

A Skeleton-Based Model for Promoting Coherence Among Sentences in Narrative Story Generation

Jingjing Xu¹, Xuancheng Ren¹, Yi Zhang¹, Qi Zeng¹, Xiaoyan Cai², Xu Sun¹

¹MOE Key Lab of Computational Linguistics, School of EECS, Peking University

²School of Automation, Northwestern Polytechnical University

{jingjingxu, renxc, zhangyi16, pkuzengqi, xusun}@pku.edu.cn
xiaoyanc@nwpu.edu.cn

Abstract

Narrative story generation is a challenging problem because it demands the generated sentences with tight semantic connections, which has not been well studied by most existing generative models. To address this problem, we propose a skeleton-based model to promote the coherence of generated stories. Different from traditional models that generate a complete sentence at a stroke, the proposed model first generates the most critical phrases, called skeleton, and then expands the skeleton to a complete and fluent sentence. The skeleton is not manually defined, but learned by a reinforcement learning method. Compared to the state-of-the-art models, our skeleton-based model can generate significantly more coherent text according to human evaluation and automatic evaluation. The G-score is improved by 20.1% in human evaluation.¹

1 Introduction

We focus on the problem of narrative story generation, a special kind of story generation (Li et al., 2013). It requires systems to generate a narrative story based on a short description of a scene or an event, as shown in Table 1. In general, a narrative story is described with several inter-related scenes. Different from traditional text generation tasks, this task is more challenging because it demands the generated sentences with tight semantic connections. Currently, most state-of-the-art approaches (Jain et al., 2017; Liu et al., 2017; Fan et al., 2018; Ma et al., 2018a; Xu et al., 2018b) are largely based on Sequence-to-Sequence (Seq2Seq) models (Sutskever et al., 2014), which generate a sentence at a stroke in a left-to-right manner.

However, we find it hard for these approaches to model the semantic dependency among sentences,

¹The code is available at <https://github.com/lancopku/Skeleton-Based-Generation-Model>

Task Description
Input: A short description of a scene or an event. Output: A relevant narrative story following the input.
Examples
Input: <i>Fans came together to celebrate the opening of a new studio for an artist.</i> Output: <i>The artist provided champagne in flutes for everyone. Friends toasted and cheered the artist as she opened her new studio.</i>
Input: <i>Last week I attended a wedding for the first time.</i> Output: <i>There were a lot of families there. They were all taking pictures together. Everyone was very happy. The bride and groom got to ride in a limo that they rented.</i>

Table 1: An illustration of narrative story generation.

which causes low-quality generated stories where the scenes are irrelevant. In fact, as shown in Figure 1, we observe that the connection among sentences is mainly reflected through key phrases, such as predicates, subjects, objects and so on. In this work, we regard the phrases that express the key meanings of a sentence as a skeleton. The other words (e.g., modifiers) not only are redundant for understanding semantic dependency, but also make the dependency sparse. Therefore, generating all information at a stroke makes it difficult to learn the dependency of sentences. In contrast, the sentences written by humans are closely tied and the whole story is more coherent and fluent. It is mainly attributed to the way of human writing where we often first come up with a skeleton and then reorganize them into a fluent sentence.

Therefore, motivated by the way of human writing, we propose a skeleton-based model to improve the coherence of generated text. The key idea is to first generate a skeleton and then expand the skeleton to a complete sentence. As a simplified sentence representation, the skeleton can help machines learn the dependency of sentences by avoiding the interference of irrelevant information. Our model contains two parts: a skeleton-

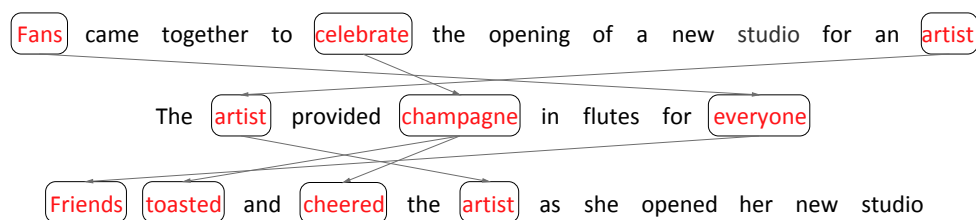


Figure 1: The semantic dependency among sentences in a narrative story. It can be seen that the connection among sentences is mainly reflected through key phrases (shown in red). In this work, we regard such key phrases as a skeleton.

based generative module and a skeleton extraction module.

