Enhancing Textual Representation for Abstractive Summarization: Leveraging Masked Decoder

Ruipeng Jia
1. Institute of Information Engineering
Chinese Academy of Sciences
2. School of Cyber Security
University of Chinese Academy of Sciences
Beijing, China
jiaruipeng@iie.ac.cn

Yannan Cao
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
caoyanan@iie.ac.cn

Fang Fang*
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
fangfang0703@iie.ac.cn

Jinpeng Li
1. Institute of Information Engineering
Chinese Academy of Sciences
2. School of Cyber Security
University of Chinese Academy of Sciences
Beijing, China
lijinpeng@iie.ac.cn

Yanbing Liu
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
liuyanbing@iie.ac.cn

Pengfei Yin
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
yinpengfei@iie.ac.cn

***Abstract*—For existing models of abstractive summarization, the paradigm of autoregressive decoder inherently prefers relying on former tokens and the prediction error will propagate subsequently. To effectively eliminate the errors, we need a way to remodeling dependency during text generation. In this paper, we introduce MDSumma (as shorthand for Masked Decoder for Summarization), which masks partial tokens in decoder, aiming to alleviate the over-reliance on the antecedent. Moreover, with further facilitating the flexibility and diversity of textual representation, we employ a variational autoencoder model, sampling continuous latent variables from the probability distribution to explicitly model underlying semantics of the target summaries. Our architecture gives good balance between encoder contextual representation and decoder prediction, sidestepping the gap between training and inference. Experimental results on three benchmark datasets validate the effectiveness that our proposed method significantly outperforms the existing state-of-the-art approaches both on ROUGE and diversity scores.**

***Index Terms*—text summarization, diversity, VAE, model fusion**

I. Introduction

Abstractive summarization aims to generate summaries that express the central idea of original articles, with an encoder to convert source sequence into continuous space representations, from which the decoder generates target sequence. Existing decoders for abstractive summarization are autoregressive sequence models based on deep neural networks, such as RNNs, Wavenet and Transformer. However, they are trained to predict next token given the previous ground truth one, which is by feeding the generated token back at test time. This process is very brittle because prediction extremely relies on the antecedent sequence and discrepancy comes from the mismatch of different distributions, namely, words drawn from data distribution in training and model distribution in inference. As a result, for Seq2Seq and its offsprings, the prediction error will propagate along the way. And we refer to this discrepancy as exposure bias [1]. To effectively eliminate the errors, we need a way to remodeling dependency during text generation.

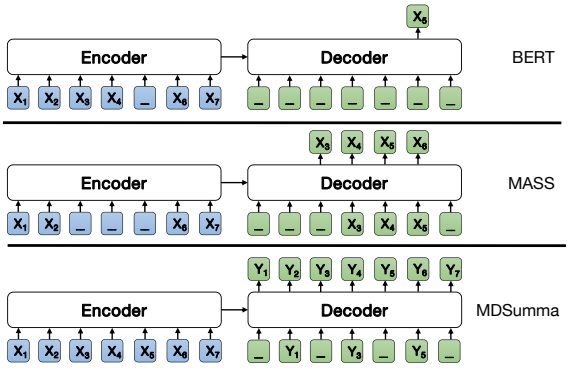

Fig. 1. **Comparison for different architectures**. Encoder and decoder are all Transformer layers.

Masked and pretrained language model is an effective way to alleviate the various task-specific problems, achieving great success in language understanding by transferring knowledge from rich-resource pretraining task to low-resource downstream tasks [2]. BERT [3] combines both word and sentence representations in a single Transformer [4], pretrained

*Corresponding author: Fang Fang

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

on vast amounts of text, with masked language modeling objectives. The mask task randomly masks some tokens and predicts these missing ones from candidate pools. MASS [2] employs masked sequence to sequence pre-training for language generation, which can jointly train the masked encoder and local autoregressive decoder to develop the capability of representation extraction and language modeling, still keeping the exposure bias. As shown in Figure 1, both of the BERT and MASS are self-supervised with masked encoder to learn from the intrinsic structure of raw data. In this paper, we present masked decoder for abstractive summarization, which masks partial tokens in decoder, aiming to alleviate over-reliance on the antecedent, shown as Figure 1. Our MDSumma extends textual representation and text generation by leveraging masked mechanism in decoder, forcing decoder rely more on the source representation other than previous tokens, encouraging to extract more useful information from encoder, breaking the propagation of previous error.

Recent advances in abstractive summarization are driven by the success of encoder-decoder framework [5] and attention mechanism [6]. These models commonly use maximum likelihood estimation (MLE) principle for training, which are more likely to generate the words with higher frequency in the training corpus. In this way, the generated summaries are in general curt and inflexible. However, we argue that different people might generate different summaries from the same source article according to the diversity of language. So, compared to 'computer summary', the 'human summary' is more readable without being restricted by fixed mode. So, we aim to design a model which generates summaries not only keeping in line with ground-truth but also reserving diversity.

