Content Learning with Structure-Aware Writing: A Graph-Infused Dual Conditional Variational Autoencoder for Automatic Storytelling

Meng-Hsuan Yu,[1*] **Juntao Li,**[1*] **Zhangming Chan**[1]
Rui Yan[2] **and Dongyan Zhao**[1†]
[1]Wangxuan Institute of Computer Technology, Peking University, Beijing, China
[2]Gaoling School of Artificial Intelligence, Renmin University of China
{yumenghsuan,lijuntao,zhangming.chan,zhaody}@pku.edu.cn
ruiyan@ruc.edu.cn

Abstract

Recent automatic storytelling methods mainly rely on keyword planning or plot skeleton generation to model long-range dependencies and create consistent narrative texts. However, these approaches generate story plans or plots sequentially, leaving the non-sequential conception and structural design processes of human writers unexplored. To mimic human writers and exploit the fine-grained, intrinsic structural information of each story, we decompose automatic story generation into sub-problems of graph construction, graph generation, and graph-infused sequence generation. Specifically, we propose a graph-infused dual conditional variational autoencoder model to capture multi-level intra-story structures (i.e., graph) by continuous variational latent variables and generate consistent stories through dual-infusion of story structure planning and content learning. Experimental results on the ROCStories dataset and the CMU Movie Summary corpus confirm that our proposed model outperforms strong baselines in both human judges and widely-used automatic metrics.

Introduction

Human-made stories generally consist of a series of locally coherent events or actions, connected by an overarching structure (Fan, Lewis, and Dauphin 2019). To automatically generate coherent and fluent content, current state-of-the-art techniques mainly consist of two themes. Some existing models (Fan, Lewis, and Dauphin 2018a; Zhang et al. 2019; Goldfarb-Tarrant, Feng, and Peng 2019) exploit a hierarchical generation pipeline to model long-range dependencies. Others have generated stories by first creating intermediate elements e.g., events (Martin et al. 2018), skeletons (Xu et al. 2018a), keywords (Yao et al. 2019), verbs (Tambwekar et al. 2019), arguments (Fan, Lewis, and Dauphin 2019) and then generating the story conditioned on these given elements. By converting long sequence modeling to relatively short dependency learning, these methods effectively enhance the consistency of generated stories. Although they achieve noticeable improvements, these methods rely highly on the recent success of sequence modeling

[*]Equal contribution

[†]Corresponding author: Dongyan Zhao (zhaody@pku.edu.cn)

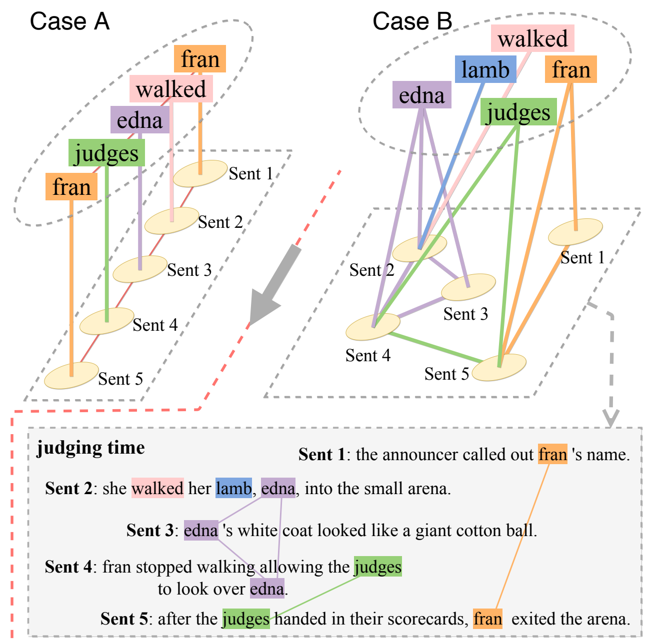


Figure 1: Case A is a typical case of planning based story generation process, while case B represents a more human-like situation for story composition.

methods via sequentially outputting each intermediate element. By doing so, the modeling of intermediate elements with multi-level structure information degenerates to constructing a sequential learning model, which neglects intra-story structures, i.e., interactions between the intermediate elements and story sentences.

To capture intra-story structures, we proposed to leverage multi-level interconnected structural information to automatically generate full stories. Specifically, we combine state-of-the-art storyline planning (Yao et al. 2019; Goldfarb-Tarrant, Feng, and Peng 2019) and hierarchical generation methods (Fan, Lewis, and Dauphin 2018a) to structure both the keywords and each sentence of the story. We consider different types of structured interactions, including sentence-level connections and keyword-sentence interactions, which presents a more human-like situation for

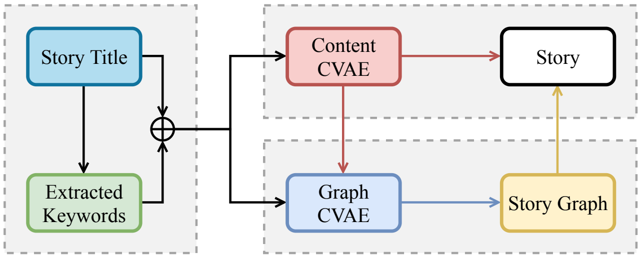


Figure 2: The overview of the GraphD-CVAE.

story composition. Figure 1 gives example cases to intuitively demonstrate the difference between generation with and without interconnected keywords and sentences.