The generative module consists of an input-to-skeleton component and a skeleton-to-sentence component. The input-to-skeleton component learns to associate inputs and skeletons, and the skeleton-to-sentence component learns to expand a skeleton to a sentence. In our model, a good skeleton that can capture key semantic information is a critical supervisory signal.

The skeleton extraction module is used to generate sentence skeletons. In real-world datasets, the human-annotated skeleton is usually unavailable. In addition, it is difficult to define the unified rules of extracting skeletons, because different sentences have different focuses. To address this problem, we build a skeleton extraction module to automatically explore sentence skeletons. Considering the discrete choice of skeleton words causes the loss function to be non-differentiable, we use a reinforcement learning method to build the connection between the skeleton extraction module and the generative module.

Our contributions are listed as follows:

- A skeleton-based model is proposed to promote the coherence of generated stories.
- The proposed model contains a skeleton-based generative module and a skeleton extraction module. Two modules are connected by a reinforcement learning method to automatically explore sentence skeletons.
- The experimental results on automatic evaluation and human evaluation show that our model can generate significantly more coherent text compared to the state-of-the-art models.

2 Related Work

Strictly speaking, the story generation task requires systems to generate a story from scratch

without any external materials. However, for simplification, many existing story generation models rely on their given materials, such as short text descriptions (Harrison et al., 2017; Jain et al., 2017), visual images (Charles et al., 2001; Huang et al., 2016), and so on. Different from these studies, we get rid of external materials and consider the complete story generation task (McIntyre and Lapata, 2009). For this task, the widely used models are based on Seq2Seq models. However, although they can generate a fluent sentence (Xu et al., 2018a), these models still perform badly on generating inter-related sentences, which are necessary for a coherent story.

To address this problem, there are several models that build the mid-level sentence semantic representation to simplify the dependency among sentences. Clark et al. (2018) extract the entities in sentences, and combine the entity context and text context together when generating a target sentence. Cao et al. (2018) encode the words with specific pre-defined dependency labels to a mid-level sentence representation. Martin et al. (2018) use additional knowledge bases to get a generalized sentence representation. Ma et al. (2018b) use the bag-of-words which occur in all references as a representation of the correct translation. Luo et al. (2018) propose to use two auto-encoders to learn the semantic representation of utterance in dialogue. However, although these models reduce the dependency sparsity to some extent, the unified rules are non-flexible and tend to generate oversimplified representations, resulting in the loss of key information.

Different from these models, we use a reinforcement learning method to automatically extract sentence skeletons for simplifying the dependency of sentences, rather than manual rules. Therefore, our proposed skeleton-based model is more flexible and can adaptively determine the appropriate granularity of sentence representations for a balance between keeping key semantics and

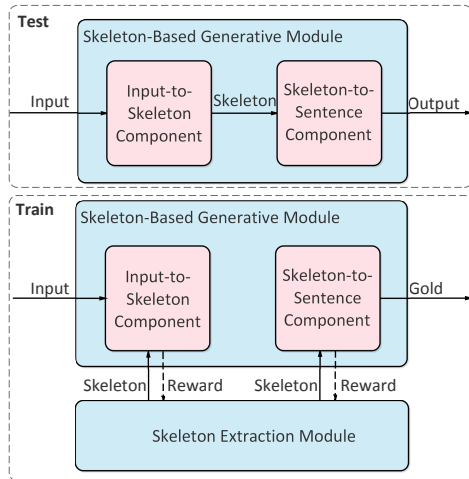


Figure 2: An illustration of the proposed model. Top: Testing. Bottom: Training. The “input” means the existing text, including the source input and the already generated text. The “skeleton” means the skeleton of the output. The skeleton extraction module first extracts skeletons from gold outputs. The pairs of input-skeleton and skeleton-gold are used to train the input-to-skeleton component and the skeleton-to-sentence component. In return, the generative module can be used to evaluate the quality of extracted skeletons. Therefore, we use the feedback of the generative module to reward extracted skeletons. By cooperation, the two modules can promote each other until convergence.

simplifying sentence representations.

