CHAE: Fine-Grained Controllable Story Generation with Characters, Actions and Emotions

Xinpeng Wang[1], **Han Jiang**[1], **Zhihua Wei**[1]*, **Shanlin Zhou**[2]
[1]Department of Computer Science and Technology, Tongji University, Shanghai, China
[2]School of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, China
{wangxinpeng, 2230780, zhihua_wei}@tongji.edu.cn
zhoushanlin@mail.shiep.edu.cn

Abstract

Story generation has emerged as an interesting yet challenging NLP task in recent years. Some existing studies aim at generating fluent and coherent stories from keywords and outlines; while others attempt to control the global features of the story, such as emotion, style and topic. However, these works focus on coarse-grained control on the story, neglecting control on the details of the story, which is also crucial for the task. To fill the gap, this paper proposes a model for fine-grained control on the story, which allows the generation of customized stories with characters, corresponding actions and emotions arbitrarily assigned. Extensive experimental results on both automatic and human manual evaluations show the superiority of our method. It has strong controllability to generate stories according to the fine-grained personalized guidance, unveiling the effectiveness of our methodology. Our code is available at https://github.com/victorup/CHAE.

1 Introduction

Story generation, one of emergent tasks in the field of natural language generation, requires following sentences given the beginning of the story. For human beings, it is believed that storytelling requires strong logical thinking ability and organizational competence, and for machines it may be even more intractable. Nonetheless, works on story generation can help machines communicate with humans and drive improvements in natural language processing (Alabdulkarim et al., 2021).

At present, most works on story generation focus on the coherence of the story generated according to keywords, outlines and commonsense knowledge (Yao et al., 2019; Guan et al., 2019; Rashkin et al., 2020; Guan et al., 2020; Ji et al., 2020). Some other works aim at generating stories controlled by overall emotion, style, and topic (Keskar et al., 2019; Xu et al., 2020; Brahman and Chaturvedi, 2020; Kong et al., 2021). However, in reality, people often expect more detailed designs catering to their needs rather than a simple theme or topic in the generated story. For example, a novel with more complete elements, i.e., plot, character, theme, viewpoint, symbol, and setting is usually preferred to those made up out of thin air.

Taking the control in story generation as the cutting point, GPT-2 (Radford et al., 2019) can fulfill the story according to the beginning, but the process of generation cannot be controlled by people, resulting in unlogical outputs that lack practicality. CTRL (Keskar et al., 2019) can specify the generation of articles with different styles through some style words, but such control stays at the coarse-grained level, and makes a relatively weak influence. CoCon (Chan et al., 2020) introduces natural language to guide text generation. Fang et al. (2021) propose a new task that guides paragraph generation through a given sequence of outline events. However, the above two studies just explicitly add some contents to the generated sentences, which is similar to forming sentences with given phrases, not using the input as a condition guide for the generative models. SoCP presented by Xu et al. (2020) can generate stories under changeable psychological state control, while it does not govern the detailed contents of the story.

In this paper, we consider more fine-grained control on story generation, and propose a model, **CHAE** for fine-grained controllable story generation, allowing the generation of stories with customized **CH**aracters, and their **A**ctions and **E**motions. Characters are the core of the story. Their actions drive the story along, and their emotions make the story lively and interesting. Consequently, we take the characters along with their actions and emotions as the control conditions. It is a challenge that our model needs to control multiple characters with their actions and emotions respec-

*Corresponding author

tively in a story, especially under the guidance in the form of natural language. To crack the nut, a novel input form conducive to fine-grained control on story generation is introduced into CHAE. Concretely, we use various prompts for fine-grained control conditions in different aspects. Moreover, we design different methods for different control conditions to improve the control effect. Inspired by multi-task learning, we incorporate a character-wise emotion loss while training, thus enforcing the relevance between the characters and their emotions respectively.

The contributions of our work can be summarized as follows:

- We first take the characters with their actions and emotions of the story into account to conduct more fine-grained controllable story generation.
- We propose a model CHAE with a novel input form that helps the model control the story in various aspects, and a character-wise emotion loss to relate the characters and the corresponding emotions.
- The results of both automatic and human evaluation show that our model has strong controllability to generate customized stories.

2 Related Work

Story Generation Story generation has attracted more and more researchers to explore in recent years. There are many challenges in the task, such as context coherence and control. For context coherence, some works are devoted to introducing a series of keywords (Yao et al., 2019), outlines (Rashkin et al., 2020), or incorporating external knowledge (Guan et al., 2019, 2020; Ji et al., 2020) into the story. For style and sentiment control, Kong et al. (2021) generate stories with specified style given a leading context. However, it only focuses on the global attributes of the story. Brahman and Chaturvedi (2020) work on generating stories with desired titles and the protagonists' emotion arcs, and Xu et al. (2020) generate stories considering the changes in the psychological state, while they just control the emotion lines instead of the detailed contents.

