbf{z}_t$ and the deterministic decoding hidden state $\mathbf{h}_t^{d_2}$ to a new hidden variable:

$$\mathbf{h}_t^{d_y} = tanh(\mathbf{W}_{zh}^{d_y}\mathbf{z}_t + \mathbf{W}_{hh}^{d_z}\mathbf{h}_t^{d_2} + \mathbf{b}_h^{d_y}) \tag{12}$$

Given the combined decoding state $\mathbf{h}_t^{d_y}$ at the time t, the probability of generating any target word y_t is given as follows:

$$\mathbf{y}_t = \varsigma(\mathbf{W}_{hy}^{d}\mathbf{h}_t^{d_y} + \mathbf{b}_{hy}^{d}) \tag{13}$$

where $\mathbf{W}_{hy}^d \in \mathbb{R}^{k_y \times k_h}$ and $\mathbf{b}_{hy}^d \in \mathbb{R}^{k_y}$. $\varsigma(\cdot)$ is the softmax function. Finally, we use a beam search algorithm (Koehn, 2004) for decoding and generating the best summary.

3.4 Learning

Although the proposed model contains a recurrent generative decoder, the whole framework is fully differentiable. As shown in Section 3.3, both the recurrent deterministic decoder and the recurrent generative decoder are designed based on neural networks. Therefore, all the parameters in our model can be optimized in an end-to-end paradigm using back-propagation. We use $\{X\}_N$ and $\{Y\}_N$ to denote the training source and target sequence. Generally, the objective of our framework consists of two terms. One term is the negative log-likelihood of the generated summaries, and the other one is the variational lower bound $\mathcal{L}(\theta, \phi; Y)$ mentioned in Equation 5. Since the variational lower bound $\mathcal{L}(\theta, \phi; Y)$ also contains a likelihood term, we can merge it with the likelihood term of summaries. The final objective function, which needs to be minimized, is formulated as follows:

$$\begin{aligned} \mathcal{J} = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}\Big\{ & -\log\Big[p(y_t^{(n)}|y_{<t}^{(n)}, X^{(n)})\Big] \\ & + D_{KL}\Big[q_\phi(\mathbf{z}_t^{(n)}|\mathbf{y}_{<t}^{(n)}, \mathbf{z}_{<t}^{(n)}) \| p_\theta(\mathbf{z}_t^{(n)})\Big]\Big\} \end{aligned} \tag{14}$$

4 Experimental Setup

4.1 Datesets

We train and evaluate our framework on three popular datasets. **Gigawords** is an English sentence summarization dataset prepared based on Annotated Gigawords[1] by extracting the first sentence from articles with the headline to form a source-summary pair. We directly download the prepared dataset used in (Rush et al., 2015). It roughly contains 3.8M training pairs, 190K validation pairs, and 2,000 test pairs. **DUC-2004**[2] is another English dataset only used for testing in our experiments. It contains 500 documents. Each document contains 4 model summaries written by experts. The length of the summary is limited to 75 bytes. **LCSTS** is a large-scale Chinese short text summarization dataset, consisting of pairs of (short text, summary) collected from Sina Weibo[3] (Hu et al., 2015). We take Part-I as the training set, Part-II as the development set, and Part-III as the test set. There is a score in range $1 \sim 5$ labeled by human to indicate how relevant an article and its summary is. We only reserve those pairs with scores no less than 3. The size of the three sets are 2.4M, 8.7k, and 725 respectively. In our experiments, we only take Chinese character sequence as input, without performing word segmentation.

4.2 Evaluation Metrics

We use ROUGE score (Lin, 2004) as our evaluation metric with standard options. The basic idea of ROUGE is to count the number of overlapping units between generated summaries and the reference summaries, such as overlapped n-grams, word sequences, and word pairs. F-measures of ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L) and ROUGE-SU4 (R-SU4) are reported.

4.3 Comparative Methods

We compare our model with some baselines and state-of-the-art methods. Because the datasets are

[1] https://catalog.ldc.upenn.edu/ldc2012t21

[2] http://duc.nist.gov/duc2004

[3] http://www.weibo.com

quite standard, so we just extract the results from their papers. Therefore the baseline methods on different datasets may be slightly different.