In this paper, we propose a novel graph-infused dual conditional variational autoencoder (GraphD-CVAE), based on the standard CVAE model (Sohn, Lee, and Yan 2015), to capture and seamlessly fuse structure information into story generation. We first extract keywords and then design a graph construction strategy to extend the keywords with sentence-level connections and keyword-sentence interactions to form an intra-story graphical structure, where each vertex represents a keyword or a sentence. To capture content-aware structure information, we also utilize a variational graph autoencoder (VGAE) (Kipf and Welling 2016) as a down-sampling strategy to obtain a contextualized graph for generating each sentence. To fuse the graph information into a story generator, we implement a dual connection between the VGAE model with a CVAE story generation module. By doing so, our proposed method can capture content-aware and structure-aware intra-story information in the story generation process.

We evaluate our method on two benchmark datasets, including the ROCStories dataset (Mostafazadeh et al. 2016) and the CMU Movie Summary corpus (Bamman, O'Connor, and Smith 2013), which comprise of everyday short stories and long narrative texts, respectively. Automatic evaluation results confirm that our proposed method can capture long-range dependencies and multi-level interactions. Through taking structural information into account, our proposed method generates human preferable stories.

Model Overview

Our proposed pipeline evolves from the recent progress in decomposing the challenging story generation task into a few specialized stages to model high-level dependencies (Fan, Lewis, and Dauphin 2019). We structure the story generation task as three sub-problems: keyword extraction, story graph construction, and story content generation. Here, we will provide a brief overview of our GraphD-CVAE model and then further delve into more details.

To prepare training data for solving these sub-problems, we first leverage the RAKE algorithm (Rose et al. 2010) to retrieve keywords for each story automatically. Next, the keywords of the story are expanded to a graph by adding each story sentence as a vertex and calculate the correlations between each other. By doing so, the constructed graph can

Algorithm 1 Graph Construction

Input: The extracted keywords K, the story S
Output: The story graph G
Require: Edge weight calculation method λ
Tokenize keywords K and story S into words
for keyword k **do**
    Assign k to vertex v_{k_i};
end for
for sentence s **do**
    Assign s to vertex v_{s_i};
end for
for vertex v_i and $\mathrm{v}_j \in (\mathrm{v}_k \cap \mathrm{v}_s)$ **do**
    Calculate edge weight: $\mathrm{w}_{ij} = \lambda(\mathrm{v}_i, \mathrm{v}_j)$
end for

capture multi-level interconnected structural information of the story, including keyword-to-sentence and sentence-to-sentence correlations. To this end, each story has a story title, a group of keywords, and a story graph representing the intra-story structure.

Figure 2 presents an overview of our proposed GraphD-CVAE model. We first utilize the pre-constructed title-keyword pairs to train a generation model for conducting keyword planning given a story title as input. Then, we combine the story title and the corresponding keywords of each story as the input of the graph CVAE (G-CVAE) module to automatically generate a graph for each story. The content CVAE (C-CVAE) is then tasked with automatically composing each story using both the input title, storyline keywords, and intra-story structure (i.e., graph) as input. We also devise two interactions between the G-CVAE module and the C-CVAE module to facilitate generating *content-aware* graphs and *structure-sensitive* stories.

Graph-Infused Dual CVAE Model

In this section, we will elaborate on each module of our proposed model, i.e., keyword extraction, graph construction, content CVAE, and graph CVAE.

Keyword Extraction

As mentioned before, a story title $T = (w_1, w_2, w_3, ..., w_n)$ is first given as input. We then conduct storyline planning to obtain keywords $K = (k_1, k_2, k_3, ..., k_m)$ from the given title. Concretely, we follow the setting of Yao et al. (2019) to sequentially extract keywords from the existing story dataset using the RAKE algorithm (Rose et al. 2010), which combines word-frequency based and graph-based metrics to weight the importance of each keyword. The extracted pairs of titles and corresponding keywords are used to train a language model conditioned on given story titles.[1]

Graph Construction

Algorithm 1 gives the details for constructing the story graph. Specifically, given the keywords K, and the story

[1] We follow the same method of Yao et al. (2019) to extract storyline keywords.