Controllable Text Generation We have witnessed the great performance of SOTA models for text generation these years. Despite the progress in coherence and rationality of the text generated, controllability remains to be challenging, which means generating text with specific attributes, such as emotion, style, topic, format, etc. CTRL (Keskar et al., 2019) can control the overall attributes such as domain, style and topic of the generated text by adding control codes and prompts. By plugging in a discriminator, PPLM (Dathathri et al., 2020) can guide text generation without further training the language model. CoCon (Chan et al., 2020) fine-tunes an intermediate block with self-supervised learning to control high-level attributes i.e., sentiment and topic. Compared to the previous works, our work places the emphasis on more fine-grained, all-round control on the generating process, including the control of characters with their emotions and actions in the story.

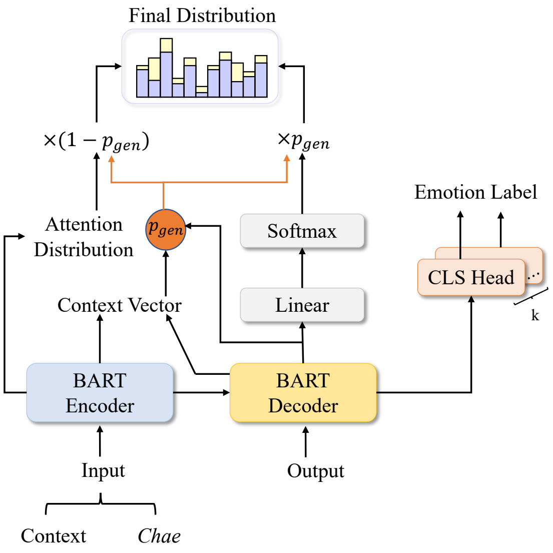

Figure 1: The architecture of CHAE. The input is the concatenation of two components, $Context$ and $Chae$, which will be further explained in Sec 3. The emotion labels are used for calculating a character-wise emotion loss to tie up the characters and their emotions respectively.

3 Methodology

3.1 Problem Formulation

The process of fine-grained controllable story generation in this work is defined as follows.

The input of the task has two components. We refer to the one as $Context$. Let $Context = (x_1, x_2, ..., x_p)$ denote the beginning sentence of the story, which will be the initial $Context$. The

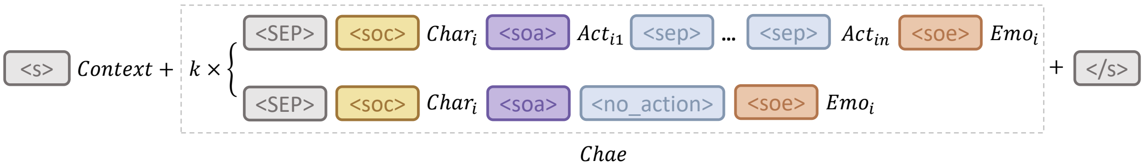

Figure 2: The input form of CHAE. The input starts with $\langle s\rangle$ and ends with $\langle /s\rangle$, comprising $Context$ and $Chae$. The latter is a sequence of k control conditions on the next sentence to be generated, and each condition controls a character. We show two possible forms of the control conditions after the brace. The special tokens in $Chae$ are further explained in Table 1.

other component is $Chae$, a sequence of k fine-grained control conditions on the next sentence. Each condition in $Chae$ is the combination of the name $Char_i$, n actions $Act_{i1}, Act_{i2}, ..., Act_{in}$, and emotion Emo_i of a character appearing in the next sentence to be generated, where n is not fixed and i is the index of the character. Note that we use Italic $Chae$ to distinguish the special input component from our model **CHAE**.

The model predicts one sentence denoted as $Y = (y_1, y_2, ..., y_q)$ at a time by estimating the conditional probability $P(Y|Context, Chae)$. Here we embody the idea of auto-regression by adopting an iterative generation strategy, that is, the sentence generated is then concatenated to $Context$ for next prediction. Especially, at training time, we concatenate the gold sentences instead of generated sentences to $Context$ incrementally like teacher forcing.

The goal of this task is to generate a story where each sentence adheres to the input condition $Chae$ in terms of character, action, and emotion, through which elevate the quality of generation in a fine-grained manner.

