T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion

Tianming Wang and **Xiaojun Wan**

Institute of Computer Science and Technology, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
{wangtm, wanxiaojun}@pku.edu.cn

Abstract

Story completion is a very challenging task of generating the missing plot for an incomplete story, which requires not only understanding but also inference of the given contextual clues. In this paper, we present a novel conditional variational autoencoder based on Transformer for missing plot generation. Our model uses shared attention layers for encoder and decoder, which make the most of the contextual clues, and a latent variable for learning the distribution of coherent story plots. Through drawing samples from the learned distribution, diverse reasonable plots can be generated. Both automatic and manual evaluations show that our model generates better story plots than state-of-the-art models in terms of readability, diversity and coherence.

1 Introduction

Story completion is a task of generating the missing plot for an incomplete story. It is a big challenge in machine comprehension and natural language generation, related to story understanding and generation [Winograd, 1972; Black and Bower, 1980]. This task requires machine to first understand what happens in the given story and then infer and write what would happen in the missing part. It involves two aspects: understanding and generation. Story understanding includes identifying persona [Bamman *et al.*, 2014], narratives schema construction [Chambers and Jurafsky, 2009] and so on. Generation is the next step based on understanding, regarded as making inference based on clues in the given story. A good generated story plot should be meaningful and coherent with the context. Moreover, the incontinuity of the input text makes the understanding and generation more difficult.

A recently proposed commonsense stories corpus named ROCStories [Mostafazadeh *et al.*, 2016a] provides a suitable dataset for the story completion task. The stories consist of five sentences that reflect causal and temporal commonsense relations of daily events. Based on this corpus, we define our task as follows: given any four sentences of a story, our goal is to generate the missing sentence, which is regarded as the missing plot, to complete this story. Many previous works focus on selecting or generating a reasonable ending for an incomplete story [Guan *et al.*, 2018; Li *et al.*, 2018; Chen *et al.*, 2018]. These tasks are the specialization of our story completion task and thus prior approaches are not suitable for generating the beginning or middle plot of the story. In addition, they tend to generate generic and non-coherent plot. Figure 1 shows an example.

Given Story: My Dad loves chocolate chip cookies. _____. I decided I would learn how to make them. I made my first batch the other day. My Dad was very surprised and quite happy!
Gold standard: My Mom doesn't like to make cookies because they take too long.
Non-coherent: He has been making them all week.
Generic or dull: He always ate them.

Figure 1: An example incomplete story with different generated plots.

To address the issues above, we propose a novel Transformer-based Conditional Variational AutoEncoder model (T-CVAE) for story completion. We abandon the RNN/CNN architecture and use the Transformer [Vaswani *et al.*, 2017], which is a stacked attention architecture, as the basis of our model. We adopt a modified Transformer with shared self-attention layers in our model. The shared self-attention layer allows decoder to attend to the encoder state and the decoder state at the same time. The encoder and decoder are put in the same stack so that information can be passed in every attention layer. This modification helps the model make the most of the contextual clues. Upon this modified Transformer, we further build a conditional variational autoencoder model for improving the diversity and coherence of the answer. A latent variable is used for learning the distribution of coherent story plots and then it is incorporated in the decoder state by a combination layer. Through drawing samples from the learned distribution, our model can generate story plots of higher quality.

We perform experiments on the benchmark ROCStories dataset. Our model strongly outperforms prior methods and achieves the state-of-the-art performance. Both automatic and manual evaluations show that our model generates better story plots in terms of readability, diversity and coherence. Our model also outperforms the state-of-the-art model on the story ending generation task. We further study an interesting phenomenon that the scores of neural models on automatic metrics vary when the position of missing plot in story varies, and we attribute the reason to the structure of human-written

stories. Our contribution can be summarized as follows:

- To the best of our knowledge, this is the first attempt to address the story completion task of generating missing plots in any position and we propose a novel Transformer-based conditional variational autoencoder(T-CVAE) for this task. Our code is available at https://github.com/sodawater/T-CVAE.
- Our model achieves the state-of-the-art performance and both automatic and manual evaluations show that our model can generate better story plots in terms of readability, diversity and coherence.
- We study the difference of generating story plots in different positions.