3 Skeleton-Based Model

An overview of our proposed skeleton-based model is presented in Section 3.1. The details of the skeleton-based generative module and the skeleton extraction module are shown in Section 3.2 and Section 3.3. The reinforcement learning method is explained in Section 3.4.

3.1 Overview

As shown in Figure 2, our model consists of two parts, a skeleton-based generative module G_ϕ and a skeleton extraction module E_γ . The generative module consists of an input-to-skeleton component and a skeleton-to-sentence component.

The generative module generates a story sentence by sentence. When decoding a sentence, the input-to-skeleton component first generates a skeleton based on the existing text, including the source input and the already generated text, and then the skeleton-to-sentence component expands and reorganizes the skeleton into a complete sentence. We keep running this process until the gen-

erative module generates an ending symbol.

In the training process, we first use a weakly supervised method to assign the skeleton extraction module with initial extraction ability. Then, we use extracted skeletons to train the input-to-skeleton component and the skeleton-to-sentence component. In return, the generative module can be used to evaluate the quality of extracted skeletons. Therefore, we use the feedback of the generative module to reward extracted skeletons. The reward refines the skeleton extraction module. The improved skeleton extraction module further enhances the generative module. By cooperation, the two modules can promote each other until convergence.

3.2 Skeleton-Based Generative Module

The skeleton-based generative module G_ϕ consists of two parts: an input-to-skeleton component Q_α and a skeleton-to-sentence component D_θ .

3.2.1 Input-to-Skeleton Component

The input-to-skeleton component Q_α builds on a Seq2Seq structure with a hierarchical encoder (Li et al., 2015) and an attention-based decoder (Bahdanau et al., 2014). It is responsible for learning the dependency between inputs and skeletons. In the encoding process, we first obtain sentence representations via a word-level Long Short Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997), and then generate a compressed vector h via a sentence-level LSTM network. Finally, given the compressed vector h , the attention-based decoder is responsible for imagining a skeleton.

Given the training pair of input c and skeleton $s = \{s_1, \dots, s_i, \dots, s_T\}$, the cross-entropy loss is computed as

$$L_\alpha = - \sum_{i=1}^T P_Q(s_i | c, \alpha) \quad (1)$$

where α refers to the parameters of the input-to-skeleton component. The skeleton s is extracted by the skeleton extraction module. The extracting details will be introduced in Section 3.3.

3.2.2 Skeleton-to-Sentence Component

The skeleton-to-sentence component D_θ builds on a Seq2Seq structure. Both the encoder and the decoder are one-layer LSTM networks with the attention mechanism. Given a skeleton s , the encoder first generates a compressed representation,

which is then used to generate a detailed and polished sentence via the decoder.

Given the training pair of skeleton \mathbf{s} and target sentence $\mathbf{y} = \{y_1, \dots, y_i, \dots, y_M\}$, the cross-entropy loss is computed as

$$L_\theta = - \sum_{i=1}^M P_D(y_i | \mathbf{s}, \theta) \quad (2)$$

where θ refers to the parameters of the skeleton-to-sentence component.

3.3 Skeleton Extraction Module

Given a sentence \mathbf{x} , the skeleton extraction module E_γ is responsible for extracting its skeleton that only preserves the key information. Specially, we use the Seq2Seq model with the attention mechanism as the implementation. Both the encoder and the decoder are based on LSTM structures.

Since the extracted skeletons are treated as supervisory signals for the generative module, the extraction ability needs to be initialized. To pre-train the skeleton extraction module, we propose a weakly supervised method. We reformulate skeleton extraction as a sentence compression problem and use a sentence compression dataset to train this module. In sentence compression, the compressed sentence is required to be grammatical and convey the most important information. From the aspect of keeping important information, the sentence compression dataset can be used to help the training of the skeleton extraction module. However, since the style of the sentence compression dataset is very different from that of the narrative story dataset, it is difficult for the pre-trained module to give narrative text accurate compression results. Therefore, the supervisory signals are noisy and need to be further improved.

Given the training pair of the original text \mathbf{x} and the compressed version $\mathbf{s} = \{s_1, \dots, s_i, \dots, s_T\}$, we use the following cross-entropy loss to pre-train E_γ :

$$L_\gamma = - \sum_{i=1}^T P_E(s_i | \mathbf{x}, \gamma) \quad (3)$$

where γ is the parameters of the skeleton extraction module.