In order to diversify the generated summaries, we propose a conditional variational autoencoder (CVAE) based framework for text summarization. This framework takes both the source article and target summary to guide the model learning. The source article is used as a conditional information to guide the decoder. The target summary is represented by VAE which learns the parameters of a probability distribution of the summary, instead of encoding it into a vector. VAE introduces a continuous latent variable sampled from the probability distribution to explicitly model underlying semantics of the target summary, which advances the flexibility and diversity of textual representation. Due to VAE's great capacity of sampling, our model could generate more diverse summaries. In this framework, the decoder may bypass the sampled latent variable and focus solely on modeling the conditional information. In order to enforce decoder modeling both the conditional information and the latent variable equally, we use a fusion mechanism, which make the decoder more robust.

Experiments are conducted on three large-scale corpus, CNN/DM, XSum and ByteCup, to evaluate the efficiency of our method. We not only use the traditional metric ROUGE to evaluate the holistic performance, but also propose a new metric to evaluate the diversity. Diversity is designed based on entropy which measure the variety of words. It also employs KL divergence to ensure the frequency distribution of generated words follow the $Zipf$ law. [1] Experimental results show that our model achieves significant improvements both on ROUGE and diversity, compared with several state of the art methods. To sum up, our contributions are as follows:

- A novel masked decoder model is proposed to remodeling the dependency and alleviate over-reliance on antecedent, extracting more information from encoder, breaking the propagation.
- A variational autoencoder module to advance the flexibility and diversity.
- A new metric to evaluate the diversity of natural textual representation.
- Extensive experiments on three datasets verify the effectiveness of the proposed approach.

II. Related Work

Text summarization is a task of automatically generating a shorter version while remaining the main information and the task has two paradigms: abstractive summarization and extractive summarization. Abstractive summarization requires text rewriting and may contain words not appeared in the original text. Most recent generative models [12], [13] are based on Seq2Seq and attention mechanism. However, the generated summaries are liable to reproduce factual details inaccurately and tend to repeat themselves. Pointer mechanism [11], GAN model [17], reinforcement learning [16] and, bottom-up attention [18] are introduced to promote the performance of the basic Seq2seq model. While the deep semantic information is still hard to control through these RNN-based generative approaches. MASS [2] and UniLM [22] advance summarization by applying masked and deep language model, reaching satisfied performance. For extractive models, Hierarchical Transformer [23] and BERTSUMEXT [19] are pre-trained with large unlabeled data with deep unidirectional architectures, and achieve state-of-the-art performance on CNN/DM with extractive method.

VAE [24] has shown strong capability for improving the diversity of text generation tasks, such as machine translation [25] and question generation [26]–[28]. Latent variable model is a statistical model that seek to model the relationship of observed variables with a set of unobserved, latent variables, and can allow for modeling of more complex, generative processes [29]. [26] addressed the diversity of generation in question generation with variational attention, but doesn't pay attention to the subtle dependence between sampled latent variable and summarization.

III. Model

A. Preliminary

Let X denotes an source text document . Let Y denotes the corresponding summary, and the masked summary is Y'. For convenience, lowercase x and y is equivalent to X and Y.

[1] Zipf's law states that Given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table.

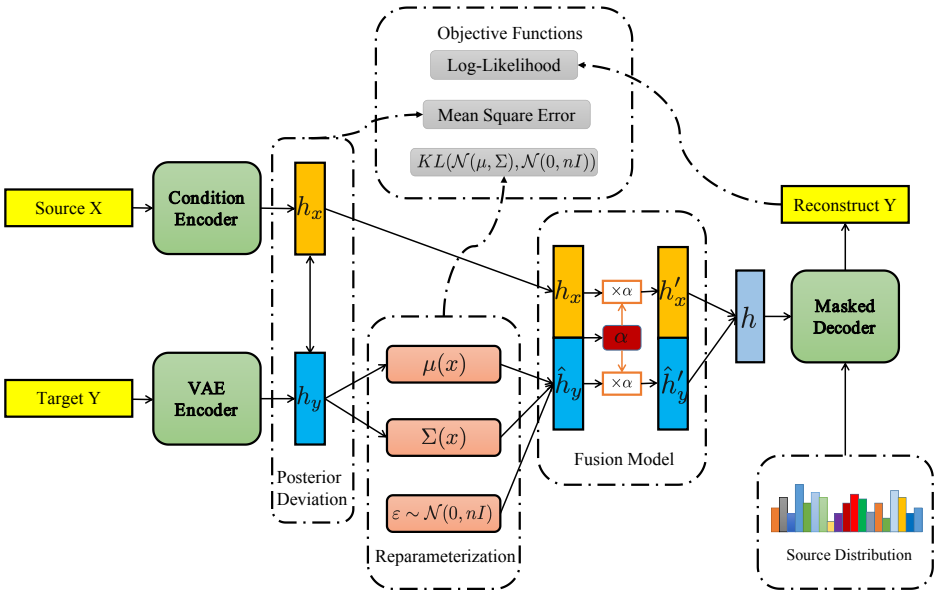

Fig. 2. **Architecture of the MDSumma**. For each pair (article, summary), we will compress the article and summary separately with two encoders, and there only be one decoder for generation. For convenience of illustration, we omit attention mechanism from the architecture figure.