2 Related Work

2.1 Story Understanding

Several lines of research have been done in the field of story understanding. Early works focus on learning the representation of narratives [Schank and Abelson, 1977; Chambers and Jurafsky, 2008]. Narrative plots understanding [Goyal *et al.*, 2010] and character understanding [Bamman *et al.*, 2014] have also been studied. Recent works attempt to tackle the stoy-cloze task proposed by [Mostafazadeh *et al.*, 2016a], which requires to select a correct ending from two candidates given a story context. Feature-based classification models [Mostafazadeh *et al.*, 2016b; Chaturvedi *et al.*, 2017] measure the coherence between candidates and the given story context from aspects of sentiment and topic. Neural network models have also been applied to this task [Chen *et al.*, 2018].

2.2 Story Generation

In story generation, most previous automatic story generation works are limited to selecting a sequence of events that meet a set of criteria and then generating a story based on the sequence [Li *et al.*, 2013; Martin *et al.*, 2018]. These systems are considered as story planning systems. Recent researches focus on generating coherent and fluent stories about a given topic. These models generate stories based on skeleton [Xu *et al.*, 2018], storyline [Yao *et al.*, 2018] and premise [Fan *et al.*, 2018]. The above story-cloze task has also been expanded to a generation task that requires to generate a reasonable ending for a given story. Model based on adversarial learning [Li *et al.*, 2018] and model leveraging external structured knowledge [Guan *et al.*, 2018] have been proposed for addressing the task, and the latter achieves the state of the art performance.

2.3 Conditional Variational Autoencoder

The variational autoencoder [Kingma and Welling, 2013; Rezende *et al.*, 2014] is one of the most popular frameworks for generation. The basic idea of VAE is to encode the input into a probability distribution z and apply a decoder to reconstruct the input using samples z. Conditional variational autoencoder(CVAE) is a modification of VAE to generate text or image conditioned certain given attributes. VAE/CVAE has been widely used and explored in text generation, especially dialog generation: VAE conditioned on dual encoder [Cao and Clark, 2017], hierarchical VAE [Serban *et al.*, 2017], knowledge-guided CVAE [Zhao *et al.*, 2017] and so on.

3 Our Approach

Our model is a Transformer-based conditional variational autoencoder, which can generate diverse and coherent story plots. We begin by formulating the story completion task. Then our Transformer model with shared self-attention layers will be introduced, which is also the basis of T-CVAE. Finally we will describe our T-CVAE model that incorporates a latent variable for encoding coherent story plots. Figure 2 shows the overall architecture of our model.

3.1 Problem Formulation

The story completion task can be formulated as follows: given an incomplete story consisting of $M-1$ sentences $x = \{s_1, ..., s_{k-1}, s_{k+1}, ..., s_M\}$, where $s_i = w_1^i w_2^i ... w_{n_i}^i$ represents the i-th sentence containing n_i words and k represents the position of the missing sentence in the story, our goal is to generate a one-sentence plot which is coherent with the given context. The model is trained to maximize the probability $p(y|x)$, where y is the gold plot.

3.2 Our Transformer

Our model is adapted from the Transformer, whose overall architecture is composed of a stack of L multi-head attention layers and point-wise, fully connected feed-forward network for both the encoder and the decoder. We omit the background description and follow the formula and notations proposed by [Vaswani *et al.*, 2017] in this paper. We denote queries, keys and values for attention as Q, K and V and multi-head attention as MultiHead(Q, K, V), feedfoward networks as FFN(x) and layer normalization as LayerNorm(x).

Input Representation

Our input representation is different from the original Transformer, since the input text in our task is not continuous. We use a similar idea proposed in [Devlin *et al.*, 2018], where the input representation of a given word is constructed by concatenating the word, segment and position embeddings:

$$IR_{w_j^i} = WE_{w_j^i} \oplus SE_i \oplus PE_j \quad (1)$$

where $IR_{w_j^i}$ is the input representation of j-th word in i-th sentence, $WE_{w_j^i}$ is the word embedding of w_j^i, SE_i is the segment embedding of i-th sentence and PE_j is the position embedding of j-th word. For convenience, we denote the packages of a set of input representations for encoder and decoder as IR^E and IR^D respectively.