- **TOPIARY** (Zajic et al., 2004) is the best on DUC2004 Task-1 for compressive text summarization. It combines a system using linguistic based transformations and an unsupervised topic detection algorithm for compressive text summarization.

- **MOSES+** (Rush et al., 2015) uses a phrase-based statistical machine translation system trained on Gigaword to produce summaries. It also augments the phrase table with "deletion" rulesto improve the baseline performance, and MERT is also used to improve the quality of generated summaries.

- **ABS** and **ABS+** (Rush et al., 2015) are both the neural network based models with local attention modeling for abstractive sentence summarization. ABS+ is trained on the Gigaword corpus, but combined with an additional log-linear extractive summarization model with handcrafted features.

- **RNN** and **RNN-context** (Hu et al., 2015) are two seq2seq architectures. RNN-context integrates attention mechanism to model the context.

- **CopyNet** (Gu et al., 2016) integrates a copying mechanism into the sequence-to-sequence framework.

- **RNN-distract** (Chen et al., 2016) uses a new attention mechanism by distracting the historical attention in the decoding steps.

- **RAS-LSTM** and **RAS-Elman** (Chopra et al., 2016) both consider words and word positions as input and use convolutional encoders to handle the source information. For the attention based sequence decoding process, RAS-Elman selects Elman RNN (Elman, 1990) as decoder, and RAS-LSTM selects Long Short-Term Memory architecture (Hochreiter and Schmidhuber, 1997).

- **LenEmb** (Kikuchi et al., 2016) uses a mechanism to control the summary length by considering the length embedding vector as the input.

- **ASC+FSC$_1$** (Miao and Blunsom, 2016) uses a generative model with attention mechanism to conduct the sentence compression problem. The model first draws a latent summary sentence from a background language model, and then subsequently draws the observed sentence conditioned on this latent summary.

- **lvt2k-1sent** and **lvt5k-1sent** (Nallapati et al., 2016) utilize a trick to control the vocabulary size to improve the training efficiency.

4.4 Experimental Settings

For the experiments on the English dataset Gigawords, we set the dimension of word embeddings to 300, and the dimension of hidden states and latent variables to 500. The maximum length of documents and summaries is 100 and 50 respectively. The batch size of mini-batch training is 256. For DUC-2004, the maximum length of summaries is 75 bytes. For the dataset of LCSTS, the dimension of word embeddings is 350. We also set the dimension of hidden states and latent variables to 500. The maximum length of documents and summaries is 120 and 25 respectively, and the batch size is also 256. The beam size of the decoder was set to be 10. Adadelta (Schmidhuber, 2015) with hyperparameter $\rho = 0.95$ and $\epsilon = 1e - 6$ is used for gradient based optimization. Our neural network based framework is implemented using Theano (Theano Development Team, 2016).

5 Results and Discussions

5.1 ROUGE Evaluation

Table 1: ROUGE-F1 on validation sets

Dataset	System	R-1	R-2	R-L
GIGA	StanD	32.69	15.29	30.60
	DRGD	**36.25**	**17.61**	**33.55**
LCSTS	StanD	33.88	21.49	31.05
	DRGD	**36.71**	**24.00**	**34.10**

We first depict the performance of our model DRGD by comparing to the standard decoders (StanD) of our own implementation. The comparison results on the validation datasets of Gigawords and LCSTS are shown in Table 1. From the results we can see that our proposed generative decoders DRGD can obtain obvious improvements on abstractive summarization than the standard decoders. Actually, the performance of the standard

Table 2: ROUGE-F1 on Gigawords

System	R-1	R-2	R-L
ABS	29.55	11.32	26.42
ABS+	29.78	11.89	26.97
RAS-LSTM	32.55	14.70	30.03
RAS-Elman	33.78	15.97	31.15
ASC + FSC_1	34.17	15.94	31.92
lvt2k-1sent	32.67	15.59	30.64
lvt5k-1sent	35.30	16.64	32.62
DRGD	**36.27**	**17.57**	**33.62**