We separate our model with modules, as shown in Figure 2:

- **Condition Encoder.** This encoder encodes the source X into a vector h_x with pre-trained BERT.
- **VAE Encoder.** This encoder is composed of shared pre-trained BERT and encodes the target Y into a vector h_y. We can get a gaussian distribution with $\mu(x)$ and $\Sigma(x)$, which is generated from h_y. With the reparameterization method, we can sample a vector $\hat{h}_y$ from gaussian distribution and make the back propagation work effectively. We should have a restrain on the $\mathcal{N}(\mu,\Sigma)$ with $\mathcal{N}(0,nI)$, so we design another objective function.
- **Fusion Model.** This model encourages the decoder to generate summaries based on both source vector h_x and sampled vector $\hat{h}_y$ equally.
- **Masked Decoder.** The masked decoder randomly masks some tokens and replace them with `[MASK]`. This decoder takes the mixed vector from fusion model, masked sequence $\hat{Y}$ and the attention-based h_t^* (not shown in architecture Figure 2). It employs Transformer decoder and will generate the reconstructed target summary Y.
- **Prior & Posterior Distributions.** To restrain the state space and accelerate the speed of convergence, we utilize the prior distributions of X and $Y_{<t}$, the posterior distributions of h_x and h_y.

In training, the condition encoder and VAE encoder will deliver source and target information to fusion model. Then the mixed semantic vector emitted by fusion model is the initial hidden state for decoder. While in the process of inference and generation, there is no VAE encoder information in our architecture. Under the effective guidance of posterior knowledge distribution in training, h_x can approximately access the same semantic with h_y when the posterior knowledge of target sequence is not available.

B. Conditional Variational Autoencoder

VAE assumes that it is easier to optimize parametric distribution $p_\theta(y,z)$ defined over the words of a sequence y, as well as a latent representation z. It is therefore possible to sample y from the distribution $p(y|z)$, since the latent variable follows a certain distribution instead of being a deterministic value. So, the decoder is able to generate summary via sampling.

In our task, we hope the the source articles could provide a guidance for VAE generators. So, we make the original article serve as a condition for the generation of summary in $p(y|z,c)$, where c is the source tokens x, to restrain the semantic latent vector z emitted by VAE encoder. In this way, VAE is able to generate summary from a fine-grained setting which can further enforce generation and diversity.

The log-likelihood of y can be written as follows, with introducing a conditional latent distribution $q_\phi(z|y,c)$:

$$\begin{aligned}
\ln p_\theta(y|c) &= \sum_z q_\phi(z|y,c) \ln p_\theta(y|c) \\
&= \sum_z q_\phi(z|y,c) \ln \frac{p_\theta(z,y|c)}{q_\phi(z|y,c)} \frac{q_\phi(z|y,c)}{p_\theta(z|y,c)} \\
&= \sum_z q_\phi(z|y,c) \ln \frac{p_\theta(z,y|c)}{q_\phi(z|y,c)} \\
&\quad + KL(q_\phi(z|y,c)||p_\theta(z|y,c))
\end{aligned} \tag{1}$$

where KL is Kullback-Leibler divergence, which is used to measure the proximity between $q_\phi(z|y,c)$ and $p_\theta(z|y,c)$

Since KL-divergence is non-negative, the low-bound of $\ln p_\theta(y|c)$ is the first part of Equation 1, aliased as $\mathcal{L}$. So we can directly maximize $\mathcal{L}$, which is shown as below:

$$\begin{aligned}
&\sum_z q_\phi(z|y,c) \ln \frac{p_\theta(z,y|c)}{q_\phi(z|y,c)} \\
&\quad = \sum_z q_\phi(z|y,c) \ln \frac{p_\theta(z|c)p_\theta(y|z,c)}{q_\phi(z|y,c)} \\
&\quad = -KL(q_\phi(z|y,c)||p_\theta(z|c)) \\
&\qquad + \mathbb{E}_{z\sim q_\phi(z|y,c)}[\ln p_\theta(y|z,c)] \\
&\quad = -KL(q_\phi(z|y,c)||p_\theta(z|c) \\
&\qquad + \frac{1}{N}\sum_{i=1}^{N} ln(p_\theta(y|z_i,c)
\end{aligned} \tag{2}$$

where $p_\theta(z|c)$ is a prior distribution over the latent space, typically the same as $\mathcal{N}(0, nI)$. The expectation over the model distribution $q_\phi(z|y,c)$ is approximated with N samples $z_i \sim q_\phi(z|y,c)$. In the testing stage, z_i is directly sampled from the learned latent space.