Shared Attention Layers

The original Transformer has separated encoder stack and decoder stack and their self-attention layers are independent. It is suitable for machine translation since source language and target language have different distributions. It is better to represent them in different spaces. But in our task, the missing plot to be generated is a part of a story and representing it in the same space as the given context could make the completed story more coherent.

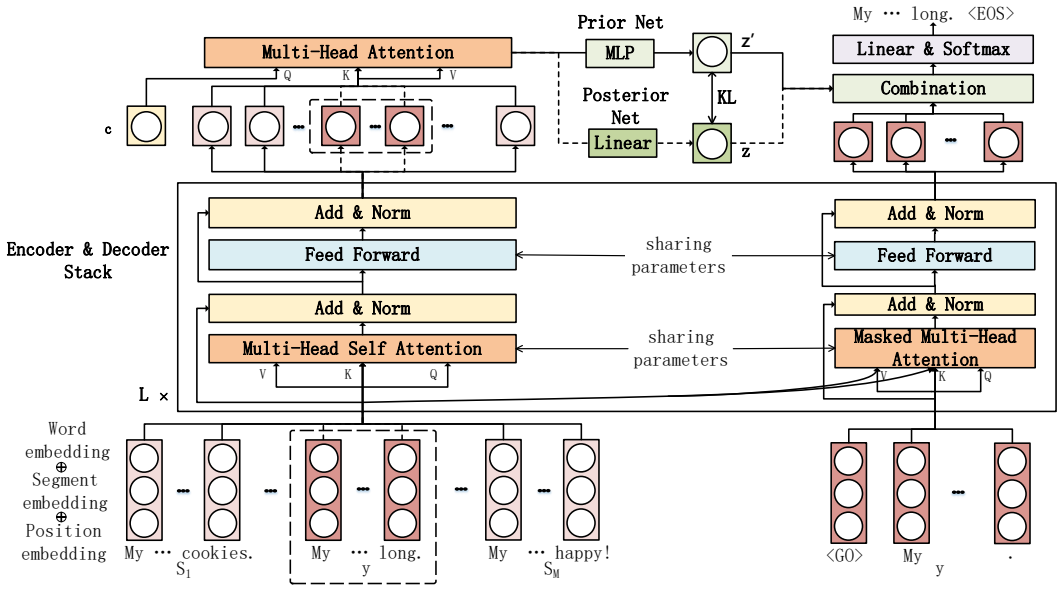

Figure 2: Architecture of our T-CVAE model. Both prior net and the posterior net are built upon the encoder, and the posterior net takes an extra input y which is enclosed by a dashed line. In training phase, latent variable z fed to the combination layer is derived by the posterior net, which is connected by the dashed line; in inference phase, the prior net is used for replacing the posterior net to derive latent variable $z^{'}$, which is connected by solid line. The reparametrization trick is used to obtain samples of latent variable either from z while training or $z^{'}$ while inferring.

To better capture contextual clues, we propose shared attention layers for the encoder and the decoder. It not only means that the attention layers in the encoder and the decoder share the same parameters, but also allows the decoder to attend to the encoder state and the decoder state at the same time. In this way, information can pass between the encoder and the decoder in every layer.

Specially, we denote the input and output of the l-th layer in the encoder and the decoder as E^l_{in}, E^l_{out} and D^l_{in}, D^l_{out} respectively. Particularly, $E^1_{in} = IR^E W_e$ and $D^1_{in} = IR^D W_e$, where $W_e \in \mathbb{R}^{3d_{emb} \times d_{model}}$ is parameter matrix, d_{emb} is the dimension of embedding and d_{model} is the dimension of hidden layers in the model. Then for encoder, the input of multi-head self-attention in the encoder is the same as that in the original Transformer.

$$\begin{aligned} E^l_{in} &= E^{l-1}_{out} \\ A &= \text{MultiHead}(E^l_{in}, E^l_{in}, E^l_{in}) \\ B &= \text{LayerNorm}(A + E^l_{in}) \\ E^l_{out} &= \text{LayerNorm}(FFN(B) + B) \end{aligned} \tag{2}$$

For decoder, the inputs K and V for attention layers are the combination of E^l_{in} and D^l_{in}. Specifically,

$$\begin{aligned} D^l_{in} &= D^{l-1}_{out} \\ A &= \text{MultiHead}(D^l_{in}, [E^l_{in}; D^l_{in}], [E^l_{in}; D^l_{in}]) \\ B &= \text{LayerNorm}(A + D^l_{in}) \\ D^l_{out} &= \text{LayerNorm}(FFN(B) + B) \end{aligned} \tag{3}$$

Similar to the original Transformer, we use a masking in the decoder to ensure that the attention and prediction for position j can depend only on the known words at positions preceding j.