3.2 Model Architecture

The architecture of our model CHAE is shown in Figure 1. CHAE is built upon a BART model (Lewis et al., 2020). As mentioned, our model embodies the idea of auto-regression by adopting an iterative generation strategy. On the one hand, the iteratively updated $Chae$ helps control the content of each sentence at a detailed level of granularity. On the other hand, the strategy ensures that the model can always see the foregoing. Considering the incremental $Context$ can be extra long, we employ BART rather than GPT-2. GPT-2 is an auto-regressive model fully based on transformer decoder, while BART has a bidirectional encoder, which might make it better in understanding and encoding long input sequences. To confirm the hypothesis, we also compared BART with GPT-2 on the benchmark dataset in Sec 4, and found that BART outperformed GPT-2 in story generation.

Special tokens	Meaning
$\langle SEP\rangle$	The start token of a condition.
$\langle soc\rangle$	The start token of a character's name.
$\langle soa\rangle$	The start token of actions.
$\langle soe\rangle$	The start token of an emotion.
$\langle sep\rangle$	The start token of a single action.
$\langle no\_action\rangle$	The token representing no action.

Table 1: The meanings of the special tokens in $Chae$.

3.3 Generation Based on Fine-Grained Control Conditions

To generate a story with the characters, their actions and emotions specified, we need to remind the BART model of the elements controlled currently from time to time. Inspired by the practice of leveraging special tokens for controllable generation (Fang et al., 2021; Keskar et al., 2019; Tsutsui and Crandall, 2017), we propose a novel form of input (titled $Chae$), which is a sequence of k fine-grained control conditions on the next sentence to be generated. Each condition in $Chae$ controls a character, and each segment in the condition controls an element (i.e., character's name, action, and emotion) of the corresponding character. Note that any number of actions can be assigned in a condition. The nested sequence form of $Chae$ facilitates the neat combination of various fine-grained control conditions.

We design several special tokens and add them between each segment as the control prompts (see Figure 2). In this study, 6 special tokens are used to prompt the model. They are $\langle SEP\rangle$, $\langle soc\rangle$, $\langle soa\rangle$, $\langle soe\rangle$, $\langle sep\rangle$, and $\langle no\_action\rangle$. The meanings of the tokens are shown in Table 1.

Then, we encode the input $Context$ and $Chae$

as h_{enc} by the BART encoder:

$$\boldsymbol{h}_{enc} = \mathbf{Enc}(\boldsymbol{e}_c), \tag{1}$$

where $\boldsymbol{e}_c$ is the embedding of $Context$ and $Chae$. The vocab generation probability is calculated by the BART decoder as:

$$\begin{aligned} P_{voc}(y) &= P(y_t|y_{<t}, Context, Chae) \\ &= \text{softmax}(\boldsymbol{W}_{voc}\boldsymbol{h}_{dec}), \end{aligned} \tag{2}$$

$$\boldsymbol{h}_{dec} = \mathbf{Dec}(\boldsymbol{e}_{y<t}, \boldsymbol{h}_{enc}), \tag{3}$$

where $\boldsymbol{e}_{y<t}$ is the embeddings of the generated tokens before timestep t, and $\boldsymbol{W}_{voc}$ is a trainable parameter.

After fine-tuning with these special tokens, the model is aware of the elements controlled by each segment of each condition.

3.4 Improvement in Control Effect

3.4.1 Character and Action Control

For characters and their actions, we expect to see the characters and their actions in current $Chae$ appear in the coming sentence. Inspired by the usage of copy mechanism (See et al., 2017; Deaton et al., 2019; Prabhu and Kann, 2020) in copying significant tokens from the input sequences, we add a copy pointer to BART for the information in $Chae$. The attention distribution on $Chae$ denoted by $\widetilde{a}$ is attained by averaging the multiple heads in the cross attention block of BART decoder:

$$\widetilde{a} = \frac{\sum_{i=1}^{h} a_i}{h}, \tag{4}$$

where h is the number of the attention heads, and a_i denotes the attention distribution on $Chae$ from the i-th attention head.

When generating stories, we first combine the hidden state of the decoder $\boldsymbol{h}_{dec}$, the context vector $\boldsymbol{h}_{con}$, and the embedding of the decoder input $\boldsymbol{e}_y$. Secondly, we calculate a generation probability p_{gen}, which is a soft switch to choose a word from the vocabulary according to P_{voc}, or to copy a word from $Chae$ by sampling from the mean attention distribution $\widetilde{a}$. The final distribution of a word can be represented as follows:

$$P(y) = p_{gen}P_{voc}(y) + (1 - p_{gen}) \sum_{j:y_j=y} \widetilde{a_j}, \tag{5}$$

$$p_{gen} = \sigma(\boldsymbol{W}_p^{\top}[\boldsymbol{h}_{dec}; \boldsymbol{h}_{con}; \boldsymbol{e}_y]), \tag{6}$$