The KL-divergence of gaussian distributions, such as $q_\phi(z|y,c) \sim \mathcal{N}(\mu_1, \sigma_1)$ and $p_\theta(z|c) \sim \mathcal{N}(0, \sigma_2)$, can be calculated as below:

$$\begin{aligned}
\int q_\phi \log q_\phi dx &= -\frac{1}{2}(1 + \log 2\pi\sigma_1^2) \\
\int q_\phi \log p_\theta dx &= \frac{1}{2}\log(2\pi\sigma_2^2) + \frac{\sigma_1^2 + \mu_1^2}{2\sigma_2^2}
\end{aligned} \tag{3}$$

$$KL(p,q) = \frac{1}{2}\left(-\log\frac{\sigma_1^2}{\sigma_2^2} + \frac{\sigma_1^2+\mu_1^2}{\sigma_2^2} - 1\right) \tag{4}$$

where σ_2 is used to manage diversity and we can sample z from a probability distribution with larger variance, such as $\mathcal{N}(0, 2I)$.

The final objective function of CVAE model is shown as follows:

$$\begin{aligned}
loss_{cvae} = &-\frac{1}{N}\sum_{i=1}^{N} ln(p_\theta(y|z_i,c) \\
&+ KL(q_\phi(z|y,c)||p_\theta(z|c)
\end{aligned} \tag{5}$$

where θ and ϕ are parameters corresponding to each model. The variational bound breaks into two terms: the data Maximum-Likelihood Loss (MLLoss) and the Kullback-Leibler divergence loss (KLDivLoss).

C. Masked Decoder Model

We introduce a novel masked mechanism for decoder in this section. Given a summary y, we randomly mask 15% of all tokens and the modified version of summary is $\hat{y}$. We replace each masked token by a special symbol [MASK], and the length of masked summary is not changed. MDSumma predict the token y^i by taking the masked sequence $\hat{y}_{<i}$. We also use the log likelihood as the objective function:

$$\begin{aligned}
L(\theta;(\mathcal{X},\mathcal{Y})) &= \sum_{(x,y)\in(\mathcal{X},\mathcal{Y})} \log p_\theta(y|x) \\
p_\theta(y|x) &= \prod_{i=1}^{n} p_\theta(y^i|\hat{y}_{<i}, x)
\end{aligned} \tag{6}$$

With the masked $\hat{y}_{<i}$, x and unmasked tokens $\hat{y}'_{<i}$ are crucial for the prediction, which will alleviate the over-reliance on the previous token.

D. Fusion Model

Unlike the standard Seq2Seq for text summarization, where the initial semantic information for decoder is fully specified by the source sequence, the sampled latent variable is significant for generation. However, the subtle dependencies between sampled latent variable and summarization are hard to model.

Inspired by [7], we design a fusion mechanism to concatenate the compressed vector by condition encoder and the latent vector sampled by VAE encoder, as shown in Figure 2. The fusion model will learn a gate mechanism to leverage these two latent vectors for decoding during training.

$$\begin{aligned}
\alpha &= \sigma(W[h_x, \hat{h}_y] + b) \\
h'_x &= \alpha \otimes h_x \\
\hat{h}'_y &= \alpha \otimes \hat{h}_y
\end{aligned} \tag{7}$$

where W, b are parameters learned in the training process.

E. Prior & Posterior Distributions

In the text summarization task, the source sequence has similar word distribution and semantic information with target sequence. The training of fusion-based CVAE model is time consuming, so we employ some distributions to guide the generation and accelerate convergence.

a) Prior Distributions: The semantic latent vector is restricted with a prior distribution by KLDivLoss. When predicting the target word y_t at time step t, the summary should obey the prior distributions from source sequence $\mathbf{x}$ and target sequence $y_1, y_2, ..., y_{t-1}$.

In fact, most words in the generated summary have appeared in the source article. So, we can merge a prior distribution $\boldsymbol{\psi}$ of source sequence to decoder, which will make the generator prefer to generate the words in source article. We can get a binary vector as prior distribution $\boldsymbol{\psi}$ with a fix dimension of V, i.e., the size of vocabulary.

$$\boldsymbol{\psi} = (\psi_1, \psi_2, ..., \psi_V) \tag{8}$$

Here, if the ith word in vocabulary appears in the source sequence x, $\psi_i = 1$; otherwise, $\psi_i = 0$.

When predicting the word w_t at time step t, we also make use of the prior distribution of target sequence $[y_1^*, y_2^*, ..., y_{t-1}^*]$ to deliver more information to the decoder. But we can't apply that information by the same way as source sequence. The prior distribution of the target words is changing over time, which will occupy large memory of GPU to update. So

we merge the prior distribution of the target sequence into attention, similar to the mechanism of Coverage [8].