We also share the point-wise, fully connected layers of the encoder and the decoder. The Transformer with shared self-attention layers is the basis of T-CVAE, and it can handle the completion task too. We directly use the linear transformation and the softmax function to convert the final output of the decoder so that it can predict word probabilities and generate words.

$$\begin{aligned} O_t &= D^L_{out,t} W_o + b_o \\ P_t &= \text{softmax}(O_t) \end{aligned} \tag{4}$$

where $D^L_{out,t}$ is the final decoder output at time-step t, $W_o \in \mathbb{R}^{d_{model} \times d_{vocab}}$ and $b_o \in \mathbb{R}^{d_{vocab}}$ are parameters, and d_{vocab} is vocabulary size. P_t is the probability distribution of the word to be generated at time-step t.

3.3 T-CVAE

Upon the Transformer, we further build T-CVAE which uses a latent variable for learning the distribution of the coherent story plots. In T-CVAE, the missing plot y is generated conditioned on the given incomplete story x and a diversity and coherence promoting latent variable z which captures the distribution of the plots. We define the conditional distribution $p(y|x) = \int_z p(y|x,z)p(z|x)dz$ and our goal is to use neural networks to approximate $p(z|x)$ and $p(y|x,z)$. We refer to $p(z|x)$ as the *prior net* and $p(y|x,z)$ as the *plot generator*.

Since the integration over z is intractable, we therefore apply variational inference and optimize the corresponding evidence lower bound (ELBO):

$$\begin{aligned} \log p(y|x) &= \log \int_z p(y|x,z)p(z|x)dz \\ &\geq \mathbb{E}_{q(z|x,y)}[\log p(y|x,z)] \\ &- D_{KL}(q_(z|x,y)||p(z|x)) \end{aligned} \tag{5}$$

where $q(z|x,y)$ is the *posterior net* (i.e. *recognition net*) to approximate the true posterior distribution of the latent variable z, and $D_{KL}(||)$ denotes the KL-divergence. We assume that z follows multivariate Gaussian distribution with a diagonal covariance matrix.

Model Details

Figure 2 demonstrate an overview of our model T-CVAE and the pipeline of the training and inference procedures. In T-CVAE, the prior net and the posterior net are both built upon the encoder of the modified Transformer.

The posterior net encodes both the given incomplete story x and the missing plot y. Since we assume z follows isotropic Gaussian distribution, $q(z|x,y) \sim N(\mu, \sigma^2\mathrm{I})$ and then we have

$$\begin{aligned} h &= \text{MultiHead}(c, E^L_{out}(x;y), E^L_{out}(x;y)) \\ \begin{bmatrix} \mu \\ \log(\sigma^2) \end{bmatrix} &= hW_q + b_q \end{aligned} \tag{6}$$

where c is a context vector(random initialized), which is regarded as a single query for the multi-head attention to get the representation of the story h. $E^L_{out}(x;y)$ stands for the final outputs of the encoder when taking both x and y as input, $W_q \in \mathbb{R}^{d_{model} \times d_z}$ and $b_q \in \mathbb{R}^{d_z}$ are parameters and d_z is the dimension of latent variable.

The prior net only encodes the given story x. Similarly, $p_\theta(z|x) \sim N(\mu', \sigma'^2\mathrm{I})$ and we have

$$\begin{aligned} h' &= \text{MultiHead}(c, E^L_{out}(x), E^L_{out}(x)) \\ \begin{bmatrix} \mu' \\ \log(\sigma'^2) \end{bmatrix} &= \text{MLP}_p(h') \end{aligned} \tag{7}$$

where MLP_p is a multi-layer perceptron.

