A Character-Centric Neural Model for Automated Story Generation

Danyang Liu,[1,2*] **Juntao Li,**[1,3*] **Meng-Hsuan Yu,**[1,3] **Ziming Huang,**[4] **Gongshen Liu,**[2] **Dongyan Zhao,**[1,3] **Rui Yan**[1,2,3†]

[1]Wangxuan Institute of Computer Technology, Peking University, Beijing, China
[2]Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai Jiao Tong University, Shanghai, China
[3]Center for Data Science, AAIS, Peking University, Beijing, China
[4]IBM Research-China, Beijing, China
{danyliu, lgshen}@sjtu.edu.cn, hzmzi@cn.ibm.com.
{lijuntao, yumenghsuan, zhaody, ruiyan}@pku.edu.cn

Abstract

Automated story generation is a challenging task which aims to automatically generate convincing stories composed of successive plots correlated with consistent characters. Most recent generation models are built upon advanced neural networks, e.g., variational autoencoder, generative adversarial network, convolutional sequence to sequence model. Although these models have achieved prompting results on learning linguistic patterns, very few methods consider the attributes and prior knowledge of the story genre, especially from the perspectives of explainability and consistency. To fill this gap, we propose a character-centric neural storytelling model, where a story is created encircling the given character, i.e., each part of a story is conditioned on a given character and corresponded context environment. In this way, we explicitly capture the character information and the relations between plots and characters to improve explainability and consistency. Experimental results on open dataset indicate that our model yields meaningful improvements over several strong baselines on both human and automatic evaluations.

Introduction

In view of its aesthetic and entertainment values, automatic story generation is becoming a new hotspot in the field of natural language generation, which involves many-faceted challenges (Martin et al. 2018; Lukin, Hobbs, and Voss 2018; Goldfarb-Tarrant, Feng, and Peng 2019). Conventionally, various rule-based or template-based methods with explicitly prior domain knowledge are proposed for addressing this task, e.g., case-based reasoning (Gervás et al. 2004; Swanson and Gordon 2012), agent-based simulation (Brenner 2010), plot graphs (Li et al. 2013). Recently, a myriad of neural models is designed for modeling the story generation task without requiring hand-crafted domain knowledge and thus are more versatile for various scenarios (Fan, Lewis, and Dauphin 2018; Peng et al. 2018; Fan, Lewis, and Dauphin 2019; Mishra et al. 2019). Although encouraging progress has been achieved, neural methods are still restricted by *explainability* and *coherence* owing to their

---

*Equal contribution

†Corresponding author: Rui Yan (ruiyan@pku.edu.cn)

Title (Given)	Rush Hour
Character (Given)	A Police (Represented by a vector)
Context	... A gun battle breaks out.
Predicted Action	Arrest
Generated Sentence	Office Chan arrests a group of gun smugglers.

Table 1: An example of title, character, context, action prediction and sentence generation in our system.

black-box nature and requirements for modeling long-range dependencies and sophisticated plots and characters.

For neural story generation, previous frameworks mainly treat this problem as a standard long document generation task and propose to decompose it as a multi-stage generation process to address the long-range dependency issue. A typical generation formulation is first using a neural language model to generate intermediate representations, e.g., keywords (Yao et al. 2019), skeleton (Xu et al. 2018), prompts (Fan, Lewis, and Dauphin 2018), and then utilizing another neural language model to generate each sentence upon these intermediate representations. In doing so, these models can well capture syntactic and lexical information from training stories but might fail to take task-specific attributes into account, e.g., explicitly modeling characters, plots, etc (Riedl and Young 2010; Bahamón 2012). For one thing, paying much attention to generation framework might lead to the generated stories unexplained from the aspects of story, say a plausible story might consist of irrelevant plots and mis-matched characters. For another, previous neural models mainly focus on modeling semantic-level consistency, e.g., thematic consistency, cross-sentence coherence, leaving character consistency unexplored.

To address the aforementioned issues, we investigate to generate stories from the perspective of story genre.