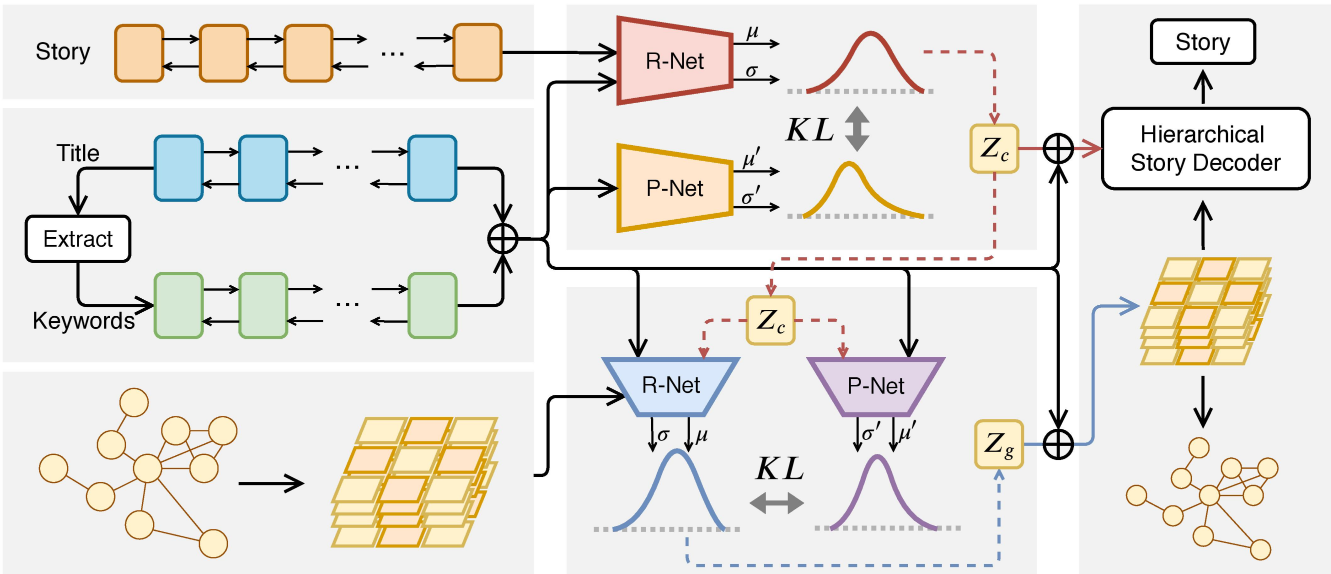


Figure 3: The detailed architecture of the Graph-Infused Dual Conditional Variational Autoencoder Model.

S, we first perform tokenization [2], and then assign each keyword k_i and sentence s_i to vertex v_{k_i} and v_{s_i}, respectively. Finally, we construct the edges between each vertex by applying the edge weight calculation method λ. In this paper, we utilize the simple and effective co-occurrence word count method for edge weight calculation by calculating the co-occurrence word count between two vertices (Li et al. 2019b; Yin et al. 2019; Liu et al. 2019). Note that this method can automatically calculate edge weight without training or learning. After the graph construction step, each input story title correlates with a graph, which captures the structural information between keywords and sentences. We then use these title-graph pairs to train the Graph CVAE module for generating the story graph during inference.

Dual CVAE

As shown in Figure 3, the Dual CVAE part consists of a content CVAE and a graph CVAE.

Content CVAE Module, is responsible for generating the story content. Following (Li et al. 2018), the CVAE works in an encoder-decoder manner. The encoder is a bi-directional recurrent neural network (Bi-RNN) (Berglund et al. 1997) with gated recurrent units (GRU) (Cho et al. 2014), which encodes the target story S and condition c into the following representation: $s = [\overrightarrow{S}, \overleftarrow{S}]$; $c = [\overrightarrow{c}, \overleftarrow{c}]$. In our case, the condition c is the concatenation of the story title T and the extracted keywords K, which is also used for the graph CVAE module. After encoding, the prior network and recognition network are introduced to approach the real prior distribution $p_\theta(z_c|c)$ and the posterior distribution $q_\phi(z_c|s,c)$. We assume the variational approximate posterior is a multivariate Gaussian N with a diagonal co-variance structure $q_\phi(z_c|s,c) = N(\mu, \sigma^2 I)$, implemented as a linear mapping

$$\begin{bmatrix} \mu \\ log(\sigma^2) \end{bmatrix} = W_q \begin{bmatrix} s \\ c \end{bmatrix} + b_q \tag{1}$$

Similarly, the prior $p_\theta(z_c|c)$ is also another multivariate Gaussian distribution $N(\mu', \sigma'^2 I)$ which is parameterized by a multilayer perceptron (MLP) with a $tanh(\cdot)$ activation function. Note that the recognition network is only applied during the training stage while the prior network is applied during the inference stage.

The decoder is a hierarchical RNN (Serban et al. 2016). The hidden state of the higher-level RNN and the lower-level RNN are calculated by,

$$\begin{aligned} h_0^{<h>} &= f(W_x[z_c, c, g]) \\ h_i^{<h>} &= f(W_x h_{i-1}^{<h>} + W_y h^{<l>}) \\ h_i^{<l>} &= f(W_x h^{<h>} + W_y h_{i-1}^{<l>} + W_z w_{i-1}) \end{aligned} \tag{2}$$

where the $h^{<h>}$ is the hidden state of the high-level RNN that guides the generation of each sentence. $h^{<l>}$ is the hidden state of the lower-level RNN for generating each word. The initial hidden state of $h_0^{<h>}$ is the concatenation of the latent variable z_c, condition c, and the output of the graph CVAE module g, which will be introduced in below.

Graph CVAE Module, generates a story graph given the story title, keywords, and pre-constructed story graph. In other words, the target of graph CVAE is to reconstruct a graph given the specified condition c, i.e., the combination of story title and keywords. To encode the story graph, we convert the graph into an adjacency matrix and uses a RNN to encode its representation, denoted as g (You et al. 2018).

Since the story graph represents the relations between keywords and each sentence, it is crucial to take into account both the keywords and story content. Accordingly, we

[2] http://www.nltk.org/

parametrize the prior network and the variational approximate posterior as $p_\vartheta(z_g|z_c, c)$ and $q_\psi(z_g|z_c, c, g)$, where z_c is the latent variable of content CVAE mentioned above. After acquiring the latent variable z_g, we then pass it to the decoder to generate the story graph. The decoder of the G-CVAE consists of a MLP with ReLU activation function and a MLP with sigmoid activation function. The output g would further be given to the C-CVAE module through a linear transformation to guide the story generation process. Through the G-CVAE module, the model learns to model the intra-story structure based on story content z_c, input title *T*, and keywords *K*.

