Story Ending Generation using Commonsense Casual Reasoning and Graph Convolutional Networks

Eunkyung Park[a;*], **Raymond K. Wong**[a] **and Victor W. Chu**[b]

[a]UNSW Sydney
[b]University of Technology Sydney

Abstract. Story Ending Generation is a task of generating a coherent and sensible ending for a given story. The key challenges of this task are i) how to obtain a good understanding of context, ii) how to capture hidden information between lines, and iii) how to obtain causal progression. However, recent machine learning models can only partially address these challenges due to the lack of causal entailment and consistency. The key novelty in our proposed approach is to capture the hidden story by generating transitional commonsense sentences between each adjacent context sentence, which substantially enriches causal and consistent story flow. Specifically, we adopt a soft causal relation using people's everyday commonsense knowledge to mimic the cognitive understanding process of readers. We then enrich the story with causal reasoning and utilize dependency parsing to capture long range text relations. Finally, we apply multi-level Graph Convolutional Networks to deliver enriched contextual information across different layers. Both automatic and human evaluation results show that our proposed model can significantly improve the quality of generated story endings.

1 Introduction

Automated storytelling is an important yet challenging problem in Natural Language Processing as it needs to cater for the logical flow of a story within the context and external commonsense knowledge [21, 20, 15, 48, 24]. The story ending generation (SEG) task includes concluding a story and completing the plot with a proper causal flow.

Previous SEG research works mainly focus on the symbolic planning method. For example, [13, 32, 35, 43] conducted reasoning directly for causality using the form of predicate precondition and postcondition matching. However, their abilities to learn extensive domain knowledge, the vocabulary of events, and their characters are limited.

On the contrary, machine learning approaches can overcome those limitations by learning a corpus of existing stories or plot summaries. They learn probabilistic relationships between words, sentences, and events.

However, they need help in modelling causal entailment and maintaining consistency. Recently, many published works have been using Sequence-to-Sequence (Seq2Seq) model [25]. Nevertheless, as the Seq2Seq generates sentences in a single direction (e.g., in a left-to-right manner) and only optimises the model using a maximum likelihood estimate, they show limitations in achieving coherence and causality.

* Corresponding Author. Email: eunkyung.park@unsw.edu.au.

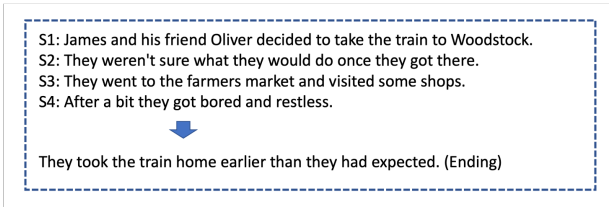

Figure 1. An example of the SEG task. Our proposed model enriches the story using causal commonsense reasoning from the four sentences to make better story endings. We do a story augmentation process for the first and second sentences, the second and third sentences, and the third and fourth sentences.

Recently, Li et al. [17] proposed a model that uses multi-level Graph Convolutional Networks over dependency trees to capture hidden clues in the context. Due to the enhanced capturing of context information, the generated endings are clearer. However, there is still a big gap between machine-generated and human-crafted endings outputs regarding causality and abundance. One reason is that the proposed method relies only on the relation between internal context sentences. The other possible reason is insufficient causality. Good story endings require capturing internal context and language understanding using the everyday commonsense knowledge of readers. Most of the time, people rely on their own experiences and implicit knowledge to understand a story.

One of the main challenges of SEG is the causal progression of the flow of a story. For example, the ending story should be connected to the previous one and can be understood by causal relation. In this paper, we adopt soft relation causality using everyday commonsense. Figure 1 shows an example of the SEG task. We have the given four sentences. Given the four sentences, we aim to generate a proper ending for the given texts. To make appropriate endings, we need to capture the internal context clue and hidden messages via external commonsense knowledge. Hence, we do story augmentation using causal commonsense knowledge. We enrich the text by generating additional texts between the first two sentences. We repeat the same process for the second and third sentences and the third and fourth sentences. This process tracks the reader's understanding process. When reading a story, people commonly apply the context and their commonsense knowledge together while reading.