Enlightened from previous story generation methods with prior knowledge about story genres, we attempt to explicitly combine deep neural generation networks with character modeling which was affirmed to be effective for enhancing the character believability (Riedl and Young 2010). Con-
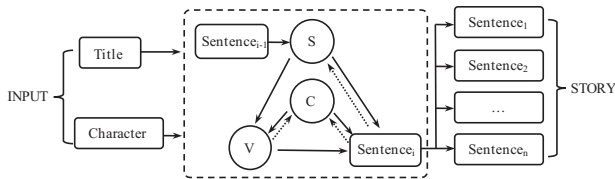
Figure 1: The overall framework of our model. Solid arrows present the generation process of each sentence. Black dotted arrows represent the updating process. S, C, V refers to context embedding, character embedding and predicted action respectively.

cretely, we allocate a consistent character to a story, and the story generation process is reformulated as selecting a sequence of actions of the given characters under the context environment. In this way, each part of a generated story explicitly correlates a character and the given context environment, which will enhance the explainability of the generated stories from the aspect of story genre. In addition, the given character guides the action selection operation at each step through the story generation process so as to promote character consistency. It is also verified in dialogue systems(Li et al. 2016) that such strategy can improve the speaker consistency in neural response generation.

In this paper, we propose a character-centric neural storytelling model, which explicitly encodes character in distributed embedding to guide the story generation. As illustrated in Figure 1, our model contains three elements: character, situation, and action. We represent each individual character as a distributed embedding, which encodes the character's personality traits and how the character will perform in different situations. In our model, story development is driven by continuous interaction between the character and the current situation. Specifically, we can decompose the story generation into two steps. Firstly, our model determines the character's reaction to the current situation at each time step. Secondly, a complete sentence is generated by incorporating the character embedding, predicted action and the situation information. Table 1 shows an example of the generation process. We conduct experiments on a corpus of movie plot summaries extracted from Wikipedia (Bamman, O'Connor, and Smith 2013) because there is rich and clear information about characters. The experimental result shows that our character-centric model achieves improvements as compared to all baseline methods by incorporating character embedding.

Related Work

Automatic story generation has always been a hot topic in the field of artificial intelligence. Before the rise of deep learning, researchers proposed various rule-based or template-based methods for this task, e.g., case-based reasoning (Gervás et al. 2004; Swanson and Gordon 2012), agent-based simulation (Brenner 2010), plot graphs (Li et al. 2013). The disadvantage of these methods is that they could only generate stories with manually-engineered knowledge and well-defined domains of characters and actions.

Recently, deep learning has been adopted to story generation. Many neural models are designed to automatically generate stories without requiring hand-crafted domain knowledge. Fan, Lewis, and Dauphin (2018) proposes a novel hierarchical convolutional sequence to sequence model with self-attention mechanisms and model fusion to cope with the long-range dependencies problems. Recently, some researchers focus on designing intermediate representations which maintain semantic meaning of story while reducing the sparsity. For instance, Martin et al. (2018) proposes a 4-tuple event representation $< s, v, o, m >$, where s is the subject of the verb, v is a verb, o is the object of the verb and m is the modifier. They decompose the story generation process into two steps: event2event and event2sentences. Both of them employ recurrent sequence to sequence networks. In this story generation pipeline, the subsequent event is generated via event2event network and then translated into natural language using event2sentence network. Xu et al. (2018) uses skeleton as the intermediate representation, which is the most critical phrases of a sentence and learned by a reinforcement learning method. The key idea is to first generate a skeleton and then expand the skeleton to a complete sentence. Yao et al. (2019) extracts a sequence of keywords from stories as the storyline and then convert them to natural language sentences.

A lot of work focus on exploring latent space to inspire the text generation. Recently, the transformer network is utilized in the variation autoencoder for sentence generation (Liu and Liu 2019). Prior studies have explored conditional variational autoencoder for generating natural language text, such as Chinese traditional poem (Li et al. 2018) and dialogue response (Qiu et al. 2019). In addition, a cache-augmented conditional variational autoencoder is proposed for story generation (Li et al. 2019), where the cache module allows to improve thematic consistency while the conditional variational autoencoder part is used for generating stories with less common words.

In this paper, we solve the problem of story generation from the perspective of characters. Specifically, we learn character embeddings directly from the corpus, which has been proved to be effective in dialogue systems (Li et al. 2016; Oraby et al. 2018). In order to make the framework more suitable for the story genre, we decompose the story generation into action prediction and sentence generation, aiming at a fine-grained control over the generation process.