3.4 Reinforcement Learning Method

We propose a reinforcement learning method to build the connection between the skeleton ex-

Algorithm 1 The reinforcement learning method for training the generative module G_ϕ and the skeleton extraction module E_γ .

- 1: Initialize the generative module G_ϕ and the skeleton extraction module E_γ with random weights ϕ, γ
 - 2: Pre-train E_γ using MLE based on Eq. 3
 - 3: **for** each iteration $j = 1, 2, \dots, J$ **do**
 - 4: Generate a skeleton \mathbf{s}_j based on E_γ
 - 5: Given \mathbf{s}_j , train G_ϕ based on Eq. 1 and Eq. 2.
 - 6: Compute the reward R_c based on Eq. 5
 - 7: Compute the gradient of E_γ based on Eq. 4
 - 8: Update the model parameter γ
 - 9: **end for**
-

traction module and the skeleton-based generative module for exploring better skeletons. The detailed training process is shown in Algorithm 1.

Due to the discrete choice of words in skeletons, the loss is no longer differentiable over the skeleton extraction module. Therefore, we use policy gradient (Sutton et al., 1999) to train the skeleton extraction module.

First, we calculate a reward R_c based on the feedback of the generative module. The details of calculation process will be introduced in Section 3.4.1. Then, we optimize the parameters through policy gradient by maximizing the expected reward to train the skeleton extraction module. According to the policy gradient theorem, the gradient for the skeleton extraction module is

$$\nabla J(\gamma) = \mathbb{E}[R_c \cdot \nabla \log(P_E(\mathbf{s} | \mathbf{x}), \gamma)] \quad (4)$$

where \mathbf{x} is the original sentence, \mathbf{s} is the skeleton generated by a sampling mechanism.

3.4.1 Reward

To design an appropriate rewarding function, there is a critical question that needs to be considered: what will good/bad skeletons bring to the generative module.

We first define what is a good (or bad) skeleton. A good skeleton is expected to contain all key information and ignore other information. In contrast, the skeletons that contain too much detailed information or lack necessary information are considered as bad skeletons and should be punished. For ease of analysis, we classify possible scenarios into three categories: good skeletons, incomplete skeletons, and redundant skeletons.

If a skeleton contains too little information, it will get harder for the skeleton-to-sentence component to reconstruct the original sentence based on the skeleton. Therefore, the cross-entropy loss

of this example will be higher compared with other skeleton-sentence pairs.

If a skeleton contains too much redundant information, the input-skeleton relation will be sparse. Therefore, the cross-entropy loss of this example will be higher compared with other input-skeleton pairs.

For a good skeleton that contains an appropriate amount of information, it will benefit the two components and will get balanced losses from the two components.

Therefore, to encourage good skeletons and punish bad skeletons, we use the multiplication of the cross-entropy loss in the input-to-skeleton component and that in the skeleton-to-sentence component as the reward:

$$R_c = [K - (R_1 \times R_2)^{\frac{1}{2}}] \quad (5)$$

where K is the upper bound of the reward, R_1 and R_2 are cross-entropy losses in the input-to-skeleton component and the skeleton-to-sentence component, respectively. Only if two components both output lower cross-entropy losses, the extracted skeleton can be rewarded.

4 Experiment

In this section, we evaluate our model on a narrative story generation dataset. We first introduce the dataset, the training details, the baselines, and the evaluation metrics. Then, we compare our model with the state-of-the-art models. Finally, we show the experimental results and provide the detailed analysis.

4.1 Dataset

We use the recently introduced visual storytelling dataset (Huang et al., 2016) in our experiments. This dataset contains the pairs of photo sequences and the associated coherent narrative of events through time written by humans. We only use the text data for our experiments. In our version of narrative story generation, the model should generate a coherent story based on a given sentence. We build a new dataset for this task by splitting the data into two parts. In each story, we take the first sentence as the input text, and the following sentences as the target text. The processed dataset contains 40153, 4990, and 5054 stories for training, validation, and testing, respectively. The maximum number of sentences in each story is 6. In

total, the number of training sentences is over 20K and the number of training words is over 2M.

To pre-train the skeleton extraction module, we use a sentence compression dataset (Filippova and Altun, 2013). In this dataset, every compression is a subsequence of tokens from the input. The dataset contains 16999, 1000, and 1998 pairs for training, validation, and testing, respectively.