This paper considers the causality between sentences and external commonsense knowledge. First, we enrich the given stories by story augmentation using story infilling, which is based on the *soft causal relations*. Story infilling [19] is inspired by *plot infilling* where an

outline of plot points is extracted from the source, and then the extracted plot points will be added to the story plots.

We query commonsense causal inferences from COMET [5] to build a graph and search the graph space via common sense knowledge reasoning. Commonsense knowledge is a set of commonly shared knowledge about how the world works. It allows us to expect what is going on if we conduct a particular behaviour and what was likely to happen in the past. Therefore, commonsense reasoning can be used for inference of the flow of stories.

Therefore, we can build a branching space of possible story continuations connecting sentences. Once the sentence graph has been constructed, we search for complete sequences. The previous work C2PO [1] proves the effectiveness of soft causal relations' effectiveness. It can improve causality between story events by applying people's everyday commonsense understanding rather than strict logical consistency. Next, we apply dependency parsing to capture information from non adjacent sentences into a dependency tree, especially targeting long range sentences. Dependency trees are already examined to effectively extract features from texts [17, 47]. Moreover, we apply forward Multi-level Graph Convolutional Networks over dependency parse trees to conduct the dependency relations of input sentences.

The key novelty in our proposed model is to capture the hidden story by generating transitional commonsense sentences between adjacent context sentences, which substantially enriches causal and consistent story flow. We achieve this by the following contributions:

- Utilizing COMET for story augmentation;
- To our best knowledge, we are the first research work to apply and investigate the effect of the recent state-of-art model C2PO for story argumentation;
- We explore the synergy between the story augmentation and multi-level GCNs and find the optimal amount of story augmentation for story ending generation in our ablation study.

Experiments show that our model can significantly improve the quality of generated story endings based on both automatic and human evaluations. In particular, it confirms that our approach effectively captures the hidden context by combining the internal context and causal commonsense reasoning. Our ablation study also demonstrates that story augmentation improves causal and cohesive endings by enhancing story quality and enjoyable ability.

2 Related Work

2.1 Story Ending Generation (SEG)

Most previous works regard the story context as a sequence of words and ignore the rich relations among them. In particular, Li et al. [22] applied a Seq2Seq and adversarial training. Gual et al. [15] adopted incremental encoding. Wang et al. [42] applied a modified Transformer model to build the contextual clues, and variational autoencoder for diversity and coherence. Other works applied control sentiment and attribute to increase the diversity of the ending [30, 16].

In general, neural network-based models overlooked the importance of causality. MGCN-DP is a proposed multi-level Graph Convolutional Networks and introduces Dependency Trees into the model to increase causality by capturing hidden clues from long range sentences. It suggests the great importance of story clues hidden in the context. However, it mainly relies on capturing internal context information using dependency trees and does not consider commonsense knowledge behind the story.

This paper enriches the context by story augmentation using causal commonsense knowledge based on the soft causal relation from people's everyday commonsense knowledge.

2.2 Story Augmentation

We adopt story infilling literature to enrich a text using commonsense causal reasoning. Text infilling [40] is a task removing sequences of words from a text and asking for a replacement. Fedus et al. [12] used the masking of random words. Collobert et al. [8] and Devlin et al. [9] used contextualized word embeddings. Sun et al. [37] used bi-directional decoding for image captioning.