The final formula of prior distributions is as follows:

$$\begin{aligned} e_i^t &= v^T tanh(W_h h_i + W_s s_t + W_c c_t + b) \\ \alpha^t &= softmax(e^t) \\ h_t^* &= \sum_i \alpha_i^t h_i \\ s_t &= RNN(s_{t-1}, y_{t-1}) \\ y_t &= V[s_t, h_t^*] + b \\ \hat{y}_t &= \boldsymbol{v} \otimes \boldsymbol{\psi} \\ P_{vocab} &= softmax(y_t + \hat{y}_t) \end{aligned} \tag{9}$$

where $c_t = \sum_{j=0}^{t-1} \alpha^j$, which is the sum of all previous attention distributions and $\boldsymbol{v}$ is the learnable vector parameter.

b) Posterior Distribution: The output of condition encoder is a hidden state vector h_x. The VAE encoder takes ground-truth summary as input, and compress it into a hidden representation vector h_y. There is a posterior probability between h_x and h_y. To achieve this, we define objective function on both source sequence and ground-truth target sequence during the training phase. We can implement assistant supervisor by minimizing the distance between h_x and h_y.

$$loss_{dis} = \frac{1}{H}||h_x - h_y||_2 \tag{10}$$

where H is the size of hidden layer.

However, the target sequence is only available in training. Therefore, in test process, the prior distribution of condition encoder is proposed to precisely approximate the posterior knowledge distribution. For this purpose, we train the prior knowledge distribution using the posterior knowledge distribution as a guidance.

F. Loss Function

Our model consists of several parts, so there are three objective functions. During training, the variational bound breaks into two terms: the data maximum-likelihood and the KL divergence, see the Equation 5. The sum of all these parts is the final loss function:

$$\begin{aligned} loss &= MLLoss + KLDivLoss + MSE \\ &= \lambda_1 loss_{cvae} + \lambda_2 loss_{dis} \end{aligned} \tag{11}$$

where λ_1 and λ_2 are hyper-parameters to balance these loss for the whole model. In experiments, we set them all to $\frac{1}{2}$.

IV. Experiments

A. Dataset

We evaluate our model on three benchmark datasets, namely CNN/DailyMail news highlights dataset [9], XSum [10] and ByteCup 2018. The description of our dataset is shown in Table I.

a) CNN/DailyMail: contains online news articles, paired with multi-sentence summaries. We used the standard splits of Hermann et al. [9] for training, validation and test. We handle non-anonymized version of data with Stanford CoreNLP toolkit and pre-process the dataset following See et al. [11].

b) XSum: contains news articles accompanied with one sentence summary, answering the question "What is this article about?". It is highly abstractive and input documents are truncated to 512 tokens.

c) Byte Cup 2018: Another dataset we use is derived from the "Byte Cup 2018 International Machine Learning Contest" [2]. The competition provides 1,300,000 pairs of article and summary, shown in Table I. We split this dataset into the training, validation and testing set as follows:

B. Baselines

Lead-3: This model directly chooses the first three sentences from a document as its summary. Because in English, the core information of an article is always at the beginning. So, this model could achieve a high ROUGE score.

Attention-Based Seq2Seq(ABS): This is a common architecture of encoder and decoder, applied by many NLP tasks [12], [13].

SummaRuNNer: A recurrent neural network based on sequence model for extractive summarization of document with sentence-level extractive labels [14].

Pointer-Generator(PGC): This model can deal well with out-of-vocabulary problem and repeated words [11].

Unified Model(Unified): This model combines the strength of extractive and abstractive summarization with word-level and sentence-level attentions [15].

DeepRL: This model introduces reinforcement learning into text summarization and make summary more readable [16].

GAN: This model applies generative adversarial network in abstractive text summarization [17].

Bottom-Up: They use a data-efficient content selector as a bottom-up attention step to constrain the model to likely phrases and a two step process achieves significant improvements on ROUGE [18].

MASS: Mass is inspired by BERT and it can jointly train decoder and encoder to develop the capability of representation extraction and language modeling [2].

BERTSUMABS & BERTSUMEXTABS: BERTSUM introduce a novel document-level encoder based on BERT, then stacking several layers and decoder for extractive and abstractive summarization [19].

C. Definition of Diversity

We aim to design a model which generates summaries not only keeping in line with the ground-truth but also reserving the diversity.

Diversity is designed based on entropy which measures the variety of words. The larger the entropy of generated words, the larger diversity the model has. Besides, we consider that

[2]https://biendata.com/competition/bytecup2018/data/

Datasets	# docs (train / val / test)	avg.doc length		avg.summary length	
		words	sentences	words	sentences
CNN	90,266 / 1,220 / 1,093	760.50	33.98	45.70	3.59
DailyMail	196,96 / 12,148 / 10.397	653.33	29.33	54.65	3.86
XSum	204,045 / 11,332 / 3,452	431.07	19.77	23.26	1.00
Byte Cup 2018	1022,176 / 10,523 / 10,581	654.89	30.25	11.90	1.00

TABLE I
DATA STATISTICS: CNN/DAILY MAIL, XSUM, BYTE CUP DATASETS, PROCESSED BY STANFORD CORENLP

the generated words should follow the $Zipf$ law, which is a well known linguistics law. $Zipf$ declares that a few very high-frequency words account the most tokens. So, we design a penalty term to measure the gap between the statistics of generated words and Zipf, which is measured by the KL divergence. The Diversity is formalized as follows.

$$\begin{aligned} H(X) &= \sum_{w_i \in X} p_{w_i} \log_2 p_{w_i} \\ Div(X) &= H(X) - KL(P_w || Q_{zipf}) \end{aligned} \tag{12}$$

where $p_{w_i} = \frac{count(w_i)}{\sum_{j=1}^{N} count(w_j)}$. The words with low $count(w_i)$, especially less than 5, would be ignored.