Method

In this section, we will introduce our character-centric neural story-telling model in details. First, we provide the problem formulation and necessary notations. We follow the same generation setting as the previous storytelling models (Martin et al. 2018; Xu et al. 2018; Li et al. 2019), where stories are generated in a sentence-by-sentence manner. As illustrated in Figure. 1, the previous generated sentence is used as input to generate the next sentence. We formally define the input and output of our approach as follows:

- INPUT. A title $T = \{t_1, t_2, ..., t_m\}$ and character embeddings $C = \{c_1, c_2, ..., c_p\}$ learned in training phase are
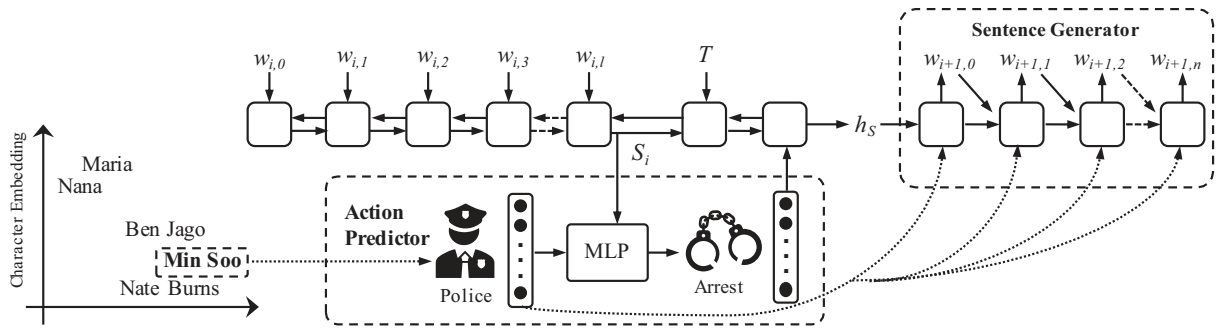
Figure 2: Illustrative example of our character-centric model. On the lower left is the learned character embedding space. Characters close in embedding space tend to have similar personality traits (e.g., job, age, gender). Suppose that the context environment embedding is computed based on the previous sentence *A fierce gun battle breaks out in Hong Kong*, the given character is a policeman *Min Soo* from Hong Kong movie *Blind*. Then our action predictor calculates that the action should be *arrest*. T represents the given title. Combining the above information, the output sentence is *Officer arrests a group of gun smugglers*. Note that the character embedding is involved in each step of sentence generator.

given to our model as the input to generate a story, where t_i denotes the i-th word and m denotes the length of title, c_i denotes the embedding of i-th character in the story.

- OUTPUT. A story $Y = \{y_1, y_2, y_3, ..., y_n\}$ is generated as the result of our model, where $y_i = \{w_{i,1}, w_{i,2}, w_{i,3}, ..., w_{i,l}\}$ represents the i-th sentence of total n sentences in generated story and $w_{i,j}$ denotes the j-th word in the i-th generated sentence. l is the length of S_i

As shown in Figure 2, our model decomposes the story generation into two steps: 1) action prediction which decides the character's reaction towards current context environment and 2) sentence generation which composes a complete sentence. This hierarchical generation framework gives fine-grained control over the generation process, where action is explicitly selected by the given character at each step to enhance the explainability and character consistency. Next, we will introduce the modules of character embedding, action prediction, sentence generation, and training strategy respectively.

Character Embedding

Our model represents each individual character as an embedding. Following (Bamman, O'Connor, and Smith 2013) and (De Marneffe and Manning 2008), we extract the following linguistic features for each character:

- Related verbs. Both agent verbs(for which the character is an agent) and patient verbs(for which the character is an patient).

- Attributes. Adjectives and adjectival modifiers(e.g., noun-noun compounds,appositives).

Inspired by (Joulin, Grave, and Mikolov 2017), these word representations are then averaged into a single representation, which is then used to initialize the character embedding:

$$C = \frac{1}{N} \sum_{i=1}^{N} embedding(w_i) \qquad (1)$$

where w_i denotes the i-th extracted word and N denotes the number of extracted words.

Character embeddings encode attributes of a character(e.g., personality, job, emotion, gender, age) that affect the action decision of the character. Note that after initialization, our model manages to cluster characters along some of these attributes(e.g., personality, job, emotion, gender, age) based on the training process. Specifically, the character embeddings C are learned by back-propagation during the training of both action predictor and the sentence generator.

Action Predictor

Our model takes the characters as the center of a story. The development of a story can be described as a sound and believable sequence of actions that characters perform based on the current context environment.