Different from the RNN-based CVAE, we do not use the latent variable z to initialize the state of the decoder. Instead, we incorporate it to the decoder state by a combination layer.

$$\begin{aligned} C_t &= \tanh([z, D^L_{out,t}]W_c) \\ O_t &= C_t W_o + b_o \\ P_t &= \text{softmax}(O_t) \end{aligned} \tag{8}$$

where $W_c \in \mathbb{R}^{d_{model} \times d_{model}}$ is parameter matrix. C_t is the output of the combination layer at time-step t and is further fed to linear transformation and softmax layer to get the probability distribution.

Training Details

Our model is trained similarly to [Zhao *et al.*, 2017]. Optimizing Eq(6) consists two parts: maximizing the probability of reconstructing y, which can push the predictions made by the posterior net and the plot generator closer to the ground truth; minimizing the KL-divergence between the posterior distribution and the prior distribution of z, which can push the prior net to produce a reasonable probability distribution when the ground truth is no longer available. KL annealing is used during training, which increases the weight of the KL term from 0 to 1 gradually.

4 Experiment

We perform experiments on the ROCStory dataset for evaluating models. The dataset is randomly split by 8:1:1 to get the training, validation and test datasets with 78529, 9817 and 9816 stories respectively. For each story, we randomly choose one sentence at any position of the story as the target to be generated.

4.1 Baselines

We compare our models with the following baselines:

Seq2Seq. We implement a bidirectional-LSTM with attention mechanism as a baseline. We concatenate the scope embedding and the word embedding as the input of the encoder.

HLSTM. The story is encoded by a hierarchical LSTM: a word-level LSTM for encoding each sentence and a sentence-level LSTM for connecting four sentences.

CVAE. We implement a LSTM-based CVAE model, in which the initial state of the decoder is the combination of a latent variable and the final state of the encoder.

Transformer. The original Transformer [Vaswani *et al.*, 2017] is also compared. The same input representation as our model is fed to the encoder.

IE+MSA. [Guan *et al.*, 2018] proposed a model using incremental encoding scheme and incorporated external structured commonsense knowledge for generating endings for the incomplete stories. It achieves state-of-the-art performance on the story ending generation task. We use the released code [1] for training and testing on our dataset. Note that the model can only be used for comparison on story ending generation.

4.2 Parameter Settings

We set our model parameters based on preliminary experiments on the development data. For all models including baselines, d_{model} is set to 512 and d_{emb} is set to 300. For Transformer models, the head of attention H is set to 8 and the number of Transformer blocks L is set to 6. The number of LSTM layers is set to 2. For VAE models, d_z is set to 64 and the annealing step is set to 20000. We apply dropout to the output of each sub-layer in Transformer blocks. We use a rate $P_{drop} = 0.15$ for all models. We use the Adam Optimizer with an initial learning rate of 10^{-4}, momentum $\beta_1 = 0.9$, $\beta_2 = 0.99$ and weight decay $\epsilon = 10^{-9}$. The batch size is set to 64. We use greedy search for all models and initialize them with 300-dimensional Glove word vectors.

4.3 Metric

We conduct both the automatic evaluation and manual evaluation on the test set.

BLEU, B1, B2, B3. The word-overlap score against gold-standard story plot is widely used in many story generation works. BLEU [Papineni *et al.*, 2002] in this paper refers to the default BLEU-4, but we also report on other n-gram scores (B1, B2, B3).

[1] https://github.com/JianGuanTHU/StoryEndGen

Methods	**BLEU%**	**B1%**	**B2%**	**B3%**	**D1%**	**D2%**	**AdverSuc%**	**Gram**	**Logic**
Human	-	-	-	-	7.15	42.98	92.30	2.84	2.80
Seq2Seq	2.90	27.41	10.56	5.20	2.69	15.95	80.97	2.59	1.69
HLSTM	2.31	25.70	9.04	4.26	2.63	14.80	72.46	2.49	1.65
CVAE	3.03	27.73	10.79	5.40	2.72	16.32	81.18	2.52	1.90
Transformer	3.05	27.53	10.70	5.31	2.93	16.75	82.51	2.63	1.92
Our(T-CVAE)	**4.25**	**29.33**	**12.75**	**6.96**	**3.63**	**23.46**	**87.54**	**2.71**	**2.13**