4.2 Baselines

We compare our proposed model with the following the state-of-the-art models.

Entity-Enhanced Seq2Seq Model (EE-Seq2Seq) (Clark et al., 2018). It regards entities as important context needed for coherent stories. When decoding a sentence, it combines entity context and text context together to reduce dependency sparsity.

Dependency-Tree Enhanced Seq2Seq Model (DE-Seq2Seq) (Cao et al., 2018). It defines some manual rules based on dependency parsing labels to find a simplified sentence representation. Following this work, we treat the extracted words based on the predefined rules as the skeleton.

Generalized-Template Enhanced Seq2Seq Model (GE-Seq2Seq) (Martin et al., 2018). It takes advantages of existing knowledge bases to get a generalized sentence representation. Following this work, we treat the generalized sentence representation as the skeleton.

4.3 Training Details

For narrative story generation, we set the number of generated sentences to 6 with the maximum length of 40 words for each generated sentence. Based on the performance on the validation set, we set the hidden size to 128, embedding size to 50, vocabulary size to 20K, and batch size to 10 for the proposed model and the state-of-the-art models. We use the Adagrad (Duchi et al., 2011) optimizer with the initial learning rate 0.6. All of the gradients are clipped when the norm exceeds 2. Both the generative module and skeleton extractive module are pre-trained for 30 and 40 epochs before reinforcement learning. The K in Equation 5 is set to 1. Due to the lack of annotated entities and dependency parsing labels, we use a popular natural language processing toolkit, Spacy², to extract entities and dependency parsing labels in the EE-Seq2Seq and DE-Seq2Seq models.

²<https://spacy.io/>

Models	BLEU
EE-Seq2Seq	0.0029
DE-Seq2Seq	0.0027
GE-Seq2Seq	0.0022
Proposed Model	0.0042 (+44.8%)

Table 2: Automatic evaluations of the proposed model and the state-of-the-art models.

4.4 Evaluation Metrics

We conduct two kinds of evaluations in this work, automatic evaluation and human evaluation. The details of evaluation metrics are shown as follows.

4.4.1 Automatic Evaluation

Following the previous work (Li et al., 2016; Martin et al., 2018), we use the BLEU score to measure the quality of generated text. BLEU (Papineni et al., 2002) is originally designed to automatically judge the machine translation quality. The key point is to compare the similarity between the results created by the machine and the references provided by the human. Currently, it is widely used in many generation tasks, such as dialogue generation, story generation, summarization, and so on. For precise results, we remove all stop words, like “the”, “a”, before computing BLEU scores.

4.4.2 Human Evaluation

Although the quantitative evaluation generally indicates the quality of generated stories, it can not accurately evaluate the generated text. Therefore, we also perform a human evaluation on the test set. We randomly choose 100 items for human evaluation. Each item contains the stories generated by different models given the same source sentence. The items are distributed to the annotators who have no knowledge about which model the story is from. It is important to note that all the annotators have linguistic background. They are asked to score the generated stories in terms of fluency and coherence. Fluency represents whether each sentence in the generated story is correct in grammar. Coherence evaluates whether the generated story is coherent. The score ranges from 1 to 10 (1 is very bad and 10 is very good). To evaluate the overall performance, we use the geometric mean of fluency and coherence as an evaluation metric.

4.5 Experimental Results

Table 2 shows the results of automatic evaluation. The proposed model performs the best ac-

Models	Fluency	Coherence	G-Score
EE-Seq2Seq	6.28	5.14	5.68
DE-Seq2Seq	8.48	3.54	5.48
GE-Seq2Seq	9.48	3.58	5.82
Proposed Model	8.69	5.62	6.99 (+20.1%)

Table 3: Human evaluations of the proposed model and the state-of-the-art models.

ording to BLEU. In particular, the differences between the existing state-of-the-art models are within 0.07, while the proposed model supersedes the best of them by 0.13.