The context embedding indicates the current environmental information and circumstances that surround the characters in the story world. At each time step, the context embedding S_i is computed by a one-layer bidirectional LSTM network (Hochreiter and Schmidhuber 1997) based on current sentence y_i and previous situation embedding S_{i-1}:

$$S_i = BiLSTM(y_i, S_{i-1}) = [\overrightarrow{S_i}, \overleftarrow{S_i}] \qquad (2)$$

where $\overrightarrow{S_i}$ and $\overleftarrow{S_i}$ are the forward and backward hidden vectors respectively. After computing the context embedding, the character will decide what to do towards the current context environment. The character embeddings carry the information about how the characters will react to the situation. The probabilities over the actions are computed by a multilayer perceptron (MLP) based on the concatenation of character embeddings C and current situation S_i. Formally:

$$p(V_i|S_i, C) = softmax(MLP(S_i, C)) \qquad (3)$$

where C indicates the character embedding and the V_i refers to the action that character will perform at time step i.

Given the ground truth verb V_i with corresponding character C at time step i, the action predictor model is trained to minimize the negative log probability of training data:

$$\mathcal{L}(\theta)_{\text{AP}} = -\frac{1}{N}\sum_{i=1}^{N}\log p\left(V_i|S_i, C\right) \qquad (4)$$

After training, our predictor will acquire the ability to infer the actions. More concretely, it is generalization ability, which means that it helps infer the actions that a character will perform even though he/she never experience such situation in training story corpus. This property is important as the training corpus can not cover all the traits of characters. Our model learns the character representations based on the action choices towards specific situations, making characters with similar acting style tend to locate in nearby area in the embedding space. This property increases the generalization capability of the action predictor. For example, consider characters i and j who are both policemen and with close character embeddings in the embedding space. In training data, policeman i made an arrest when the situation indicates that there was a man firing gun on crowded. Even though character j never experience the same situation, there is a high probability that j will perform the *arrest* action due to their similar embedding vector. This is the generalization ability of character embedding, which helps characters perform actions consistent with their traits in unfamiliar situations.

Sentence Generator

The story is generated sentence by sentence. We formulate it as a conditional generation problem which generates a sentence based on the action to be performed V_i, the character embedding C and the current situation S_i at time step i. Specifically, we employ a sequence to sequence structure in sentence generator, where both encoder and decoder are one-layer bidirectional LSTM networks with attention mechanism. As in standard sequence to sequence model, we first encode the situation embedding S_i, title T and action V into a low-dimensional vector h_S by first concatenating them with field separator $< EOS >$ and $< EOT >$ in between, and then use a bidirectional LSTM network to encode them:

$$h_S = encoder(S, T, V) = [\overrightarrow{h_S}, \overleftarrow{h_S}] \qquad (5)$$

where $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ are the forward and backward hidden vectors respectively. The decoder generates a sentence conditioned on h_S and character embedding C. Specifically, for each time step in decoder, hidden units are obtained by combining the hidden state at previous time step h_{i-1}, the word embedding at current time step e_i, and the character embedding C:

$$h_i = decoder([h_S, e_i, C]) \qquad (6)$$

where w_i denotes the i-th generated word in target sentence.

Given the target sentences y_i, the sequence to sequence model is trained to minimize the negative log-likelihood of the sentences in the training set:

$$\mathcal{L}(\gamma)_{\text{SG}} = -\frac{1}{N}\sum_{i=1}^{N}\log p\left(y_i|h_{i-1}, y_{i-1}, V, C, T\right) \qquad (7)$$

It is important to incorporate character embeddings in sentence generation stage. Because in addition to actions, a lot of character-related information(e.g. adjective, adverb, object) is needed to form a complete sentence together. For example, consider character i is a basketball player and the predicted action is *play*. Taking character embedding C into consideration makes our model tend to generate *basketball* instead of *computer* as the object of *play*.

Meanwhile, character embeddings enhance the explainability and character consistency. For example, consider character i is a new-born child, when facing dangerous all a newborn can do is crying. But the existing models based on maximum likelihood estimation (MLE) are prone to remember common patterns of the story corpus, which means they tend to predict the action *run away* or other words that are often co-occurring in dangerous situations in the story corpus. Our model explicitly refers to the character embedding at every time step, which gives the model the ability to choose appropriate actions that match the character's attributes(e.g. age, gender, personality).