Table 3: ROUGE-Recall on DUC2004

System	R-1	R-2	R-L
TOPIARY	25.12	6.46	20.12
MOSES+	26.50	8.13	22.85
ABS	26.55	7.06	22.05
ABS+	28.18	8.49	23.81
RAS-Elman	28.97	8.26	24.06
RAS-LSTM	27.41	7.69	23.06
LenEmb	26.73	8.39	23.88
lvt2k-1sen	28.35	9.46	24.59
lvt5k-1sen	28.61	9.42	25.24
DRGD	**31.79**	**10.75**	**27.48**

Table 4: ROUGE-F1 on LCSTS

System	R-1	R-2	R-L
RNN	21.50	8.90	18.60
RNN-context	29.90	17.40	27.20
CopyNet	34.40	21.60	31.30
RNN-distract	35.20	22.60	32.50
DRGD	**36.99**	**24.15**	**34.21**

decoders is similar with those mentioned popular baseline methods.

The results on the English datasets of Gigawords and DUC-2004 are shown in Table 2 and Table 3 respectively. Our model DRGD achieves the best summarization performance on all the ROUGE metrics. Although ASC+FSC_1 also uses a generative method to model the latent summary variables, the representation ability is limited and it cannot bring in noticeable improvements. It is worth noting that the methods lvt2k-1sent and lvt5k-1sent (Nallapati et al., 2016) utilize linguistic features such as parts-of-speech tags, named-entity tags, and TF and IDF statistics of the words as part of the document representation. In fact, extracting all such features is a time consuming work, especially on large-scale datasets such as Gigawords. lvt2k and lvt5k are not end-to-end style models and are more complicated than our model in practical applications.

The results on the Chinese dataset LCSTS are shown in Table 4. Our model DRGD also achieves the best performance. Although CopyNet employs a copying mechanism to improve the summary quality and RNN-distract considers attention information diversity in their decoders, our model is still better than those two methods demonstrating that the latent structure information learned from target summaries indeed plays a role in abstractive summarization. We also believe that integrating the copying mechanism and coverage diversity in our framework will further improve the summarization performance.

5.2 Summary Case Analysis

In order to analyze the reasons of improving the performance, we compare the generated summaries by DRGD and the standard decoders StanD used in some other works such as (Chopra et al., 2016). The source texts, golden summaries, and the generated summaries are shown in Table 5. From the cases we can observe that DRGD can indeed capture some latent structures which are consistent with the golden summaries. For example, our result for S(1) "*Wuhan wins men's soccer title at Chinese city games*" matches the "Who Action What" structure. However, the standard decoder StanD ignores the latent structures and generates some loose sentences, such as the results for S(1) "*Results of men's volleyball at Chinese city games*" does not catch the main points. The reason is that the recurrent variational auto-encoders used in our framework have better representation ability and can capture more effective and complicated latent structures from the sequence data. Therefore, the summaries generated by DRGD have consistent latent structures with the ground truth, leading to a better ROUGE evaluation.

6 Conclusions

We propose a deep recurrent generative decoder (DRGD) to improve the abstractive summarization performance. The model is a sequence-to-sequence oriented encoder-decoder framework equipped with a latent structure modeling component. Abstractive summaries are generated based on both the latent variables and the deterministic states. Extensive experiments on benchmark

Table 5: Examples of the generated summaries.

S(1): hosts wuhan won the men 's soccer title by beating beijing shunyi #-# here at the #th chinese city games on friday. **Golden**: hosts wuhan wins men 's soccer title at chinese city games. **StanD**: results of men 's volleyball at chinese city games. **DRGD**: **wuhan wins men 's soccer title at chinese city games.**
S(2): UNK and the china meteorological administration tuesday signed an agreement here on long - and short-term cooperation in projects involving meteorological satellites and satellite meteorology. **Golden**: UNK china to cooperate in meteorology. **StanD**: weather forecast for major chinese cities. **DRGD**: **china to cooperate in meteorological satellites.**
S(3): the rand gained ground against the dollar at the opening here wednesday , to #.# to the greenback from #.# at the close tuesday. **Golden**: rand gains ground. **StanD**: rand slightly higher against dollar. **DRGD**: **rand gains ground against dollar.**
S(4): new zealand women are having more children and the country 's birth rate reached its highest level in ## years , statistics new zealand said on wednesday. **Golden**: new zealand birth rate reaches ##-year high. **StanD**: new zealand women are having more children birth rate hits highest level in ## years. **DRGD**: **new zealand 's birth rate hits ##-year high.**