As we previously explained, the best evaluation for narrative story generation is human evaluation. The human evaluation results are listed in Table 3.³

As for fluency, the proposed model receives the score of 8.69, second to the GE-Seq2Seq model. It is expected that the generalized templates can constrain the search space in generation and the model achieves higher fluency by loss of expressive power. In particular, we find that only 0.48%, 1.01%, and 1.20% of the unigrams, bigrams, and trigrams are unique in the stories generated by the GE-Seq2Seq model, while the percentages are 3.16%, 15.33%, and 29.67% in the stories generated by our proposed model. Nonetheless, the proposed model outperforms the other two existing models by a substantial margin. In terms of coherence, the proposed model is better than all the existing models. We need to point out that the GE-Seq2Seq model is scored the lowest in coherence, while highest in fluency. It indicates that the GE-Seq2Seq model does not learn the dependency among sentences effectively, which results from the constraint of the templates. It also needs to be noted that the models are all scored below 6 in coherence, meaning that there is still a long way to go before the generated stories satisfy the requirement of humans. Overall, the proposed model is arguably better than the existing models in that it achieves a balance between coherence and fluency, with a G-score improvement of 20.1%.

Table 4 presents the examples generated by different models. Compared with the existing models, the sentences generated by our proposed model are connected more logically. For the EE-Seq2Seq model, while it connects *park* with *plants*

³The inter-annotator agreement is satisfactory considering the difficulty in the human evaluation. The Pearson’s correlation coefficient is 0.37 on coherence and 0.26 on fluency, with $p < 0.0001$.

and *rocks* successfully (4th ex.), it insists on telling *getting married* when it sees *[male]* or *[female]* (1st and 2nd ex.). Such examples suggest that some entities (e.g. *park*) embody semantics more independently, while for others (e.g. *male*), we have to associate them in the specific context. The rest of the models try to generalize the target sentences. The DE-Seq2Seq model uses the core dependency arguments as the skeleton. However, the results demonstrate the generated sentences are quite irrelevant. The sentence may have links such as *walked through* to *came out* (1st ex.), but the objects in the generated stories are hardly related. The GE-Seq2Seq model replaces the specific words with more general concepts and generates some good examples, e.g. the second one in the table. However, there can be over-generalizations. For example, as for the third example, the GE-Seq2Seq model associates *driving* with *car show*, causing the incoherent description. In the last example, the generated story completely diverges from the input. These results prove the drawbacks of static rule-based skeletons. The proposed model uses a skeleton extraction module to adaptively determine the appropriate granularity of skeletons. The skeleton keeps the main semantic of a sentence, which can be a whole sentence, phrases, or even segments. It makes the model learn the dependency of sentences more effectively so that the generated stories are much more coherent.

4.6 Incremental Analysis

In this section, we conduct a series of experiments to evaluate the contributions of our key components. The results are listed in Table 5. The Seq2Seq model is scored the lowest according to BLEU. With the skeleton extraction module, the BLEU score is slightly improved, which suggests that the model starts to learn the connection of longer segments. Finally, with reinforcement learning, the BLEU score significantly overpasses the Seq2Seq model by 40%.

Table 6 shows the human evaluation results. The slight improvement with the skeleton extraction module in BLEU reflects as the decreases in both fluency and coherence. It suggests the necessity of human evaluation. The decreased results can be explained by the fact that the style of the dataset for pre-training the skeleton extraction module is very different from the narrative story

Input 1: <i>[Female] and her friend had a night out on the town.</i>
EE-Seq2Seq: <i>They were very happy to see us. [Male] and [female] were getting married today.</i>
DE-Seq2Seq: <i>We walked through the streets of the city and found a place to sit. The whole family came out to support their families.</i>
GE-Seq2Seq: <i>My friends and I went to a bar last night. [Female] was so happy to be there.</i>
Proposed Model: <i>They went to the bar. They had a great time.</i>
Input 2: <i>[Male] and [male] are talented artists.</i>
EE-Seq2Seq: <i>They were very happy to see us. [Male] and [female] were getting married today.</i>
DE-Seq2Seq: <i>The kids get to play with the kids. We had a great time.</i>
GE-Seq2Seq: <i>The art was very subjective. The artist had a keen eye.</i>
Proposed Model: <i>They are discussing the importance of the lecture.</i>
Input 3: <i>The bride was driving to the wedding.</i>
EE-Seq2Seq: <i>The family was so happy to be there. They were very happy to see us.</i>
DE-Seq2Seq: <i>The view from the top was amazing. There were a lot of people there.</i>
GE-Seq2Seq: <i>I went to the car show last weekend. People are getting ready to go.</i>
Proposed Model: <i>The groom and groomsmen watched the bride. They kissed and shared their pictures.</i>
Input 4: <i>The park was filled with beauty.</i>
EE-Seq2Seq: <i>The plants were so beautiful. The rocks were very pretty.</i>
DE-Seq2Seq: <i>The family was so happy to be there.</i>
GE-Seq2Seq: <i>There were a lot of people there. We had a great time.</i>
Proposed Model: <i>The trees were very sparse. There were also some flowers.</i>

Table 4: Examples generated by the proposed model and the state-of-the-art models.