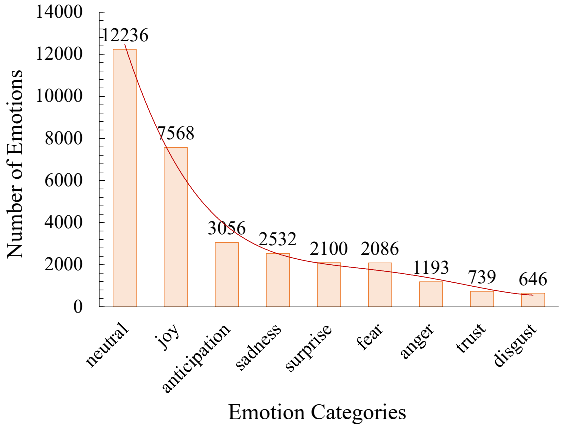

Figure 3: Statistics of emotion categories of sentences in all stories.

where $\boldsymbol{W}_p$ is a trainable parameter. In this way, the characters and actions in $Chae$ will be produced with higher probability, and the output can be flexibly changed according to the different input.

3.4.2 Character-Wise Emotion Control

For characters' emotions, we additionally incorporate a character-wise emotion loss, by which the model is forced to generate sentences with specified emotions tied up with corresponding characters. We add k emotion classification heads to the top of the decoder output layer (Ide and Kawahara, 2021), and k is equivalent to the number of conditions in $Chae$. It provides direct supervision on emotion control to predict the emotion of every character in the story.

However, as shown in Figure 3, the emotion categories present a long tail distribution. To relieve the class-imbalance problem, we use a Weighted Cross-Entropy (WCE) loss between the predicted emotion distribution P_{emo} and the emotion labels l_e:

$$L_{EMO} = -\alpha_e l_e \log(P_{emo}), \tag{7}$$

$$P_{emo} = \text{softmax}(\boldsymbol{W}_{emo}\boldsymbol{h}_{dec}), \tag{8}$$

$$\alpha_e = \frac{N}{e * \text{count}(l_e)}, \tag{9}$$

where $\boldsymbol{W}_{emo}$ is a trainable parameter, α_e denotes the weights of emotion classification labels. N is the number of the training samples, e is the number of emotion categories, and count$(\cdot)$ is a function to calculate the number of samples in each emotion category. Now, the model has both explicit control from emotion in $Chae$ and implicit control from the emotion loss to generate sentences with target emotions.

3.5 Objective Function

We minimize the Negative Log-Likelihood (NLL) loss of the target story sentence Y with the input $Context$ and $Chae$:

$$L_{NLL} = -\sum_{t=1}^{T} \log P(y_t | y_{<t}, Context, Chae), \tag{10}$$

The total loss L is as follow:

$$L = L_{NLL} + \lambda L_{EMO}, \tag{11}$$

where λ is a hyper-parameter. After training with the above two objectives, our model can generate fluent stories under desired conditions.

4 Experiments

4.1 Dataset

We use ROCStory with labeled characters' emotions and actions as our dataset (Rashkin et al., 2018), which contains 14738 five-sentence stories, including 9885 stories for training, 2483 stories for validation and 2370 stories for testing. To conform to the iterative generation, we divide the stories into 39540 / 9932 / 9480 sentence pairs for training/validation/testing. Each pair of sentences consists of two adjacent sentences in the story. The characters with their actions and emotions in the dataset are labeled by three crowdsourced workers from Amazon Mechanical Turk. The emotions come from Plutchik psychology theory (Plutchik, 1980), including 8 species, such as "joy", "anger", etc. Most actions represent the character's underlying motivations, and they generally take the infinitive such as "to win all games" and "to have fun".

In preprocessing, we integrate the annotations of the three workers, and take the emotions with the highest confidence as the final labels. However, we still notice that some emotion labels have excessively low confidence, indicating subtle emotion tendencies. We modify them to "neutral" to avoid distortion of the emotions. Moreover, we find that 9885 stories in the training set are not labeled with emotions, but only with actions. As a remedy, we fine-tune the model on these 9885 stories to keep the fluency of the story. Later, we re-divide all the stories with labeled emotions in the validation set and testing set into another 3 splits. Finally, we fine-tune the model on the new training set to capture emotional information. The re-divided dataset includes 15528 / 1944 / 1940 sentence pairs by 8:1:1 for training/validation/testing.