Ippolioto et al. [19] suggested a task filling in missing parts from a story by conditioning a text generator on rare words. Donahue et al. [10] attempted to fill in the blanks given left and right contexts. C2PO [1] incorporates commonsense knowledge into this task by applying *soft causal relation* and plot graph learning, which is initially inspired by [21]. Du et al. [11] proposed an autoregressive pretrained language model for blank infilling and Xiao et al. [44] proposed an interactive machine translation model for a bilingual text infilling method. They published work on a Bilingual Text Infilling Method for Interactive Machine Translation (BiTIIMT). Du et al. [11] published a pretraining framework for three main categories - natural language understanding (NLU), unconditional generation, and conditional generation. To address the challenge, it proposes a General Language Model (GLM) based on autoregressive blank infilling. In this paper, we apply these infilling methods for story infilling.

3 Proposed Model

3.1 Overall Framework

Figure 2 illustrates our model. We have two stages: 1) story augmentation using soft causal relation and 2) multi-level Graph Convolutional Networks (GCNs). From the story augmentation, we generate additional context sentences using commonsense knowledge. The enriched context is fed as multi-level Graph Convolutional Networks input for information delivery. We use a dependency parse tree to capture long range dependency and prune unrelated information.

Given a story context consisting of a sentence sequence, the SEG task can be formulated as follows:

$$X = X_1, X_2, \cdots, X_\mu$$

where $X_s = x_1^{(s)} x_2^{(s)} \cdots x_n^{(s)}$ contains n words in the sth sentence. This task aims to generate a story ending Y related to the given context X. Therefore, we can formalize a one-sentence ending $Y = y_1 y_2 \cdots y_l$ as follows:

$$Y^* = \underset{Y}{\operatorname{argmin}}\, Pr(Y|X).$$

As illustrated in Figure 2, our model has 2 phases. The first part is *story augmentation* to enhance context by generating additional sentences using commonsense knowledge reasoning. Given the task, we have four sentences. We generate additional sentences $x_{\text{start:end}_i}$ from the two adjacent sentences x_{start} and x_{end} where start $\in \{1, 2, 3\}$ and end $=$ start $+ 1$.

This is from the idea that when readers read a story, they understand the content, not only sentences in the book but also use their commonsense knowledge. We use soft causal relations using daily commonsense reasoning inspired by C2PO. Our contribution is introducing a method of story augmentation using COMET. To our best