Models	BLEU
Seq2Seq	0.0028
+Skeleton Extraction Module	0.0029
+Reinforcement Learning	0.0042

Table 5: Automatic evaluations of key components.

dataset. While it may help extract some useful skeletons, it is likely that many of them are not suitable for learning the dependency of sentences. Finally, when the skeleton extraction module is trained on the target domain using reinforcement learning, the human evaluation is improved significantly by 14% on G-score.

Table 7 further shows the results of the skeleton extraction module. As we can see, the module keeps only the essential parts of the sentence. Most of the adjectival phrases and adverbial phrases are removed. Furthermore, we can find that for longer sentences that contain too detailed information, it only extracts the key information. For shorter sentences where all information is nec-

Models	Fluency	Coherence	G-Score
Seq2Seq	7.54	4.98	6.13
+Skeleton Extraction Module	7.26	4.32	5.60
+Reinforcement Learning	8.69	5.62	6.99

Table 6: Human evaluations of the key components.

- 1) *There was a small power station on the side of the building.*
- 2) *The lady wearing the pink shirt decided to stop playing the video and chatted with other guests.*
- 3) *At the end of the night, guests taking pictures before saying goodbye to each other.*
- 4) *Afterwards, we celebrated with some drinks and watched the rest of the parade.*
- 5) *A few miles away was a lake that we really enjoyed watching.*
- 6) *Some of the guests partied harder than others.*
- 7) *The bride was driving to the wedding.*

Table 7: Analysis of the skeleton extraction module. Given a whole sentence as input, the words in red are the extracted skeleton.

essary, it choose to keep all words. It proves that the skeleton extraction module is effective and is expert in only removing detailed information that is not needed.

Furthermore, it is not quite surprising to see that on our dataset, the Seq2Seq model beats the existing state-of-the-art models (DE-Seq2Seq and GE-Seq2Seq) in human evaluation and automatic evaluation. It is mainly attributed to the oversimplification of sentences. For narrative sentences, the key information is usually expressed in a complicated way. It can be a segment, a phrase, or a whole sentence. The simple rules lead to the excessive loss of key information while our proposed model can adaptively determine the appropriate granularity.

4.7 Error Analysis

Although the proposed model outperforms the state-of-the-art models, it needs to be noted that the highest coherence score, 5.62, is a moderate result in human evaluation, indicating that there is still a long way to go before the generated stories reach the human level. Therefore, in this subsection, we give a detailed error analysis to explore what factors affect the performance.

First, we classify the generated stories with scores below 6 that are considered less coherent. We conclude 4 types of errors from these outputs and the distribution of error types are shown in Figure 3. It is expected that the irrelevant scenes make up most of the errors. In addition, there are several examples that are hard to be understood

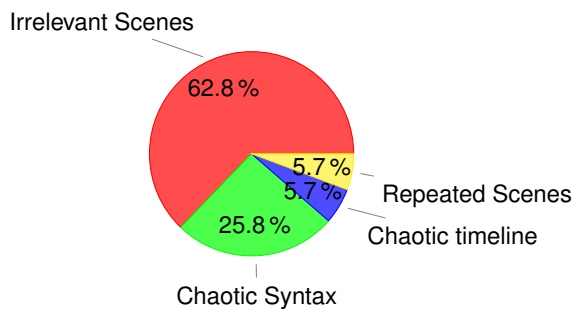


Figure 3: The distribution of error types.