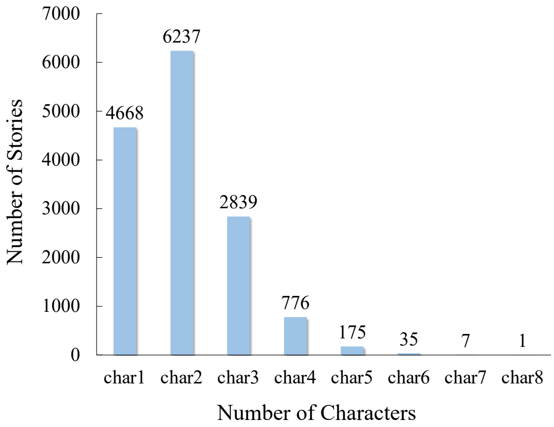

Figure 4: Statistics of stories with different numbers of characters. The "char1" means that the stories have one character.

4.2 Baselines

We compare our model with a carefully selected set of baselines as shown below.

GPT-2 (Radford et al., 2019): GPT-2 is a transformer-based model pre-trained on a very large corpus, which is very commonly used in natural language generation. A lot of works witness a good performance of GPT-2 in dialogue, story and other text generation in recent years, demonstrating its auto-regression quality.

BART (Lewis et al., 2020): BART is a transformer-based seq2seq pre-training model with a bidirectional (BERT-like) encoder and an autoregressive (GPT-like) decoder. A lot of text generation tasks, like neural machine translation and automatic summarization, can achieve effective results by fine-tuning on BART.

SoCP (Xu et al., 2020): Stories with multi-characters and multi-psychology generated by SoCP can change with the emotional lines of assigned characters. In addition, SoCP can generate stories with different emotional intensities. It also designe a metric to evaluate the accuracy of controlling emotions of roles.

Stylized-Story-Generation (SSG) (Kong et al., 2021): SSG can generate stories with specified style given a leading context by first planning the stylized keywords and then generating the whole story with the guidance of the keywords. Two story styles are considered in SSG, including emotion-

Models	PPL ↓	B-1 ↑	B-2 ↑	D-1 ↑	D-2 ↑	ACC ↑
SoCP	101.33	22.93	7.32	0.478	0.650	0.893
SSG	15.22	26.74	10.77	0.573	0.904	-
GPT-2	29.21	21.30	6.35	0.744	0.960	-
BART	14.00	24.15	7.93	0.729	0.964	-
CHAE	**11.58**	27.10	10.20	0.750	0.971	**0.941**
w/o copy	11.65	26.53	9.67	**0.754**	**0.972**	0.879
w/o emo	11.75	**27.51**	10.61	0.732	0.965	-
w/o copy w/o emo	11.70	27.19	**10.79**	0.735	0.967	-

Table 2: Automatic metrics.

Setting	B-1 ↑	B-2 ↑	D-1 ↑	D-2 ↑
Greedy	31.08	15.65	0.586	0.797
Beam=2	31.96	15.61	0.595	0.811
Beam=3	**32.04**	**15.89**	0.584	0.804
Beam=4	31.86	15.84	0.578	0.801
Beam=5	31.60	15.40	0.576	0.802
Top-k=30, Temperature=0.8	30.86	13.63	0.697	0.939
Top-k=50, Temperature=0.8	30.58	13.64	0.702	0.939
Top-k=30, Temperature=1	29.36	12.00	0.720	0.957
Top-k=50, Temperature=1	29.22	11.94	0.730	0.962
Top-k=30, Temperature=1.2	27.68	10.49	0.745	0.968
Top-k=50, Temperature=1.2	27.10	10.20	**0.750**	**0.971**

Table 3: Decoding strategies adjustment.

driven and event-driven stories.

4.3 Implementation Details

We build our model based on BART using the Huggingface's Transformers library in Pytorch (Wolf et al., 2019). We initialize our model with the public checkpoint of bart-large-cnn [1]. The batch size during training is 8. We use the AdamW optimization (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and the initial learning rate is $5e - 5$. According to the statistics of characters in the stories shown in Figure 4, most stories contain two characters, so we fix the number of characters in each story k to 2. The hyper-parameter λ defaults to 1.0.

For all models, We generate stories by using top-k sampling (Fan et al., 2018) with $k = 50$ and a softmax temperature of 0.8.

4.4 Automatic Evaluation

Evaluation Metrics We use the following metrics for automatic evaluation: (1) **Perplexity (PPL)**: PPL represents the general quality of the generated stories, which estimates the probability of sentences according to each word. (2) **BLEU (B-n)** (Papineni et al., 2002): We use BLEU to compare the coverage n-gram in the candidate stories and the reference stories because the words in $Chae$ usually appear in the reference stories. (3) **Distinct (D-n)**: (Li et al., 2016): Distinct is used to evaluate generation diversity by calculating the percentage of unique n-grams. (4) **Accuracy of emotions (ACC)**: We use emotion labels to calculate the accuracy of the emotions of generated sentences to reflect the controllability of emotion.