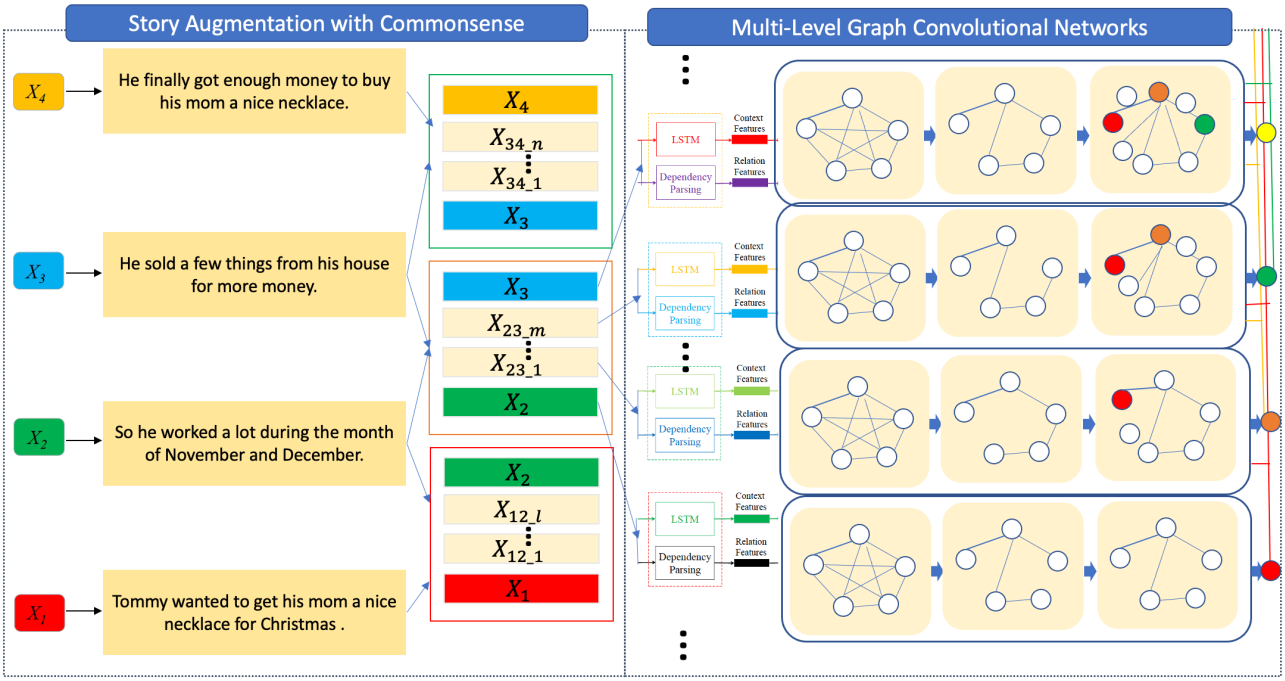

Figure 2. An illustration of our proposed model for SEG task. The model has 1) a story augmentation stage using causal commonsense and 2) multi-level Graph Convolutional Networks (GCNs) for information delivery. The enriched context by story augmentation is an input of multi-level Graph Convolutional Networks. The final outputs are the encoder's input to generate proper story endings.

knowledge, we are the first research work to apply and investigate the effect of the recent state-of-art model C2PO method for story argumentation. We use these enriched stories as inputs for multi-level GCNs.

The second part is based on an encoder-decoder architecture. We use Stanford Dependency parser [6] to parse dependency relations. We use Tree LSTM [38] to build a story graph from each input sentence. We prune some unrelated edges to obtain the sparse graphs. We apply MGCN-DP [17] to build graphs and update each node by aggregating information from the neighbours. The MGCN-DP has not been applied for story argumentation in other related works.

3.2 *Subject Extraction*

Before story augmentation, we extract the subject to generate sentences. We use coreference resolution [7] and information extraction to identify the subject for the sentences. First, we use a pre-trained neural coreference resolution model to extract all the coreference clusters. The clusters include all the mentions in the story belonging to a single possible character. We randomly select one of them and let $M = \{m_1, m_2, ..., m_n\}$. Also, we extract a set R of <subject, relation, object> triples from the story text using OpenIE [3]. Next, we align them to find a subset of triple $P \subset R$ relevant for a single character based on their character-level positions within the original story text. Let $pos(.)$ be a function to do this. We randomly select the subject among the subject and object to generate additional sentences.

3.3 *Story Augmentation using Commonsense Casual Reasoning*

We build two directed acyclic graphs to enrich the story between x_{start} and x_{end}. x_{start} and x_{end} are adjacent sentences in the given text. We recursively query COMET [5] to p sentence candidates q times starting from x_{start}. We define this as G^f. The $wants$ relation is a direct forward cause, meaning a character has wanted and therefore acts on for this. Also, we recursively query COMET to generate p sentences q times starting x_{end}. We define this as G^b. For backward entailment, we use $needs$ relation. Note that $needs$ relation means a character needs something to be true to act on. The relations in G^f and G^b are weighted proportional to the likelihood by COMET for each inference.

COMET is a transformer-based language model designed for commonsense inference. It is trained on ATOMIC [36], a dataset containing 877k instances of information relevant for everyday commonsense reasoning with the form of if-then relation types.

Next, we follow [1] to find the optimal way to connect G^f and G^b. The link's weight is defined as follows:

$$w(u,v) = \frac{Pr^{wants}(u|v)}{\alpha_u^{wants}} + \frac{Pr^{needs}(u|v)}{\alpha_v^{needs}},$$

where $Pr^{needs}(u|v)$ is the probability of generating sentence x_{end} as inferences by COMET under the $needs$ relation, conditioned on x_{start}. $Pr^{wants}(u|v)$ is calculated in the same by but under the $wants$ relation. α_u^{wants} and α_u^{needs} are normalized constants. This process is repeated for all nodes until a set of optimal links is found.