D. Implementation Details

We utilize the pre-trained BERT[3] for sentence encoder and fine-tune them in training process. Our code is base on BERTSUMABS [4]. Following the recommended settings of BERT [3], we set a maximum sentence length of 128 tokens and total length of 512 tokens. We train the model for 100,000 steps with batch size of 16 on four Nvidia Tesla V100 GPUs. Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$ is used as optimizer. Learning rate schedule follows with warming-up on first 10,000 steps using the strategies in Transformer [4]. The dropout rates in all layers are 0.1. The common evaluation metric used for summarization are F1 of ROUGE-1, ROUGE-2, ROUGE-L[5], which are computed based on overlapping lexical units between generated summaries and golden ones.

The Generative strength of our model is associated with different sampling schemes of latent variable z. We try to use different σ_2 for the prior distribution $p_\theta(z|c) \sim \mathcal{N}(0, \sigma_2)$ in KLDivLoss, such as $\sigma_2 = \{0.5I, I, 1.5I, 2I, 3I\}$. In experiments, we set σ_2 with I and we find that σ_2 with larger or smaller value will cause model unstable and worse rouge score.

V. RESULTS

A. Evaluation of ROUGE

In experiments, we use the common metrics ROUGE-1, ROUGE-2, ROUGE-L, which are computed based on overlapping lexical units between generated summaries and golden ones. We compare the performance between our model and different kinds of comparative methods.

[3]https://github.com/huggingface/transformers
[4]https://github.com/nlpyang/PreSumm
[5]https://github.com/andersjo/pyrouge

Table 2 presents the ROUGE and Diversity scores of each model. Our MDSumma performs the best and significantly outperforms all baseline models on three datasets, demonstrating that masked decoder can model suitable dependency between encoder and previous generated sequence, bridging the gap between autoregressive training and inference. For more in-depth performance analysis, we note that MDSumma without CVAE performs well on almost all ROUGE metrics except for ROUGE-L in XSum, while MDSumma with CVAE is more diverse. That because CVAE samples continuous latent variables from the probability distribution to model underlying semantics of the target summaries, leading to various similar summaries. In contrast to CNN/DM, XSum is more difficult to handle with these model trained from scratch, for there is a large margin between corresponding ROUGE scores. While these models with pre-training process will learn about more knowledge, which is transferred and benefit on the summarization task, especially for XSum datasets. RNN-based approaches perform worse than Transformer variations, the only comparable result are Bottom-Up on CNN/DM, where data are highly extractive, indicating that attention mechanism even bidirectional encoder representations in transformer and its offsprings are mainstream in the future. The fundamental TransformerABS achieves satisfactory results on three datasets, but it is clearly inferior to other pre-trained models. This is likely due to the fact that training on extensive corpus is import for natural language process.

B. Evaluation of Diversity

As mentioned above, we use diversity to measure the flexibility of the summarization models. From Table 2, we can find that: (i) the extractive summarization methods perform worst in diversity, because they directly select words and phrases from the original articles; (ii) the basic sequence to sequence model with attention mechanism (ABS) achieves good performance in diversity, while using GAN and RL framework reduces the diversity of basic model; (iii) CVAE performs better than the extractive methods and other abstractive methods, which proved the effectiveness of the CVAE framework.

We aim to design a model which generates summaries not only keeping in line with the ground-truth but also reserving the diversity. MDSumma with CVAE achieves the best and there is obvious boundary between VAE-base and non-VAE methods, for the reason that the popular choice is the maximum likelihood estimation and it likely to generate words

Models (%)	CNN/DM				XSum				ByteCup			
	R-1	R-2	R-L	*Div*	R-1	R-2	R-L	*Div*	R-1	R-2	R-L	*Div*
Lead-3	40.34	17.70	36.57	6.20	-	-	-	-	-	-	-	-
TextRank	40.20	17.56	36.44	6.18	-	-	-	-	-	-	-	-
SummaRuNNer	38.66	16.11	34.77	6.14	-	-	-	-	-	-	-	-
ABS	35.46	13.30	32.65	6.41	28.42	8.77	22.48	6.42	36.02	18.81	33.35	6.39
PGC	39.53	17.28	36.38	6.38	28.10	8.02	21.72	6.37	39.13	21.20	35.68	6.48
Unified	40.19	17.67	36.68	6.39	29.67	9.34	22.92	6.47	40.86	21.57	35.88	6.40
DeepRL	39.76	15.82	36.90	6.29	28.53	9.07	22.66	6.28	40.03	21.82	36.03	6.53
GAN	39.92	17.65	36.71	6.33	28.74	9.12	22.83	6.31	39.44	21.31	35.83	6.34
Bottom-Up	41.22	18.68	38.34	6.28	30.21	11.36	24.59	6.38	41.22	22.93	37.64	6.58
MASS	42.12	19.50	39.01	6.30	35.88	15.14	28.86	6.45	43.25	23.98	40.23	6.43
BERTSUMABS	41.72	19.39	38.76	6.33	38.76	16.33	31.15	6.40	43.14	23.72	39.91	6.50
BERTSUMEXTABS	42.13	19.60	39.18	6.40	38.81	16.50	31.27	6.45	44.33	24.88	40.68	6.47
MDSumma (w/o CVAE)	**42.53**	19.82	**39.33**	6.31	**39.14**	**16.94**	31.42	6.47	**44.54**	**24.97**	**40.83**	6.51
MDSumma	42.47	**19.84**	39.27	**6.54**	39.03	16.83	**31.45**	**6.58**	44.41	24.83	40.78	**6.69**