Results Table 2 shows the automatic evaluation results. All baseline models are trained on our dataset ROCStory. Our model achieves the lowest Perplexity, which reflects the high quality of the stories generated by CHAE. Besides, the BART-based models (BART and SSG) are better than the GPT-2 model. The traditional seq2seq model SoCP has worse performance. The greater BLEU scores of CHAE and its varieties imply stories closer to the golden truth, proving that the introduction of fine-grained control, mainly by means of the special input $Chae$, is conducive to improving the quality of the generated stories. As for Distinct, the pre-trained models show excellent performance compared with the traditional seq2seq model. Our model attains the best score, which demonstrates CHAE's ability in generating more diverse stories. We further compare the ACC with SoCP, and our model gets higher emotion accuracy.

Additionally, we also explore some decoding strategies including greedy search, beam search, top-k, and temperature, as shown in Table 3. The BLEU scores reach the best when we use the beam size equal to 3, while we get the best Distinct scores with the top-k equal to 3 and the temperature equal to 1.2. The BLEU scores decrease with the increase of beam size. The BLEU scores with top-k are lower than beam search, but the Distinct scores are higher. When top-k is fixed, the higher the temperature, the lower the BLEU and the higher the Distinct. When the temperature is fixed, the higher the top-k, the lower the BLEU and the higher the Distinct.

4.5 Ablation Studies

As shown in Table 2, we also conduct the ablation studies to verify the effectiveness of the additional control methods in our model. When we remove the copy mechanism **(w/o copy)**, the BLEU decreases and the Diversity increases, which suggests that the copy mechanism is beneficial in controlling content, but it also affects the diversity at the

[1] https://huggingface.co/facebook/bart-large-cnn/tree/main

Models	Fluency			Coherence			Informativeness		
	Win(%)	Lose(%)	Tie(%)	Win(%)	Lose(%)	Tie(%)	Win(%)	Lose(%)	Tie(%)
CHAE vs. SSG	**56.0**	30.0	14.0	**52.0**	30.0	18.0	**90.0**	6.0	4.0
CHAE vs. BART	**40.0**	40.0	20.0	**38.0**	36.0	26.0	**46.0**	42.0	12.0
CHAE vs. GPT2	**38.0**	36.0	26.0	**40.0**	40.0	20.0	**72.0**	12.0	16.0
Controllability of CHAE	**65.0%**								

Table 4: Human evaluation results. It shows the percentage of win, lose and tie of CHAE compared with other baselines. The controllability of CHAE indicates the average percentage of sentences in a story that can be controlled by control condition $Chae$.

Context	Jessica had to go to the city.
SoCP	She was very excited to see a new . She was very proud of her friends . She was very happy . She was happy she was going to get .
GPT-2	She left her friends and their cars behind. They waited outside the station for her. She knew she'd go to a bar one day. Unfortunately they all stayed away.
BART	She told her mom to take a bus. The bus didn't have enough time to get back to the stations. When she came back her mother was upset. Jessica was upset that everyone didn't believe her to go now.
SSG	She was leaving the bus at ten o'clock. She saw a traffic light coming up in the distance. But she got in the way quickly. She was late late to her bus stop so she had to wait.
Chae	1. $\langle SEP\rangle$ $\langle soc\rangle$ Jessica $\langle soa\rangle$ to learn new things $\langle sep\rangle$ to see the **museums** $\langle sep\rangle$ to learn something $\langle soe\rangle$ **joy** 2. $\langle SEP\rangle$ $\langle soc\rangle$ Jessica $\langle soa\rangle$ to go out to some interesting place $\langle soe\rangle$ joy 3. $\langle SEP\rangle$ $\langle soc\rangle$ Jessica $\langle soa\rangle$ to save the **artwork** $\langle sep\rangle$ to **remember** what she had seen $\langle soe\rangle$ joy 4. $\langle SEP\rangle$ $\langle soc\rangle$ **Jessica** $\langle soa\rangle$ to **photoshoot** $\langle sep\rangle$ **to see the artistic beauty** inside the famous museum $\langle soe\rangle$ joy
CHAE	She had always **enjoyed** going to a **museum**. However, she could not walk with better to the **museum**. She had to **remember** all her **paintings** from this one place. **Jessica noticed an amazing view** that her **camera** had of a gallery.
Gold	She went to some museums. She really enjoyed the artwork at the MET. She took a lot of photographs. She was there for hours.