Table 1: Comparison results on the story completion task

Methods	**BLEU%**	**D1%**	**D2%**	**AdverSuc%**
Our(T-CVAE)	**4.25**	**3.63**	**23.46**	**87.54**
-CVAE	3.98	3.50	21.40	86.22
-Shared	3.56	3.05	18.79	84.83
-Shared, -CVAE	3.05	2.93	16.75	82.51

Table 2: Ablation study on story completion. -CVAE means only using Transformer and -Shared means using seperated attention layers.

D1, D2. The proportions of distinct unigrams and bigrams in the outputs [Li *et al.*, 2015] are common metrics to evaluate the diversity of generated results.

AdverSuc. Considering there might exist many reasonable plots for a single incomplete story, BLEU score might be one-sided when only one reference is provided. Adversarial Success is the fraction of instances in which a model is capable of fooling the evaluator [Li *et al.*, 2017], which can reflect the quality of the generated answer. We use a pre-trained coherence model as the evaluator. We treat original human-written stories as positive examples and stories consisting of a random sentence(chosen from another story) as negative examples. We use BERT as the coherence discriminator.[2]

Gram & Logic. We also use two metrics - grammaticality (Gram) and logicality (Logic) for manual evaluation. Gram is used to evaluate whether the generated story plot is natural and fluent while Logic for evaluating whether the plot is reasonable and coherent with the story. The score ranges from 1 to 3. 1 means bad, 2 means okay and 3 means good. We employ crowdsourced judges on Amazon Mechanical Turk to provide evaluations for a random sample of 100 items. Each incomplete story is given and 3 judges are asked to grade the results. The final scores are averaged across different judges and stories.

4.4 Results and Analysis

Automatic and Manual Evaluation

Table 1 presents both automatic and manual evaluation results over different metrics. We can see that our T-CVAE model strongly outperforms other baselines. Among all prior methods, Transformer has the best performance. Our T-CVAE model achieves the state-of-the-art scores on all automatic metrics, which improves the state of art from 3.05% to 4.25% on BLEU, 2.93% to 3.63% on D1, 16.75% to 23.45% on D2 and 82.51% to 87.54% on AdverSuc. Higher BLEU score indicates that the plot generated by our model are more close to the gold standard answer than others. Noted that all methods get a much higher score on BLEU-1 than BLEU-2, BLEU-3, and it means unigrams(especially pronouns and prepositions) are much easier to be matched than bigrams and trigrams. All the methods have high scores on AdverSuc because the negative examples for training are not strong. Our model ranks only second to human and it shows that our model can fool the automatic evaluator than any prior method. Moreover, our method is also significantly better than the baseline models on D1 and D2, which indicates that our model can generate more diverse and non-generic plots for incomplete stories.

Our model also outperforms the baseline models on manual metrics, and it achieves 2.71 and 2.13 on Gram and Logic respectively. The results show that the our model can generate more coherent and readable plots than baselines. We further do t-test on manual evaluation results for our T-CVAE model and the orginial Transformer, and p-values are 0.0452 and 0.00012 on Gram and Logic respectively, which indicates that our T-CVAE model is significantly better than Transformer. The Kappa measuring inter-rater agreement is 0.52, which implies a moderate agreement.

Ablation Study

Table 2 shows the results of ablation study on automatic metrics. Without shared attention layer, the performance of our model drops by 0.69% on BLEU, 0.58% on D1, 4.67% on D2 and 2.69% on AdverSuc respectively, which indicates its effectiveness. These scores also drop when latent variable is removed, which means using latent variable for learning the distribution can help generate more coherent and diverse plots. Removing both shared attention layer and latent variable, the model degrades to the standard Transformer and achieves the lowest score.

Position Study

We also compare our method with the prior method which achieves the state-of-the-art score on the story ending generation task. In Table 3, we can see that our T-CVAE model strongly outperforms IE+MSA on all metrics. Comparing with the results in Table 1, we see that models achieve a lower score on D1 and D2 but a higher score on AdverSuc, which indicates that models tend to generate a generic plot for story ending generation. We can also see that all these models achieve a much lower BLEU score on story ending generation. We guess that the difficulty of generating plots varies from position to position.