Training Strategy

The overall objective of our character-centric model is to minimize:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{AP}} + \beta \cdot \mathcal{L}_{\text{SG}} \qquad (8)$$

where $\mathcal{L}_{\text{AP}}$ and $\mathcal{L}_{\text{SG}}$ refer to the loss of action predictor and sentence generator respectively. Herein α and β are both hyper-parameters used to balance the loss of two parts. Our code will be available at https://github.com/liudany/character-centric.

Experiment

Dataset

We conduct experiments on the corpus of movie plot summaries extracted from Wikipedia. This corpus contains a concise summary of the film's events, with implicit descriptions of the characters such as *rebel leader Princess Leia* and *evil lord Darth Vader*.

There are 42,306 stories in this corpus. The median and average lengths of these summaries are approximately 176 and 310 words, respectively. Each sentence contains 16 words on average. To preprocess the corpus, we first employed Mosesdecoder tools for tokenization and then converted all the words to lowercase. The vocabulary size is set to 50,000 and 2.06% words were replaced by $< unk >$ symbol. We randomly split the corpus into 34,306/4,000/4,000 stories for training, validating and testing respectively.

The original corpus contains no movie name but a Wikipedia movie ID at the beginning of each story. We replace the movie ID with the corresponding movie name because our model is designed to generate a story based on a story title. Figure. 2 shows an example story and its corresponding title.

We use the Stanford CoreNLP library to extract the structure and implicit information from the corpus. Specifically, all of the plot summaries run through the Stanford CoreNLP

Title: A Song Is Born
Mild-mannered Professor Hobart Frisbee and his fellow academics, among them Professor Magenbruch , are writing a musical encyclopedia. When they discover that there is some new popular music that is called jazz, swing, boogie woogie or rebop introduced by two window washers Buck and Bubbles. The professors become entangled in the problems of night club singer Honey Swanson ....

Table 2: An example in movie plots corpus.

Criterion	Meaning
Explainibility (Exp.)	Is the story explainable, meaningful, and with reasonable motivation?
Character-Believability (CB.)	Does the story display consistent and believable characters?
Fluency (Flu.)	Is the sentence grammar in the generated story correct?
Overall (Ovr.)	The average of the above criteria.

Table 3: Details about the human evaluation criteria.

pipeline(tagging, parsing, NER, and coreference). We followed (Martin et al. 2018) for splitting and pruning sentences. Prepositional phrases were removed and we split sentential phrases on conjunctions. The result contains a structured representation for each story. We extract the related verbs(for which the character is the subject or object) and attributes(e.g. adjectives and appositives) to initialize our model.

Baselines

We compare our proposed model to several related story generation methods which are non-character-based to evaluate the effectiveness of our character-based generation method. These baselines are described as follow:

Conditional Language Model(C-LM). The conditional language model generates the whole story word by word. The title of each story is encoded to a vector representation as the initial hidden state of a RNN-based language model. Specifically, we employ GRU (Chung et al. 2015) cells as the basic unit of the language model.

Seq2Seq with Attention(Vanilla-Seq2Seq). Seq2Seq (Sutskever, Vinyals, and Le 2014) with attention mechanism is the most effective generation method in many natural language processing tasks, which follows a sentence-by-sentence generation manner. Given a tile, the i-th sentence is generated from the title and last sentence.

Incremental Seq2Seq with Attention(Incre-Seq2Seq). Incremental Seq2Seq enhances the Vanilla-Seq2Seq method by not only taking the previous sentence into consideration but also all the previous generated sentences, which generates a story in an incremental manner.

Plan-and-write. It is a hierarchical generation approach which is proposed in (Yao et al. 2019). This method generates a story by first generate some keywords as a storyline which indicates the story plots and then decodes the storyline to a complete story.

Event Representation. Martin et al. (2018) proposes a hierarchical generation framework where story generation is decomposed to two subtasks: the generation of successive events as the story plots and the generation of human-readable sentences from event. Each event is represented by a 4-tuple $< s, v, o, m >$ where s and o are the subject and object of verb v, and m indicates the redundancy.

Hierarchical Convolution Sequence Model (Hierarchical). The state-of-the-art story generation model (Fan, Lewis, and Dauphin 2018) which is combination of a novel hierarchical convolution models, self-attention mechanisms and model fusion.