Table 5: Case study of the generated stories by our model and the baselines. The $Chae$ represents the assigned control conditions including characters, actions and emotions.

same time. In addition, we also observed that the copy mechanism can help the model to improve the accuracy of emotions. On the contrary, when we remove the emotion loss **(w/o emo)**, the BLEU increases and the Diversity decreases, which reflects that the emotion loss does improve the diversity of the stories, but sacrifices controllability. Furthermore, we remove both the copy mechanism and the emotion loss **(w/o copy w/o emo)**, which means just introducing the conditions $Chae$ to the vanilla BART. The results are still better than BART, illustrating the benefit brought by the fine-grained control conditions.

In general, the copy mechanism and the emotion loss complement each other. The results show that the integrated model (CHAE) can obtain good scores and relatively balance on BLEU and Diversity, and has the best performance on PPL and ACC.

4.6 Human Evaluation

We conduct a human evaluation to compare CHAE with baselines on the following three metrics. (1) **Fluency**: The fluency of a sentence can reflect the quality of intra-sentence. (2) **Coherence**: The coherence of the story can reflect the cohesion of context and the quality of inter-sentence. (3) **Informativeness**: The good performance of a story in terms of informativeness indicates that there are a variety of rich words in the story. We recruit six annotators and divided them into two groups to annotate 50 stories randomly sampled. Each story is annotated by three workers to ensure fairness. The workers have two tasks: one is to compare the results generated by CHAE and other baselines, and the other is to score the controllability of CHAE according to whether the predicted sentences adhere to the assigned conditions $Chae$. The majority

Context	A polite thief was making robberies in the small town.
Chae1	$\langle SEP\rangle$ $\langle soc\rangle$ People $\langle soa\rangle$ $\langle no\_action\rangle$ $\langle soe\rangle$ fear $\langle SEP\rangle$ $\langle soc\rangle$ **Man** $\langle soa\rangle$ to catch the thief $\langle soe\rangle$ **anger**
Result1	One day, a **man** walked up to him and **asked him to stop** .
Chae2	$\langle SEP\rangle$ $\langle soc\rangle$ People $\langle soa\rangle$ $\langle no\_action\rangle$ $\langle soe\rangle$ fear $\langle SEP\rangle$ $\langle soc\rangle$ **Man** $\langle soa\rangle$ $\langle no\_action\rangle$ $\langle soe\rangle$ **joy**
Result2	The **man** who was supposed to stop him was a **nice man** .
Chae3	$\langle SEP\rangle$ $\langle soc\rangle$ People $\langle soa\rangle$ $\langle no\_action\rangle$ $\langle soe\rangle$ fear $\langle SEP\rangle$ $\langle soc\rangle$ **Tom** $\langle soa\rangle$ **to catch the thief** $\langle soe\rangle$ **anger**
Result3	**Tom** decided to investigate and **caught the thief** .
Chae4	$\langle SEP\rangle$ $\langle soc\rangle$ **People** $\langle soa\rangle$ **call the police** $\langle soe\rangle$ **fear** $\langle SEP\rangle$ $\langle soc\rangle$ **Tom** $\langle soa\rangle$ **call the police** $\langle soe\rangle$ **anger**
Result4	**Tom** **called the police** and **they** **told him** to **call the police** .

Table 6: Case study of controllability.

votes among the annotators will be the final decisions for the first task and we average the scores of all annotators for the second task. The results are shown in Table 4. Our model achieves the best performance in each metric compared with baselines and has 65% controllability.

4.7 Case Study

Comparison Table 5 shows the stories generated from our model and the baselines. The story generated by the SoCP model reflects that the results generated by the models built on the seq2seq architecture are usually simple, short and less informative. The transformer-based models, such as GPT-2, BART, and SSG, show strong generative capabilities, which can generate coherent and informative stories. However, the content of these stories can not be controlled, and the characters' emotions and actions in the stories are not obvious.

For our model, we can specify the characters, actions and emotions in the story that we desired, and the model can generate stories based on this information, i.e. $Chae$. The meaning of the four $Chae$ is to control the content of the next four sentences respectively. The first sentence reflects that the character's action is to go to the "museum", and the word "enjoyed" reflects the emotion of "joy". The word "remember" and "artwork" in the third sentence corresponds to "remember" and "paintings" in $Chae3$. The model is connected to the "camera" from the "photoshoot" in $Chae4$, and the phrase "noticed an amazing view" corresponds to "to see the artistic beauty", and the character "Jessica" appears in the sentence. The results show that our model can generate coherent and informative stories according to the control information.

Controllability Table 6 shows the examples of controllability. We use beam search (beam=2) to generate results instead of top-k, because the top-k method usually generates diverse words, which can not ensure controllability. Our model generates sentences based on the same context and the given different $Chae$. The first two examples show that for the same character "Man", given the emotion "anger" and "joy", the model can generate sentences with corresponding emotions, such as the phrase "asked him to stop" and "he was a nice man". In the third example, we change the character to "Tom", and the generated sentence is also changed from "Man" to "Tom", and the phrase "caught the thief" also reflects the action "to catch the thief". In the last example we change Tom's action to "call the police", and the resulting change from "caught the thief" to "called the police". Also, we set people's actions as "call the police" and the emotion is fear, which results in the expression "they told him to call the police." The above reflects that our model has a good control effect on characters, actions and emotions.