Infusion. Given the keywords *K*, the story title *T*, the story graph g, and the story *S*, the GraphD-CVAE model jointly encodes them into latent variable z_c and z_g and further decode them to generate the corresponding story. Formally, the GraphD-CVAE models the joint probability of the encoded story *s* and the story graph *g* as below:

$$p(s,g) = \int_{d_{z_c}} \int_{d_{z_g}} p(s|g,c,z_c)p(g|z_c,z_g,c) \\ p(z_g|z_c,c)p(z_c|c)d_{z_c}d_{z_g} \tag{3}$$

where z_g is drawn based on z_c from a conditional prior network $p_\vartheta(z_g|z_c, c)$.

Both content CVAE and graph CVAE are optimized by maximizing the log-likelihood over the story *s* and the story graph *g* conditioned on *c*. Below we state the training objective of *s* and *g* in equation 4 and 5 respectively:

$$L_c(\theta,\phi;s,c) = E_{q_\phi(z_c|s,c)}[logp_\theta(s|z_c,g,c)] \\ -KL(q_\phi(z_c|s,c) \parallel p_\theta(z_c|c)) \tag{4}$$

$$L_g(\vartheta,\psi;g,c) = E_{q_\psi(z_g|g,z_c,c)}[logp_\vartheta(g|z_g,c)] \\ -KL(q_\psi(z_g|g,z_c,c) \parallel p_\vartheta(z_g|z_c,c)) \tag{5}$$

where $\theta, \vartheta, \phi, \psi$ are the parameters of the networks. Moreover, to tackle the vanishing latent variable problem, we present the bag-of-word (bow) loss (Zhao, Zhao, and Eskenazi 2017) to force the model to capture global information of the target story, written by

$$L_{bow}(\phi;s) = E_{q_\phi(z_c|s,c)}[logp(s_{bow}|z_c,g,c)] \tag{6}$$

In training, we jointly optimize the loss of the C-CVAE module, the G-CVAE module and the bow loss (equation 7):

$$L_{total} = L_c + L_g + L_{bow} \tag{7}$$

Experiment

Dataset

We train our story generation models on two widely-used datasets (May and Knight 2018; Luo et al. 2019; Wang and Wan 2019; Li et al. 2019a; Martin et al. 2018), i.e., the ROCStories dataset and the CMU Movie Summary corpus. The ROCStories dataset [3] (Mostafazadeh et al. 2016) consist of 98,163 high-quality hand-crafted stories that can capture causal and temporal commonsense relations of daily events. Each story in the ROCStories dataset consists of one story title and a five-sentence paragraph, in which the average length of each story content is approximately 43 words. The CMU Movie Summary corpus[4] (Bamman, O'Connor, and Smith 2013) extracted from Wikipedia consists of 42,306 stories that contain concise summary of the corresponding movies. Each movie title is paired with a movie summary that contains multiple sentences. The average length of the stories in the Movie Summary corpus is approximately 310 words, which is much longer than the stories in the ROCStories dataset. This allows us to evaluate the effectiveness of our model in generating either short or long stories with intra-story structural information. To conduct experiments, we split these stories into 8:1:1 for training, validation, and testing.

Baselines

We compare the performance of our proposed model with the following necessary baselines:

S2S, the standard Sequence to Sequence model (Sutskever, Vinyals, and Le 2014), which serves as the benchmark of a significant number of natural language generation tasks. In our implementation, a story title is given as the input of the model to generate the story sentence by sentence.

HCVAE, an augmented conventional Conditional Variational Autoencoder (CVAE) model with a hierarchical decoder, employed to investigate the performance of our proposed model without story planning and graphical information guidance.

Hierarchical, a strong model for story generation (Fan, Lewis, and Dauphin 2018b), which exploits a hierarchical generation pipeline and a self-attention enhanced convolutional sequence to sequence model to generate stories. We utilized this model to explore the effectiveness of integrating structural information to the model when generating long stories.

Plan and Write, is also a top-performing story generation model on the ROCStories dataset (Yao et al. 2019). The model first creates a group of storyline keywords with an input story title and then generates the story sentence by sentence given the previous extracted keywords. This baseline is utilized to explore the influence of infusing intra-story structure information for short story generation.

Draft and Edit, is also a strong story generation model (Yu et al. 2020), which exploits a multi-pass hierarchical conditional variational autoencoder model to first create a story draft then recursively polish the initial draft to generate consistent and diversified content. We utilized this baseline to explore the effectiveness of integrating structural information to hierarchical model.

D-CVAE, an ablation model, which generates stories without the keyword information. In this case, the graph will be constructed to model sentence-level structure information, i.e., each vertex refers to a sentence in the story.

[3]http://www.cs.rochester.edu/nlp/rocstories/

[4]http://www.cs.cmu.edu/~ark/personas/

Dataset	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	Distinct-1	Distinct-2	Distinct-3	Distinct-4
ROCStories	**S2S**	23.34	8.46	3.50	1.62	0.60	2.22	5.19	8.87
	HCVAE	29.41	11.20	4.37	1.85	1.88	12.33	35.03	61.02
	Plan and Write	27.49	11.65	**5.44**	**2.77**	0.82	4.62	12.36	23.55
	Draft and Edit	29.39	11.02	4.25	1.76	1.99	14.82	40.73	67.40
	D-CVAE	29.79	11.45	4.50	1.92	1.95	13.90	38.41	64.98
	Key-CVAE	29.64	11.74	4.79	2.07	2.00	15.30	42.07	68.61
	GraphD-CVAE	**30.08**	**11.88**	4.84	2.13	**2.05**	**17.56**	**46.89**	**73.87**
Movie Summary	**S2S**	17.40	7.13	2.65	1.01	0.09	0.30	0.68	1.19
	HCVAE	25.73	10.82	4.15	1.57	0.36	1.64	4.55	9.25
	Hierarchical	27.61	10.78	3.53	1.24	0.35	2.34	8.02	18.53
	Draft and Edit	24.35	9.99	3.67	1.34	0.52	2.43	6.60	13.06
	D-CVAE	**29.16**	11.97	4.40	1.62	0.59	2.89	8.10	16.31
	Key-CVAE	27.45	11.50	4.32	1.64	0.83	3.86	10.08	19.27
	GraphD-CVAE	28.90	**12.00**	**4.46**	**1.71**	**0.92**	**4.15**	**10.59**	**19.99**