Experimental Settings

Our proposed model is trained under the following parameters and hyperparameters setting. For sentence generator, both encoder and decoder are composed of 1 layer with 512-dimensional hidden states. The balancing hyper-parameters α and β are set to 1 and 0.8 respectively. The character embedding is set to 512 which is the same as word embedding size. Word embeddings are randomly initialized and shared across the model. We use Adam optimization algorithm (Kingma and Ba 2014) with learning rate $\alpha = 0.001$.

Evaluation

We use the following kinds of evaluations:

BLEU The BLEU algorithm is originally designed for evaluating the quality of machine translation results. Currently, it is used in many generation tasks(e.g. dialogue system, summarization, story generation and so on). Specifically, it analyzes the word overlapping between the generated sentence and the ground-truth sentence.

Perplexity Perplexity is usually used for evaluating the quality of a language model. In general terms, it indicates the probability that the given sentence is fluent.

Human Evaluation Although the automatic evaluation generally computes the quality of generated stories from a statistical perspective, it can not comprehensively or accurately evaluate the text. Therefore, we conduct human evaluation to score generated stories from three aspects: explainability, character-believability, and fluency, as defined in Table 3. We hired five well-educated human evaluators to annotate generated stories from above aspects with four score levels: 1, 2, 3 and 4. A higher score indicates better performance. We randomly choose 100 stories generated by different models and distributes them to evaluators. Note that all the evaluators have no idea about which model the story is from. We use the average of the three aspects as the final score to indicate the overall performance.

Model	Automatic Evaluation					Human-evaluation			
	BLEU-1	BLEU-2	BLEU-3	BLEU-4	PPL	Exp.	CB.	Flu.	Ovr.
C-LM	21.29	7.11	1.84	0.19	112.45	1.32	2.38	1.97	1.89
Vanilla-Seq2Seq	19.61	5.19	1.71	0.21	78.23	2.43	1.71	1.98	2.04
Incre-Seq2Seq	22.83	6.15	1.91	0.18	74.56	2.52	1.89	2.01	2.14
Plan-and-Write	28.79	7.21	1.28	0.21	58.72	2.65	2.45	2.36	2.49
Event Representation	29.30	6.72	1.54	0.23	54.25	2.87	2.64	2.41	2.64
Hierarchical	**30.91**	6.11	1.46	0.31	**52.28**	2.82	2.59	**2.75**	2.72
Character-centric	30.78	**7.24**	**1.97**	**0.32**	52.93	**2.93**	**2.87**	2.74	**2.85**

Table 4: Results of automaic and human evaluations. **BLEU-[1-4]** represents BLEU scores on [1-4]-grams; **PPL** represents perplexity. **Exp., CB., Flu. and Ovr.** represent explanibility, character-beliveability, fluency and overall scores retrospectively.

Results

Overall Result

In this section, we will show the performance of different models on test dataset and offer some further analysis. Table 4 shows the performance of proposed model and baselines in BLEU and perplexity on movie plot summaries corpus. Compared with conditional language model, our proposed model achieves better perplexity and BLEU values. Because the average length of our corpus is too long for a RNN-based language model. The one pass generation manner is not suitable for story generation. We observe that our model achieves better evaluation results than vanilla sequence to sequence model and incremental sequence to sequence model, which can be regarded as ablation experiments. The results prove the effectiveness of character embedding and action prediction. More detailed ablation experiments will be described later. Compared to some state-of-the-art story generation method (Martin et al. 2018; Yao et al. 2019; Fan, Lewis, and Dauphin 2018), we can observe that our proposed model performs the best according to BLEU-[2-4] and human evaluation. Specifically, plan-and-write model employ language model conditioned on a sequence of keywords to generate a story word-by-word, which focus on local word relations and yields comparable BLEU and perplexity values. Hierarchical model employs convolutional sequence to sequence model with model fusion and self-attention machanism, which is capable of capturing long-range dependencies. We observe that hierarchical model achieves the best results in terms of BLEU-1, perplexity, and fluency. Event representation model employs sequence to sequence model in both generating event sequence and translate events to natural language sentences, which could better capture the global information between events and sentences. Event representation model and our model achieve comparable values on automatic evaluations, but our model generates stories from the perspective of characters while event representation model focus on tight semantic intermediate representations. As for human evaluation, our proposed model receives the best score of consistency and character believability. These results support the intuition that the character-centric model could enhance the explainibility and character consistency.