5 Conclusion and Future Work

Through our model CHAE, we can create stories with fine-grained control according to the specified characters and corresponding actions and emotions, which is more convenient for practical applications, such as the creation of script novels, providing inspiration for screenwriters, and even acting as screenwriters in the future.

However, our model also has some disadvantages: (1) The dataset contains only stories with 5 sentences, which is not enough for learning to generate longer stories. (2) The training of our model heavily relies on the annotations of characters, emotions, and actions in the dataset, while it is very expensive to obtain the annotated data. (3) The $Chae$ in the dataset actually has some noise, i.e., some descriptions in $Chae$ are not reflected in the corresponding sente nces. (4) Our iterative generation method will result in a long time to train a story and may cause a cascade problem, which affects the overall quality of story generation.

In future work, we will adopt datasets with much more data and longer stories, such as **WritingPrompts** (Fan et al., 2018) and **WikiPlots**[2]. In addition, we will consider using commonsense reasoning techniques to reason about the emotions and actions of the characters in the story before further generating the story. Regarding the noise of $Chae$, we plan to conduct denoising in preprocessing to filter out the samples whose $Chae$ are inconsistent with the corresponding sentence. The problem can also be alleviated by dynamically controlling the weight of the conditions. We are also further exploring more convenient and effective training methods to generate controllable stories by inputting control conditions in one go, rather than iterative generation.

Acknowledgements

The work is partially supported by the National Nature Science Foundation of China (No. 61976160, 61906137, 61976158, 62076184, 62076182) and Shanghai Science and Technology Plan Project (No.21DZ1204800) and Technology research plan project of Ministry of Public and Security (Grant No.2020JSYJD01).

References

Amal Alabdulkarim, Siyan Li, and Xiangyu Peng. 2021. Automatic story generation: Challenges and attempts. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 72–83, Virtual. Association for Computational Linguistics.

Faeze Brahman and Snigdha Chaturvedi. 2020. Modeling protagonist emotions for emotion-aware storytelling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5277–5294, Online. Association for Computational Linguistics.

Alvin Chan, Yew-Soon Ong, Bill Pung, Aston Zhang, and Jie Fu. 2020. Cocon: A self-supervised approach for controlled text generation. In *International Conference on Learning Representations*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Jon Deaton, Austin Jacobs, Kathleen Kenealy, and Abigail See. 2019. Transformers and pointer-generator networks for abstractive summarization.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Le Fang, Tao Zeng, Chaochun Liu, Liefeng Bo, Wen Dong, and Changyou Chen. 2021. Outline to story: Fine-grained controllable story generation from cascaded events. *arXiv preprint arXiv:2101.00822*.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108.

Jian Guan, Yansen Wang, and Minlie Huang. 2019. Story ending generation with incremental encoding and commonsense knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6473–6480.

Tatsuya Ide and Daisuke Kawahara. 2021. Multi-task learning of generation and classification for emotion-aware dialogue response generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 119–125, Online. Association for Computational Linguistics.

Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. 2020. Language generation with multi-hop reasoning on commonsense knowledge graph. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 725–736.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Xiangzhe Kong, Jialiang Huang, Ziquan Tung, Jian Guan, and Minlie Huang. 2021. Stylized story generation with style-guided planning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2430–2436, Online. Association for Computational Linguistics.

[2] https://github.com/markriedl/WikiPlots

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. *Theories of emotion*, 1:3–31.

Nikhil Prabhu and Katharina Kann. 2020. Making a point: Pointer-generator transformers for disjoint vocabularies. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 85–92.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Hannah Rashkin, Antoine Bosselut, Maarten Sap, Kevin Knight, and Yejin Choi. 2018. Modeling naive psychology of characters in simple commonsense stories. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2289–2299, Melbourne, Australia. Association for Computational Linguistics.

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. PlotMachines: Outline-conditioned generation with dynamic plot state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4274–4295, Online. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Satoshi Tsutsui and David J. Crandall. 2017. Using artificial tokens to control languages for multilingual image caption generation. *CoRR*, abs/1706.06275.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Feifei Xu, Xinpeng Wang, Yunpu Ma, Volker Tresp, Yuyi Wang, Shanlin Zhou, and Haizhou Du. 2020. Controllable multi-character psychology-oriented story generation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1675–1684.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.