Table 1: Automatic Evaluation Results: BLEU-n represents BLEU scores on n-gram (n = 1 to 4); Distinct-n denotes to the n-gram distinctness score (n = 1 to 4)

Key-CVAE, an ablation model, which is employed to compare the performance of the GraphD-CVAE model without the graph CVAE module. In this case, only the story title and keywords are provided in the story generation process (no structural information involved).

Model Settings

We use the following parameters and hyperparameters to train our model. We set the word embedding size to 300 and the graph embedding size to 32. The hidden state dimension of the graph encoder and decoder are set to 64. The hidden state dimension of the content encoder and decoder are set to 500. The latent variable size of the content CVAE and graph CVAE are set to 300 and 32, respectively. All trainable weights are initialized from a uniform distribution: $[-0.08, 0.08]$. We optimize models using the Adam optimizer (Kingma and Ba 2015) and set the mini-batch size to 80. To avoid gradient explosion, the gradient clipping strategy is applied (Pascanu, Mikolov, and Bengio 2013). We set the clipping value to 5. Finally, we train the models with a learning rate of 0.001.

Evaluation Metrics

To evaluate stories generated from different models, we utilize the following widely-used automatic metrics and human judges.

BLEU (Papineni et al. 2002), is an automatic metric that has been widely used to evaluate story generation models (Martin et al. 2018; Xu et al. 2018a; Li et al. 2018; Yao et al. 2019), which can measure the degree of overlapped words between the predicted stories and the golden stories. Although computing BLEU scores for story generation is controversial, we still present the BLEU scores to align this work with previous studies.

Distinctness, measures the wording diversity of generated stories by calculating the proportion of distinctive n-grams (Li et al. 2016).

Human Evaluation, analyzes the generated stories in the aspect of human beings. Specifically, we conduct pair-wise comparisons in a blind way. We randomly sample 100 story titles to generate stories using each generation model[5]. We then allocate two different stories with the same story title to human evaluators and ask them to mark the better story based on the three criteria: consistency (whether the story is logically consistent), wording diversity (whether the story is narrated with diversified wording), and overall preference (which story the human judges most prefers). Each story is evaluated by five human judges.

Results and Discussion

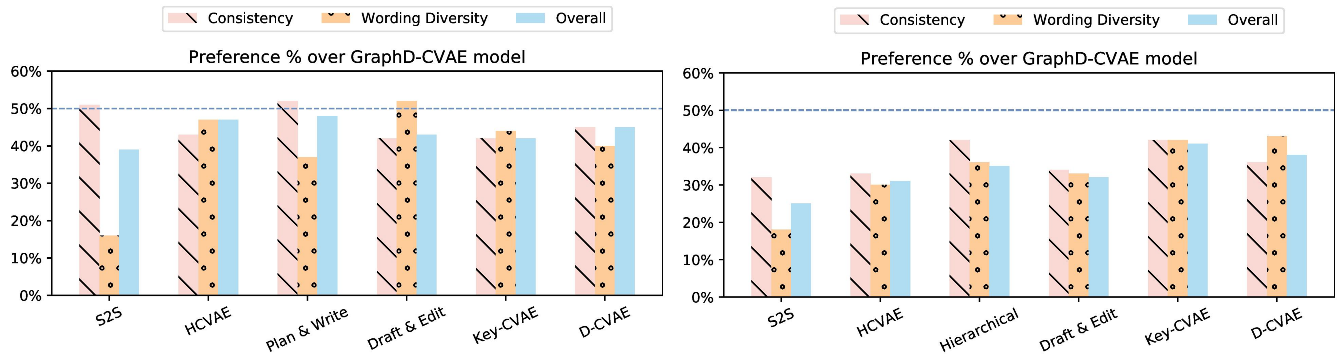
Table 1 and Figure 4 present the automatic evaluation and human evaluation results on the ROCStories and Movie Summary datasets. Figure 5 supplements a few representative story instances with intra-story structure presented. We analyze these results from the following perspectives.

Quantitative Analysis

Keyword planning effectively increases the wording diversity of the story content. Keyword planning provides the model with better storyline guidance, which forms an initial planning on the story content. Based on the results of the distinctness score in Table 1 and the wording diversity score in Figure 4, we can see that story planning can effectively improve the diversity of the generated stories.

Moreover, as shown in Table 1, we found that conventional keyword planning method is crucial for short story generation, since it can provide useful information during the generation process, however, when the story content gets longer, we found that stoyline information would be insufficient and even leads to keyword vanishing problem (i.e. the

[5]For fairness, we trimmed all generated stories to 200 words.



(a) ROCStories

(b) Movie Summary

Figure 4: Human evaluation results on three criteria, i.e., consistency, wording diversity, and overall preference.

model tends to ignores keyword information when generating long-length content).