Model	PPL	Exp.	CB.	Flu.
Non-AP Model	60.48	2.58	2.65	2.39
Non-character Model	55.27	2.51	2.45	2.37
Proposed Model	**52.93**	**2.93**	**2.87**	**2.74**

Table 5: Evaluations results of the ablation experiments.

The Effect of Character Embedding

As shown in Figure 5, we implement a non-character model to conduct the ablation study and investigate how the character embedding affect the generated stories. In non-character experiment, the model follows the same settings as the original model except that there is no character embedding. We analyze these results from the following perspectives.

Character embeddings improves character-consistency in generated stories. We observe that the character-centric model achieves lower perplexity values of 52.93 than the non-character model on test dataset. In terms of human evaluation, our proposed model outperforms the non-character model in character-belivebality criterion. The result supports the intuition that the character embedding could improve the character-consistency in generated stories.

Character embeddings enhances the explainability of stories. Generally speaking, explainability is positively correlated with character consistency i.e. a consistent character makes the story reasonable. As shown in Table 5, our proposed model achieves better explainability score in human evaluation. The result indicates that character-guided stories achieve better explainability.

Characters are clustered along some of there traits in the embedding space. As described above, our character embeddings are learned by predicting their actions in story generation, which could capture their behavioral style. Figure 2 demonstrates the character embeddings after training on movie plot corpus, which confirms our intuition that similar characters are near to each other in embedding space.

The Effect of Action Prediction

The action predictor decomposes story generation into a two-stage task, which aims at a fine-grained control over the generation process. To isolate the effect of action predictor, we perform ablation experiment which demonstrates the effectiveness of action predictor. As shown in Figure 5, the

Title: Policeman
In Hong Kong, a group of gun smugglers are making a deal. Officer Chan and his partner attempt to arrest them. They shot gang members in the street. Police officers are badly wounded. Chan wants to revenge. They meet up again. Wong wants to blow up the building. Chan kills Wong. Chan leave away from Hong Kong.
Title: Divorce
The plot focuses on a famous actor, Micheal. He hires a actor girl. When Micheal marries the actor girl, his wealthy parents leave him. Wendy tells him must deal with the problem, he decides not to talk. He leaves the town for his own. He tries to commit suicide. He swims through the river of town and drives some kilometers down the streets lines. He has been spending the night out. As the day progresses, Micheal finally divorces with her.

Table 6: Example stories generated by our proposed character-centric model.

action predictor module improves the quality of generated stories in both automated and human evaluations.

Action predictor reduces the difficulty of sentence generation. We observe a notable decrease in perplexity for our proposed model over the Non-AP model. By predicting the upcoming action of the given character, more informative elements are provided to the sentence generator, making the generated sentence fluent and accurate.

Action predictor explicitly gives character guidance to story generation process. With explicit action prediction, our model regard the story generation process as the process of character's thought and reaction towards changing context environment, which gives the story a reasonable motivation. The explanibility score in Table 5 supports our intuition.

Discussion

Compared to the previous plot-based models (Martin et al. 2018; Yao et al. 2019), we observe a meaningful performance boost introduced by the character embedding. Although there is character information involved in the event representation model (Martin et al. 2018), its design purpose is only to decrease the sentence-level sparsity as an intermediate representation, and it cannot explain the story motivation from the perspective of characters. In contrast to plan-and-write model (Yao et al. 2019) where the story generation is upon keywords, we take context environment, the decision of character and character embedding into consideration when generating a sentence, which is more explainable and leads to a better result.

Case Study

Figure 6 shows example stories generated by our proposed character-centric model on movie plots corpus. We can observe that the model can generate stories with explainability and character-consistency, which means the audience will

Title: Gangster's Love
Kota longs so much for love. Prabhakaran is a poor, fearless workaholic gangster with a dangerous mood. He begins his pursuit. Kota falls in love with him. They get engaged when they come to know about the accident when they arrive on the family flat to discuss their wealth. After going into a casino, a teacher was found as a victim. Electricity and its stabs and light doors are found - and there are rational regimes upside down between the sky and earth.

Table 7: An inferior example story generated by our proposed character-centric model.

not doubt the authenticity of the plots and characters. With the given titles, the generated stories are closely related to the titles. Although there exists a gap between generated stories and human-created ones, the meaningful generated long story still demonstrates the effectiveness of our proposed character-centric model.