datasets show that DRGD achieves improvements over the state-of-the-art methods.

References

Regina Barzilay and Kathleen R McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca Passonneau. 2015. Abstractive multi-document summarization via phrase selection and merging. In *ACL*, pages 1587–1597.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. *CoNLL*, pages 10–21.

Ziqiang Cao, Wenjie Li, Sujian Li, Furu Wei, and Yanran Li. 2016. Attsum: Joint learning of focusing and summarization with neural attention. *COLING*, pages 547–556.

Asli Celikyilmaz and Dilek Hakkani-Tur. 2010. A hybrid hierarchical model for multi-document summarization. In *ACL*, pages 815–824.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for document summarization. In *IJCAI*, pages 2754–2760.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *ACL*, pages 484–494.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734.

Sumit Chopra, Michael Auli, Alexander M Rush, and SEAS Harvard. 2016. Abstractive sentence summarization with attentive recurrent neural networks. *NAACL-HLT*, pages 93–98.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. 2015. A recurrent latent variable model for sequential data. In *NIPS*, pages 2980–2988.

Harold P Edmundson. 1969. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *NAACL-ANLPWorkshop*, pages 40–48.

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. 2015. Draw: A recurrent neural network for image generation. In *ICML*, pages 1462–1471.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*, pages 1631–1640.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. Lcsts: A large scale chinese short text summarization dataset. In *EMNLP*, pages 1962–1972.

Hongyan Jing and Kathleen R McKeown. 2000. Cut and paste based text summarization. In *NAACL*, pages 178–185.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *EMNLP*, pages 1328–1338.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Conference of the Association for Machine Translation in the Americas*, pages 115–124. Springer.

Chen Li, Fei Liu, Fuliang Weng, and Yang Liu. 2013. Document summarization via guided sentence compression. In *EMNLP*, pages 490–500.

Piji Li, Lidong Bing, Wai Lam, Hang Li, and Yi Liao. 2015. Reader-aware multi-document summarization via sparse coding. In *IJCAI*, pages 1270–1276.

Piji Li, Zihao Wang, Wai Lam, Zhaochun Ren, and Lidong Bing. 2017. Salience estimation via variational auto-encoders for multi-document summarization. In *AAAI*, pages 3497–3503.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8.

Konstantin Lopyrev. 2015. Generating news headlines with recurrent neural networks. *arXiv preprint arXiv:1512.01712*.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *EMNLP*, pages 319–328.

Ziheng Lin Min, Yen Kan Chew, and Lim Tan. 2012. Exploiting category-specific information for multi-document summarization. *COLING*, pages 2903–2108.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*, pages 3075–3081.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In *Mining Text Data*, pages 43–76. Springer.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pages 1278–1286.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*, pages 379–389.

Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

Hongya Song, Zhaochun Ren, Piji Li, Shangsong Liang, Jun Ma, and Maarten de Rijke. 2017. Summarizing answers in non-factoid community question-answering. In *WSDM*, pages 405–414.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Manifold-ranking based topic-focused multi-document summarization. In *IJCAI*, volume 7, pages 2903–2908.

Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. 2009. Multi-document summarization using sentence-based topic models. In *ACL-IJCNLP*, pages 297–300.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2013. A sentence compression based framework to query-focused multi-document summarization. In *ACL*, pages 1384–1394.

David Zajic, Bonnie Dorr, and Richard Schwartz. 2004. Bbn/umd at duc-2004: Topiary. In *HLT-NAACL*, pages 112–119.