TABLE II

EVALUATION OF COMPARATIVE METHODS ON THREE DATASETS: ROUGE AND DIVERSITY SCORES. TOP PART IS EXTRACTIVE MODELS, MIDDLE PART IS ABSTRACTIVE MODELS AND THE BOTTOM IS OUR MODELS. EXTRACTIVE APPROACHES ARE NOT SUITABLE FOR XSUM AND BYTECUP DATASETS. THE DIV SCORE IS CALCULATED ON GENERATION RESULT WITH THE BEST ROUGE SCORE.

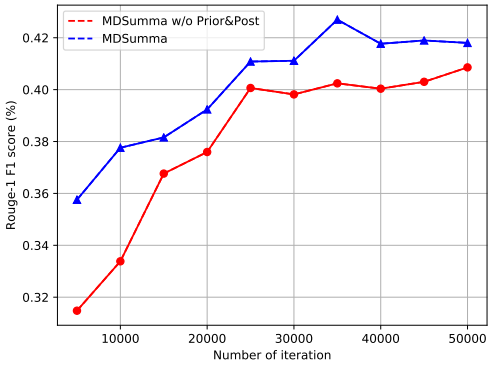

Fig. 3. First 50K iterations of Rouge-1 F1 scores on CNN/DM.

Source article (truncated): built at a cost of # 1 billion, new broadcasting house is the jewel in the crown of the bbc and the setting for its self-mocking satire w1a. (...) new broadcasting house is home to three 24-hour news channels, nine radio networks and 6,000 staff.
Reference: new broadcasting house in central london took a decade to build. it was opened by the queen in 2013 at least # over budget. but the bbc has now admitted it 'occasionally' runs out of meeting rooms.
MDSumma 1: new broadcasting house in central london covers half a million square feet, took a decade to build and ***was opened*** by the queen in 2013. ***bbc has admitted*** it 'occasionally' runs out of meeting rooms and ***spent # on booking external spaces nearby*** during the last financial year.
MDSumma 2: new broadcasting house in central london covers half a million square feet , took a decade to build and ***was opened*** by the queen in 2013 – four years behind schedule and at least # 55 million over budget. ***bbc has been criticised*** by spending watchdog the national audit office over the running ***costs of new broadcasting house***.
MDSumma 3: new broadcasting house in central london covers half a million square feet, took a decade to build and ***opened*** by the queen in 2013 – four years behind schedule and at least # 55 million over budget. ***the bbc has admitted*** it 'occasionally' runs out of meeting rooms and ***spent # on booking external spaces nearby*** during the last financial year.
ABS: built at a cost of # billion, new broadcasting house is the jewel in the crown of the bbc. but in a development that could have come straight out of the sitcom, it has been revealed that the corporation is paying tens of thousands of pounds of taxpayers money to book meetings in nearby buildings because the headquarters lacks space.
PGC: new broadcasting house in central london covers half a million square feet. bbc has admitted it 'occasionally' runs out of meeting rooms and spent # 55 million over budget.

TABLE III

THE GENERATION DIVERSITY: A TEST EXAMPLE OF CNN/DM:

The ***green fonts***, ***red fonts*** and ***cyan fonts*** show the diversity of our model. Our model can generate summaries with different phrase in the period of generation because of the non-deterministic latent semantic vector, while other models are settled and boring.

with higher frequency. That demonstrates the effectiveness of latent variables. The more lower diversity score is, the more inflexible for generated summaries. And our MDSumma with CVAE is the most readable in all models.

Besides, we give some summary cases generated by different models, as shown in Table III. We can find that, for each source article, our model could generate various summaries without weakening the accuracy. There is another simple way to get diverse generations, top-K sampling. While, top-K samplings share identical encoder context, and in generation process, the suboptimal choice of current word will accelerate more error for the rest of the sequence prediction, or syntax errors emit when replace token directly.