There are also a few inferior stories generated by our model. As shown in Figure 7, when there is no obvious leading character in the story, the logic of the story will be chaotic. With the development of the story, this error will accumulate all the time, making generated story unreadable. To address such issue, one potential solution is to add a multi-character action module to control the inter-reaction between characters. Another notable pattern is that our model is prone to generate common words with high occurrence frequencies. The issue stems from RNN architecture. To alleviate such issue, one can combine strategies such as conditional autoencoder (Li et al. 2019).

Conclusion

In this paper, we propose a character-centric storytelling model that generates stories with explicitly character representations. Specifically, by encoding personalities into distributed character embeddings, we are able to capture personal characteristics. Moreover, we introduce an action predictor module to decompose the story generation process into two-steps: action prediction and sentence generation, which provides a fine-grain control on the story generation process. Experimental results on movie plots corpus indicate that our model can generate stories with better explainability and character-consistency.

Acknowledgments

References

Bahamón, J. C. 2012. Toward a computational model of character personality for planning-based narrative generation. In *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*.

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.

Brenner, M. 2010. Creating dynamic story plots with continual multiagent planning. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2015. Gated feedback recurrent neural networks. In *International Conference on Machine Learning*, 2067–2075.

De Marneffe, M.-C., and Manning, C. D. 2008. Stanford typed dependencies manual. Technical report, Technical report, Stanford University.

Fan, A.; Lewis, M.; and Dauphin, Y. 2018. 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. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2650–2660. Florence, Italy: Association for Computational Linguistics.

Gervás, P.; Díaz-Agudo, B.; Peinado, F.; and Hervás, R. 2004. Story plot generation based on cbr. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, 33–46. Springer.

Goldfarb-Tarrant, S.; Feng, H.; and Peng, N. 2019. Plan, write, and revise: an interactive system for open-domain story generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 89–97. Minneapolis, Minnesota: Association for Computational Linguistics.

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

Joulin, A.; Grave, E.; and Mikolov, P. B. T. 2017. Bag of tricks for efficient text classification. *EACL 2017* 427.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Li, J.; Galley, M.; Brockett, C.; Spithourakis, G.; Gao, J.; and Dolan, B. 2016. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 994–1003.

Li, J.; Song, Y.; Zhang, H.; Chen, D.; Shi, S.; Zhao, D.; and Yan, R. 2018. Generating classical Chinese poems via conditional variational autoencoder and adversarial training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3890–3900. Brussels, Belgium: Association for Computational Linguistics.

Li, J.; Bing, L.; Qiu, L.; Chen, D.; Zhao, D.; and Yan, R. 2019. Learning to write stories with thematic consistency and wording novelty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1715–1722.

Liu, D., and Liu, G. 2019. A transformer-based variational autoencoder for sentence generation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–7. IEEE.

Lukin, S.; Hobbs, R.; and Voss, C. 2018. A pipeline for creative visual storytelling. In *Proceedings of the First Workshop on Storytelling*, 20–32. New Orleans, Louisiana: Association for Computational Linguistics.

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 *Thirty-Second AAAI Conference on Artificial Intelligence*.

Mishra, A.; Laha, A.; Sankaranarayanan, K.; Jain, P.; and Krishnan, S. 2019. Storytelling from structured data and knowledge graphs : An NLG perspective. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, 43–48. Florence, Italy: Association for Computational Linguistics.

Oraby, S.; Reed, L.; Tandon, S.; Sharath, T.; Lukin, S.; and Walker, M. 2018. Controlling personality-based stylistic variation with neural natural language generators. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, 180–190.

Peng, N.; Ghazvininejad, M.; May, J.; and Knight, K. 2018. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, 43–49.

Qiu, L.; Li, J.; Bi, W.; Zhao, D.; and Yan, R. 2019. Are training samples correlated? learning to generate dialogue responses with multiple references. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3826–3835.

Riedl, M. O., and Young, R. M. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research* 39:217–268.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Swanson, R., and Gordon, A. S. 2012. Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 2(3):16.

Xu, J.; Ren, X.; Zhang, Y.; Zeng, Q.; Cai, X.; and Sun, X. 2018. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4306–4315.

Yao, L.; Peng, N.; Weischedel, R.; Knight, K.; Zhao, D.; and Yan, R. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7378–7385.