Fusing structural information increases the consistency of the story. Introducing structural information of the story allows the model to learn the interrelation between keywords and sentences. Based on the results of the BLEU score, the correlation score in Table 1, and the consistency score in Figure 4, we can observe that infusing graphical information allows our model to generate much more consistent stories. However, we also found that short stories have better consistency compared to the long ones. This might be due to the fact that long stories require more complicated intra-story structure modeling compared to the short stories, while structurized keyword-content planning may be insufficient. This points out the future direction for generating longer stories with other structural information modeling.

Integrating D-CVAE with keyword planning can generate an overall better story. Based on the results in Figure 4, we observe that in short story generation, our model outperforms the hierarchical based methods in both the consistency and wording diversity criteria. However, we found that the consistency score is slightly lower when comparing to the S2S and plan and write model. The reason is that these models tend to generate repetitive words, which from a human perspective, would regard these stories consistent. However, because of the high repeatability content, it jeopardizes the wording diversity of the story, confirmed by the relatively low wording diversity score in the human evaluation results. This results in a higher overall preference percentage of our proposed model. On the other side, for long story generation, our model has outperformed all baseline models in the consistency, wording diversity, and overall preference criteria, showing that structuring story content is crucial for long story generation. To sum up, with prior keyword planning and story structure information fusion, our model can generate overall human preferable stories.

Case Study and Future Work

Figure 5 shows representative short and long stories generated by our proposed model. As presented in Figure 5, by looking at the intra-story structure of each story[6], we could observe that through fusing graphical information, it allows the model to not only learn the relations between keywords but also consider the linkage between sentences. With such information, the model could better model the intrinsic story structure and generate structurized story content instead of a plain left-to-right narrative.

Owing to the promising results of modeling structural information along with keyword planning for story generation, we could further consider introducing more complicated intra-story structure modeling such as knowledge graph, character graph and action graph. Moreover, since our proposed model degenerates story generation process into different sub-problems, we could also apply better extensive graph processing, story planning methods and large pre-trained models under the framework for both short and long story generation in the future.

Related Work

Neural Storytelling and Text Generation

As one of the tasks of text generation (Chen et al. 2018, 2019; Chan et al. 2019; Li et al. 2020; Gao et al. 2020), automatic storytelling requires to generate a sequence of sentences with logically connected contents throughout a passage. Early story generation methods mainly focus on planning-based or case-based approaches (Manshadi, Swanson, and Gordon 2008; Riedl and Young 2010; Gervás et al. 2004; Montfort, Marcus, and Prince 2007), which are restricted to domain knowledge. To extend domain-specific models to unknown domains, Li et al. (2013) proposed to generate stories with crowdsourced plot graphs. Recently, researchers focused on generating stories with storyline or intermediate representations, e.g., short descriptions (Jain et al. 2017), events (Martin et al. 2018), prompts (Fan, Lewis, and Dauphin 2018a), skeletons (Xu et al. 2018a), keywords (Yao et al. 2019), to enhance the consistency of the generated content. Li et al. (2019a) presented a cache-

[6]Blue nodes denotes to each sentence of the story; Grey nodes denotes to the extracted keywords.

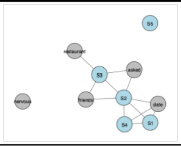
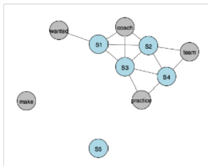
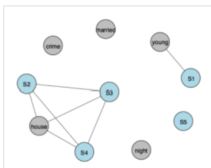
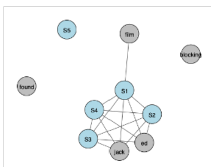
	Title	Structure	Story Content
ROC	**The Date**		Bob never had a date. He asked his friends to be a date. He asked his friends to go to a restaurant. Jim agreed to the date. The date went great and Jim had a great time.
	The Team Decision		Ryan wanted to be a professional basketball coach. His coach told him to try out for the team. The coach told him he needed to practice the day. He practiced hard and practiced hard for the team. He's coach decided to make his team the team.
Movie Summary	**Kitty Foiled**		After being released from prison, a young woman named Sam Smes , who is now a widow , is sent to a hospital for a few weeks . She is then picked up by a man named john. When she finds out that her father is, he tells her that he is going to be a good man. When she returns to the house, she finds that the man is a woman. She tells him that she is the daughter of a man who has been in the the police. After the wedding, the police arrive at the house and find a dead woman's body. The next day, the police arrive at the house and find a dead woman's body [..]
	Jack		The film tells the story of jack jack, a young american woman who is institutionalized in the early 1960s. Jack's former girlfriend sarah kelly is a recovering alcoholic who has been accused of murder by the murder of jack jack, who is a former confederate attorney. Jack's wife sarah and jack jack are in the midst of a murder. Sarah's father, jack, is killed in the murder of jack's wife sarah, who was killed by a group of zombies […]

Figure 5: Short and long stories generated by our proposed model.

augmented conditional variational autoencoder model to generate stories with wording diversity and thematic consistency. Fan, Lewis, and Dauphin (2019) decomposed story generation into a series of easier generation task to capture high-level interactions between the plot points. Yu et al. (2020) proposed a multi-pass hierarchical CVAE model to first create a story draft and then recursively polish the draft to generate consistent and diversified stories. Liu et al. (2020) presented a character-centric storytelling model that decomposes the story generation process into two-steps, including action prediction and sentence generation to provide fine-grained control on the story generation process. Unlike previous research, we explicitly capture intra-story structure information to generate better stories.