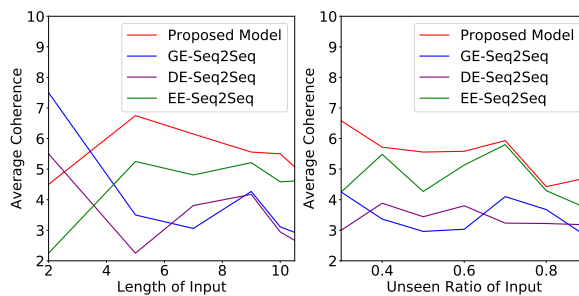


Figure 4: An illustration of how the performance is affected by the length of input (left) and the unseen ratio of input (right).

due to chaotic syntax. For the type of chaotic timeline, the model neglects the time order of scenes and the generated stories goes backward in time. The repeated scenes mean that the generated stories just describe the input again. The above errors show that there are many dimensions in coherence, including scene-specific relevance, temporal connection, and non-recurrence. Modeling such dimensions is still a hard problem.

Furthermore, we explore how the performance is affected by the length of input and the unseen ratio of input. The results are shown in Figure 4. “Unseen ratio” is the percentage of the phrases that are not seen in the training data. We use the gap between 1 and the BLEU score with the training data as the reference to compute it. When the input is short and the model often sees the input, the generated story tends to have high coherence. However, when the length of input increases and the model is not familiar with the input, the coherence goes down. Since our model extracts the key semantics better, the dependency of sentences can be easier to learned, which brings the smaller decrease in coherence.

5 Conclusion and Future Work

In this work, we propose a new skeleton-based model for generating coherent narrative stories.

Different from traditional models, the proposed model first generates a skeleton that contains the key information of a sentence, and then expands the skeleton to a complete sentence. Experimental results show that our model significantly improves the quality of generated stories, especially in coherence. However, even with the best human evaluation results, the error analysis shows that there are still many challenges in narrative story generation, which we would like to explore in the future.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (No. 61673028). We thank all the reviewers for providing the constructive suggestions. Xu Sun is the corresponding author of this paper.

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.
- Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*.
- Fred Charles, Steven J Mead, and Marc Cavazza. 2001. Character-driven story generation in interactive storytelling. In *Virtual Systems and Multimedia, 2001. Proceedings. Seventh International Conference on*, pages 609–615. IEEE.
- Elizabeth Clark, Yangfeng Ji, and Noah A. Smith. 2018. Neural text generation in stories using entity representations as context. In *NAACL HLT 2018, The 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *CoRR*, abs/1805.04833.
- Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1481–1491.
- Brent Harrison, Christopher Purdy, and Mark O Riedl. 2017. Toward automated story generation with markov chain monte carlo methods and deep neural networks. In *Proceedings of the 2017 Workshop on Intelligent Narrative Technologies*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Ting-Hao K. Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Aishwarya Agrawal, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. 2016. Visual storytelling. In *15th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*.
- Parag Jain, Priyanka Agrawal, Abhijit Mishra, Mohak Sukhwani, Anirban Laha, and Karthik Sankaranarayanan. 2017. Story generation from sequence of independent short descriptions. *CoRR*, abs/1707.05501.
- Boyang Li, Stephen Lee-Urban, George Johnston, and Mark Riedl. 2013. Story generation with crowd-sourced plot graphs. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA*.
- Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1106–1115.
- Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1192–1202.
- Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2017. Table-to-text generation by structure-aware seq2seq learning. *CoRR*, abs/1711.09724.
- Liangchen Luo, Jingjing Xu, Junyang Lin, Qi Zeng, and Xu Sun. 2018. An auto-encoder matching model for learning utterance-level semantic dependency in dialogue generation. In *EMNLP, 2018*.
- Shuming Ma, Xu Sun, Junyang Lin, and Houfeng Wang. 2018a. Autoencoder as assistant supervisor: Improving text representation for chinese social media text summarization. *CoRR*, abs/1805.04869.
- Shuming Ma, Xu Sun, Yizhong Wang, and Junyang Lin. 2018b. Bag-of-words as target for neural machine translation. *CoRR*, abs/1805.04871.

- Lara J. Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark O. Riedl. 2018. Event representations for automated story generation with deep neural nets. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*.
- Neil Duncan McIntyre and Mirella Lapata. 2009. Learning to tell tales: A data-driven approach to story generation. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 217–225.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 311–318.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.
- Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pages 1057–1063.
- Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. 2018a. Diversity-promoting gan: A cross-entropy based generative adversarial network for diversified text generation. In *EMNLP, 2018*.
- Jingjing Xu, Xu Sun, Qi Zeng, Xuancheng Ren, Xiaodong Zhang, Houfeng Wang, and Wenjie Li. 2018b. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *ACL, 2018*.