C. Avoiding collapsing to prior

We usually observe KL collapsing in VAE, with strong generators learn to ignore latent representation, and the approximate posterior "collapses" to the prior. That means the posterior is independent of the data [20]. There are two techniques to alleviate the collapsing: KL annealing and target word dropout. KL annealing consists in incorporating the KL term into objective gradually, thus allowing the posterior to move away from prior more freely at early stages of training [21]. Target word dropout is randomly masking words generated previously, and we have the same mechanism in masked decoder. In our model, the annealing steps is 50,000, and the validation $KL(q_\phi(z|y,c)\|p_\theta(z|c)$ is 1.37, indicating

that approximate posterior is different from prior at then end of training.

D. Evaluation of Convergence

We experiment the convergence acceleration of prior and posterior distributions. As shown in Figure 3, MDSumma can accelerate the convergence further with these distributions. Here we only consider the prior & posterior limitation in the start of training process, which can help to find the direction of gradient descent quickly. Figure 3 shows the first 50,000 iterations, with reaching the highest performance of MDSumma, but not the other. The training curve of MDSumma module shows that our approach significantly improves basic abstractive methods.

VI. Conclusions

In order to enhance textual representation for abstractive summarization, this proposes a masked decoder to alleviate the propagation of prediction error and remodeling dependency during text generation, a conditional variational autoencoder to facilitate the flexibility and diversity of textual representation. With the prior & posterior distributions, we accelerate the process of summarization in training process. Experiments on CNN/DM, XSum and ByteCup datasets show the effectiveness of our model on ROUGE and Diversity. In the future work, we will further try to breaking the propagation of previous error efficiently.

VII. Acknowledgments

This work is supported by the National Natural Science Foundation of China (NO.Y850371101). We thank all authors for their contributions and all anonymous reviewers for their constructive comments.

References

[1] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," *arXiv preprint arXiv:1511.06732*, 2015.

[2] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "Mass: Masked sequence to sequence pre-training for language generation," *arXiv preprint arXiv:1905.02450*, 2019.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[5] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112. [Online]. Available: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[7] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," *arXiv preprint arXiv:1805.04833*, 2018.

[8] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2016, pp. 76–85. [Online]. Available: http://aclweb.org/anthology/P16-1008

[9] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Advances in Neural Information Processing Systems*, 2015, pp. 1693–1701.

[10] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization," *arXiv preprint arXiv:1808.08745*, 2018.

[11] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *CoRR*, vol. abs/1704.04368, 2017. [Online]. Available: http://arxiv.org/abs/1704.04368

[12] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," *CoRR*, vol. abs/1509.00685, 2015. [Online]. Available: http://arxiv.org/abs/1509.00685

[13] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang *et al.*, "Abstractive text summarization using sequence-to-sequence rnns and beyond," *arXiv preprint arXiv:1602.06023*, 2016.

[14] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *AAAI*, 2017.

[15] W.-T. Hsu, C.-K. Lin, M.-Y. Lee, K. Min, J. Tang, and M. Sun, "A unified model for extractive and abstractive summarization using inconsistency loss," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2018, pp. 132–141. [Online]. Available: http://aclweb.org/anthology/P18-1013

[16] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=HkACIQgA-

[17] L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, and H. Li, "Generative adversarial network for abstractive text summarization," in *AAAI*, 2018.

[18] S. Gehrmann, Y. Deng, and A. M. Rush, "Bottom-up abstractive summarization," *arXiv preprint arXiv:1808.10792*, 2018.

[19] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," *arXiv preprint arXiv:1908.08345*, 2019.

[20] B. Eikema and W. Aziz, "Auto-encoding variational neural machine translation," *arXiv preprint arXiv:1807.10564*, 2018.

[21] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio, "Generating sentences from a continuous space," *CoRR*, vol. abs/1511.06349, 2015. [Online]. Available: http://arxiv.org/abs/1511.06349

[22] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, "Unified language model pre-training for natural language understanding and generation," *arXiv preprint arXiv:1905.03197*, 2019.

[23] X. Zhang, F. Wei, and M. Zhou, "HIBERT: document level pre-training of hierarchical bidirectional transformers for document summarization," *CoRR*, vol. abs/1905.06566, 2019.

[24] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[25] B. Zhang, D. Xiong, J. Su, H. Duan, and M. Zhang, "Variational neural machine translation," *arXiv preprint arXiv:1605.07869*, 2016.

[26] H. Bahuleyan, L. Mou, O. Vechtomova, and P. Poupart, "Variational attention for sequence-to-sequence models," in *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1672–1682. [Online]. Available: https://www.aclweb.org/anthology/C18-1142

[27] Z. Fan, Z. Wei, P. Li, Y. Lan, and X. Huang, "A question type driven framework to diversify visual question generation," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 4048–4054. [Online]. Available: https://doi.org/10.24963/ijcai.2018/563

[28] U. Jain, Z. Zhang, and A. Schwing, "Creativity: Generating diverse questions using variational autoencoders," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, ser. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. United States: Institute of Electrical and Electronics Engineers Inc., 11 2017, pp. 5415–5424.

[29] A. Pagnoni, K. Liu, and S. Li, "Conditional variational autoencoder for neural machine translation," *arXiv preprint arXiv:1812.04405*, 2018.