Graph-based Text Generation Network

Recently, Graph Neural Networks (GNN) has attracted growing attention to managing graph-structure data and have achieved impressive performance on many natural language generation tasks. Xu et al. (2018b) proposed to use a graph-based model to encode SQL queries in the SQL-to-Text task. Hu et al. (2019) utilized a graph to represent the relations between different speakers in multi-party dialogue generation. Li et al. (2019b) drew the graph from long Chinese articles and generated coherent comments with structure-aware content. Yin et al. (2019) proposed a graph-based neural sentence ordering model to learn the semantic representations of sentences in long texts. Koncel-Kedziorski et al. (2019) further adopt GNN to capture the relational structure and generate text without imposing linearization or hierarchical generation constraints. Hsu et al. (2020) leveraged external knowledge graph for visual storytelling. Different from the aforementioned models, we mainly zero in on generating graphs from a continuous latent space to model intra-story structures and infuse structure information in the story generation process.

Conclusion

In this paper, we proposed a graph-infused dual conditional varitional autoencoder model to generate both *content-aware* and *structure-aware* stories. To capture intra-story structural information, we first generate storyline keywords given the story title and further extend the keywords and content to form an intra-story structure. The obtained intra-story structure is then infused to the story content generation process via the dual connection between the graph CVAE module and the content CVAE module. Through decomposing story generation into intra-story structure learning and content modeling, our model can generate better stories. Pair-wise comparisons on both short and long story corpora confirm that human judges prefer stories generated by our model.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2020AAA0106600, No. 2020YFB1406702), the National Science Foundation of China (NSFC No. 61876196, No. 61672058, and No.62076173), Beijing Outstanding Young Scientist Program No. BJJWZYJH012019100020098. Rui Yan is supported as a young fellow at Beijing Academy of Artificial Intelligence (BAAI).

References

Bamman, D.; O'Connor, B.; and Smith, N. A. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 352–361.

Berglund, M.; Raiko, T.; Honkala, M.; Kärkkäinen, L.; Vetek, A.; and Karhunen, J. 1997. Bidirectional Recurrent Neural Networks as Generative Models - Reconstructing Gaps in Time Series. *IEEE Transactions on Signal Processing* .

Chan, Z.; Chen, X.; Wang, Y.; Li, J.; Zhang, Z.; Gai, K.; Zhao, D.; and Yan, R. 2019. Stick to the Facts: Learning towards a Fidelity-oriented E-Commerce Product Description Generation. In *EMNLP*, 4960–4969.

Chen, X.; Chan, Z.; Gao, S.; Yu, M.-H.; Zhao, D.; and Yan, R. 2019. Learning towards Abstractive Timeline Summarization. In *IJCAI*, 4939–4945.

Chen, X.; Gao, S.; Tao, C.; Song, Y.; Zhao, D.; and Yan, R. 2018. Iterative Document Representation Learning Towards Summarization with Polishing. In *EMNLP*, 4088–4097.

Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*.

Fan, A.; Lewis, M.; and Dauphin, Y. 2018a. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 889–898.

Fan, A.; Lewis, M.; and Dauphin, Y. 2018b. Hierarchical neural story generation. In *ACL*.

Fan, A.; Lewis, M.; and Dauphin, Y. 2019. Strategies for Structuring Story Generation. *ACL* .

Gao, S.; Chen, X.; Liu, L.; Zhao, D.; and Yan, R. 2020. Learning to Respond with Your Favorite Stickers: A Framework of Unifying Multi-Modality and User Preference in Multi-Turn Dialog. *ACM Trans. Inf. Syst.* .

Gervás, P.; Díaz-Agudo, B.; Peinado, F.; and Hervás, R. 2004. Story plot generation based on CBR. In *ICIAI*.

Goldfarb-Tarrant, S.; Feng, H.; and Peng, N. 2019. Plan, Write, and Revise: an Interactive System for Open-Domain Story Generation. *NAACL HLT 2019* 89.

Hsu, C.-C.; Chen, Z.-Y.; Hsu, C.-Y.; Li, C.-C.; Lin, T.-Y.; Huang, T.-H.; and Ku, L.-W. 2020. Knowledge-Enriched Visual Storytelling. *AAAI* .

Hu, W.; Chan, Z.; Liu, B.; Zhao, D.; Ma, J.; and Yan, R. 2019. GSN: A Graph-Structured Network for Multi-Party Dialogues. In *IJCAI*, 5010–5016.

Jain, P.; Agrawal, P.; Mishra, A.; Sukhwani, M.; Laha, A.; and Sankaranarayanan, K. 2017. Story Generation from Sequence of Independent Short Descriptions. *SIGKDD Workshop on Machine Learning for Creativity* .

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. *ICLR* .

Kipf, T. N.; and Welling, M. 2016. Variational Graph Auto-Encoders. *Stat* 1050: 21.

Koncel-Kedziorski, R.; Bekal, D.; Luan, Y.; Lapata, M.; and Hajishirzi, H. 2019. Text Generation from Knowledge Graphs with Graph Transformers. *NAACL* .

Li, B.; Lee-Urban, S.; Johnston, G.; and Riedl, M. 2013. Story generation with crowdsourced plot graphs. In *AAAI*.

Li, J.; Bing, L.; Qiu, L.; Chen, D.; Zhao, D.; and Yan, R. 2019a. Learning to write stories with thematic consistency and wording novelty. In *AAAI*.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of NAACL-HLT*, 110–119.