We further study this phenomenon and compare the BLEU scores of generating plots in different position k. The results are shown in Figure 3. We can clearly see that BLEU

[2]The pre-trained BERT achieves a 95% accuracy at the classification task on validation dataset.

Methods	BLEU%	B1%	B2%	B3%	D1%	D2%	AdverSuc%	Gram	Logic
IE+MSA	1.73	24.43	8.21	3.50	1.85	9.87	83.08	2.57	1.60
Our(T-CVAE)	**2.61**	**25.74**	**9.87**	**4.80**	**3.05**	**18.86**	**88.92**	**2.73**	**1.97**

Table 3: Comparison results on the story ending generation task

Given story 1	Martin hated storms. _____. Martin scampered to a nearby tree to take cover. He began to beg God to preserve his life. Just at that moment, the clouds parted and Martin felt relieved !
Seq2Seq	One day, he heard a loud noise.
Transformer	He was afraid of storms.
Ours	One day, a big storm came and hit him.
Human	One day Martin was working in the fields when a sudden storm arose.
Given story 2	_____. I discovered him last week. His songs were innovative and funny. I sat there and listened to him all day long. I decided to buy his albums when they are released.
Seq2Seq	I love music.
Transformer	My friend is a rap star.
Ours	My friend is a musician.
Human	My new favorite youtube musician is Nicky.
Given story 3	When I was younger I played basketball in a local league. I was n't very good but I was very tall. One day I accidentally scored a basket for the enemy team! Somehow I thought we were on the other side of the court. _____.
Seq2Seq	I was so happy.
Transformer	i was so sad that i did n't have to play basketball anymore.
IE+MSA	We ended up winning the tournament.
Ours	I was so upset that I quit.
Human	My team laughed it off since it was n't a big deal.

Table 4: Case Study.

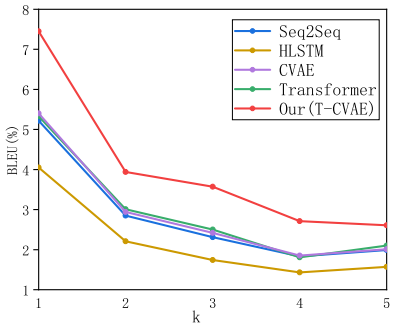

Figure 3: BLEU scores of different models on generating k-th sentence

score goes down as k increases and it drops significantly from $k = 1$ to $k = 2$. Our T-CVAE model achieves a BLEU score of 7.45% when $k = 1$, drops 3.51% when k comes to 2 and only achieves 2.61% on generating the last sentence, i.e. story ending. For Seq2Seq method, it achieves 5.22% on generating the beginning of story and only 1.99% on ending. To explain this phenomenon, we analyze the structure of stories in this dataset and find that plots become more complex when story progresses. In other words, starting plot is simple and generic, paves the way for the follow-ups; subsequent plots become more specific and informative, which are hard to predict. In Figure 3, there is another interesting phenomenon that all the BLEU scores of baseline methods rise again when k goes from 4 to 5. We attribute the reason to the continuity of the input. The given text are continuous when $k = 5$, and separated encoder can learn a better representation compared with the case $k = 4$. Sharing attention layer and putting the encoder and decoder in the same stack enable our model to handle discontinuous input text better.

Case Study

We present some examples of generated story plots in Table 4. We can see our model generates more coherent and reasonable plots than other baselines. But compared with the human-written plots, plots generated by our model still have some deficiencies in informativeness and coherence.

In example 1, both Transformer and our model find the keyword "storms". But the plot generated by Transformer is non-coherent and dull. Story 1 has a progressive structure that we mentioned in above sections: the ending is more specific and informative than the beginning. This is very common in this dataset. In example 2, all the methods generate starting plots about music but the answer generated by Seq2Seq is bad. In example 3, our model generates a generic but reasonable ending while all the baseline methods generate non-coherent endings. In general, neural models tend to generate generic and dull plots like "I was happy" , "It was fun". It is also difficult for our model to completely overcome this.