Li, J.; Song, Y.; Haisong, Z.; Chen, D.; Shi, S.; Zhao, D.; and Yan, R. 2018. Generating classical chinese poems via conditional variational autoencoder and adversarial training. In *EMNLP*.

Li, M.; Chen, X.; Gao, S.; Chan, Z.; Zhao, D.; and Yan, R. 2020. VMSMO: Learning to Generate Multimodal Summary for Video-based News Articles. In *EMNLP*, 9360–9369.

Li, W.; Xu, J.; He, Y.; Yan, S.; Wu, Y.; and Sun, X. 2019b. Coherent Comments Generation for Chinese Articles with a Graph-to-Sequence Model. In *ACL*, 4843–4852. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1479. URL https://www.aclweb.org/anthology/P19-1479.

Liu, B.; Zhang, T.; Niu, D.; Lin, J.; Lai, K.; and Xu, Y. 2019. Matching Long Text Documents via Graph Convolutional Networks. *ACL* .

Liu, D.; Li, J.; Yu, M.-H.; Huang, Z.; Liu, G.; Zhao, D.; and Yan, R. 2020. A Character-Centric Neural Model for Automated Story Generation. In *AAAI*.

Luo, F.; Dai, D.; Yang, P.; Liu, T.; Chang, B.; Sui, Z.; and Sun, X. 2019. Learning to control the fine-grained sentiment for story ending generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6020–6026.

Manshadi, M.; Swanson, R.; and Gordon, A. S. 2008. Learning a Probabilistic Model of Event Sequences from Internet Weblog Stories. In *FLAIRS Conference*, 159–164.

Martin, L. J.; Ammanabrolu, P.; Wang, X.; Hancock, W.; Singh, S.; Harrison, B.; and Riedl, M. O. 2018. Event representations for automated story generation with deep neural nets. In *AAAI*.

May, N. P. M. G. J.; and Knight, K. 2018. Towards controllable story generation. *NAACL HLT 2018* 43.

Montfort, N.; Marcus, M. P.; and Prince, G. 2007. *Generating narrative variation in interactive fiction*. University of Pennsylvania Philadelphia, PA.

Mostafazadeh, N.; Chambers, N.; He, X.; Parikh, D.; Batra, D.; Vanderwende, L.; Kohli, P.; and Allen, J. 2016. A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories. In *NAACL*, 839–849. San Diego, California: Association for Computational Linguistics. doi:10.18653/v1/N16-1098. URL https://www.aclweb.org/anthology/N16-1098.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Pascanu, R.; Mikolov, T.; and Bengio, Y. 2013. On the difficulty of training recurrent neural networks. In *ICML*.

Riedl, M. O.; and Young, R. M. 2010. Narrative planning: Balancing plot and character. *JAIR* .

Rose, S.; Engel, D.; Cramer, N.; and Cowley, W. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory* 1: 1–20.

Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A.; and Pineau, J. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Sohn, K.; Lee, H.; and Yan, X. 2015. Learning Structured Output Representation using Deep Conditional Generative Models. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28*, 3483–3491. Curran Associates, Inc. URL http://papers.neurips.cc/paper/5775-learning-structured-output-representation-using-deep-conditional-generative-models.pdf.

Sutskever, I.; Vinyals, O.; and Le, Q. 2014. Sequence to sequence learning with neural networks. *NeurIPS* .

Tambwekar, P.; Dhuliawala, M.; Martin, L. J.; Mehta, A.; Harrison, B.; and Riedl, M. O. 2019. Controllable Neural Story Plot Generation via Reward Shaping. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 7. URL https://www.ijcai.org/proceedings/2019/829.

Wang, T.; and Wan, X. 2019. T-CVAE: Transformer-based conditioned variational autoencoder for story completion. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 5233–5239. AAAI Press.

Xu, J.; Ren, X.; Zhang, Y.; Zeng, Q.; Cai, X.; and Sun, X. 2018a. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *EMNLP*.

Xu, K.; Wu, L.; Wang, Z.; Feng, Y.; and Sheinin, V. 2018b. SQL-to-Text Generation with Graph-to-Sequence Model. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 931–936. Brussels, Belgium: Association for Computational Linguistics. doi:10.18653/v1/D18-1112. URL https://www.aclweb.org/anthology/D18-1112.

Yao, L.; Peng, N.; Weischedel, R.; Knight, K.; Zhao, D.; and Yan, R. 2019. Plan-and-write: Towards better automatic storytelling. In *AAAI*.

Yin, Y.; Song, L.; Su, J.; Zeng, J.; Zhou, C.; and Luo, J. 2019. Graph-based neural sentence ordering. In *Proceedings of IJCAI*.

You, J.; Ying, R.; Ren, X.; Hamilton, W. L.; and Leskovec, J. 2018. GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models. *ICML* .

Yu, M.-H.; Li, J.; Liu, D.; Tang, B.; Zhang, H.; Zhao, D.; and Yan, R. 2020. Draft and Edit: Automatic Storytelling Through Multi-Pass Hierarchical Conditional Variational Autoencoder. In *AAAI*.

Zhang, X.; Yang, Y.; Yuan, S.; Shen, D.; and Carin, L. 2019. Syntax-Infused Variational Autoencoder for Text Generation. In *ACL*, 2069–2078. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1199. URL https://www.aclweb.org/anthology/P19-1199.

Zhao, T.; Zhao, R.; and Eskenazi, M. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL*.