5 Conclusion

We investigate the problem of generating the missing story plot at any position for an incomplete story. Our proposed T-CVAE model strongly outperforms prior methods. We evaluate models on both automatic and manual metrics and results show that our model can generate plots with better coherence and diversity. We further study the difficulty of generating plots in different positions. Our future work will focus on story completion and story generation task in open domain.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61772036) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We thank the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

References

[Bamman *et al.*, 2014] David Bamman, Brendan O'Connor, and Noah A Smith. Learning latent personas of film characters. In *ACL*, page 352, 2014.

[Black and Bower, 1980] John B Black and Gordon H Bower. Story understanding as problem-solving. *Poetics*, 9(1-3):223–250, 1980.

[Cao and Clark, 2017] Kris Cao and Stephen Clark. Latent variable dialogue models and their diversity. In *EACL, Short Papers*, volume 2, pages 182–187, 2017.

[Chambers and Jurafsky, 2008] Nathanael Chambers and Daniel Jurafsky. Unsupervised learning of narrative event chains. In *ACL*, volume 94305, pages 789–797, 2008.

[Chambers and Jurafsky, 2009] Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative schemas and their participants. In *ACL-IJCNLP 2009: Volume 2-Volume 2*, pages 602–610, 2009.

[Chaturvedi *et al.*, 2017] Snigdha Chaturvedi, Haoruo Peng, and Dan Roth. Story comprehension for predicting what happens next. In *EMNLP*, pages 1603–1614, 2017.

[Chen *et al.*, 2018] Jiaao Chen, Jianshu Chen, and Zhou Yu. Incorporating structured commonsense knowledge in story completion. *arXiv preprint arXiv:1811.00625*, 2018.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Fan *et al.*, 2018] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *ACL*, volume 1, pages 889–898, 2018.

[Goyal *et al.*, 2010] Amit Goyal, Ellen Riloff, and Hal Daumé III. Automatically producing plot unit representations for narrative text. In *EMNLP*, pages 77–86, 2010.

[Guan *et al.*, 2018] Jian Guan, Yansen Wang, and Minlie Huang. Story ending generation with incremental encoding and commonsense knowledge. *arXiv preprint arXiv:1808.10113*, 2018.

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

[Li *et al.*, 2013] Boyang Li, Stephen Lee-Urban, George Johnston, and Mark Riedl. Story generation with crowdsourced plot graphs. In *AAAI*, 2013.

[Li *et al.*, 2015] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

[Li *et al.*, 2017] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. In *EMNLP*, pages 2157–2169, 2017.

[Li *et al.*, 2018] Zhongyang Li, Xiao Ding, and Ting Liu. Generating reasonable and diversified story ending using sequence to sequence model with adversarial training. In *COLING*, pages 1033–1043, 2018.

[Martin *et al.*, 2018] Lara J Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark O Riedl. Event representations for automated story generation with deep neural nets. In *AAAI*, 2018.

[Mostafazadeh *et al.*, 2016a] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *NAACL-HLT*, pages 839–849, 2016.

[Mostafazadeh *et al.*, 2016b] Nasrin Mostafazadeh, Lucy Vanderwende, Wen-tau Yih, Pushmeet Kohli, and James Allen. Story cloze evaluator: Vector space representation evaluation by predicting what happens next. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 24–29, 2016.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002.

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

[Schank and Abelson, 1977] RC Schank and R Abelson. Script, plans, goals and understanding: An inquiry into human knowledge structures. 1977.

[Serban *et al.*, 2017] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pages 3295–3301, 2017.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[Winograd, 1972] Terry Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.

[Xu *et al.*, 2018] Jingjing Xu, Xuancheng Ren, Yi Zhang, Qi Zeng, Xiaoyan Cai, and Xu Sun. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *EMNLP*, pages 4306–4315, 2018.

[Yao *et al.*, 2018] Lili Yao, Nanyun Peng, Weischedel Ralph, Kevin Knight, Dongyan Zhao, and Rui Yan. Plan-and-write: Towards better automatic storytelling. *arXiv preprint arXiv:1811.05701*, 2018.

[Zhao *et al.*, 2017] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL*, volume 1, pages 654–664, 2017.