Therefore, we can finalise the entire story graph as follows:

$$G = \bigcup_{x_{\text{start}}, x_{\text{end}}} (G^f_{x_{\text{start}}} \cup G^b_{x_{\text{end}}}), \forall x_{\text{start}}, x_{\text{end}} \in P,$$

where $x_{\text{start}}, x_{\text{end}}$ are adjacent in X.

Finally, we can link the sentence graphs for the entire sentences of the story. A random graph walk can generate a story from the first sentence x_{start} to x_{end}. All random walks are guaranteed to terminate x_{end} as $G^{b_x}_{\text{end}}$ is built by branching backward from x_{end}.

This way, we generate l sentences from X_1 and X_2. We generate m sentences from X_2 and X_3. We generate n sentences from X_3 and X_4. Finally, we generate context-enhanced $l + m + n$ sentences from the initial four sentences.

3.4 Commonsense Graph Construction

Given a sth sentence $X_s = x^{(s)}_1 x^{(s)}_2 \cdots x^{(s)}_n$ with n words, we represent the tth word $x^{(s)}_t$ by Glove [31] word embedding as follows:

$$e^{(s)}_t = e^w(x^{(s)}_t),$$

where e^w denotes a word embedding lookup table and $e^{(s)}_k$ is the embedding vector of tth word $x^{(s)}_t$ in the sth sentence. We apply LSTM to obtain the word representation h^s_{wt}:

$$h^{(s)}_{wt} = \text{LSTM}(e^{(s)}_t).$$

Each edge represents a particular relation between two words, and each word represents a vertex. We can define an intra-sentence graph G^I as follows:

$$G^I = (\nu^I, \xi^I),$$

where ν^I is the set of nodes and ξ^I is the set of edges connected with the nodes.

3.5 Dependency Relations

We build dependency relations between words by parsing sentences. We remove unrelated edges and obtained a dependency sparse graph G^P as follows:

$$G^P = (\nu^P, \xi^P),$$

where ν^s is the set of nodes of the pruned graph and ξ^s is the set of edges connected with the nodes.

3.6 Intra-sentence Information

Next, we perform the attention-based GCNs [45, 18] node aggregation and updating. Dependency sparse graph G^p with a $n \times n$ adjacency matrix is used, where a fully connected layer reflects the relations between nodes. For a target node i and its neighbour nodes set $N(i)$, the representing of node i and node $j \in N(i)$ are h_{wi} and h_{wj}. We calculate the correlation score $w_{i,j}$ between node i and node j as follows:

$$w_{ij} = w^T_0 \sigma(W_0[h_{wi}; h_{wj}] + b_0),$$

where w_0, W_0, and b_0 are trainable parameters, σ is the non-linear activation function. h_{wi} and h_{wj} denote the concatenation. We calculate the weight α_{ij} using a softmax function over the correlation score w_{ij} as follows:

$$\alpha_{i,j} = \frac{\exp(w_{ij})}{\sum_{j \in N(i)} \exp(w_{ij})}.$$

The i-th representation of neighbor node $h^{(l)}_{wj}$ is first transformed using a learned linear transformation layer W_1 as follows:

$$h^{l+1}_{wi} = \sigma(h^{(l)}_{wi} + \sum_{j \in N(i)} \alpha_{ij}(W_1 h^{(l)}_{wj} + b_1)),$$

where W_1 and b_1 are trainable parameters. The output H_w of the stacked l layer GCNs is

$$H_w = h^{(l+1)}_{wi}.$$

3.7 Multi-level GCN for Story Augmentation

From the story infilling stage, we generate $l + m + n$ sentences. Among them, we select the first p sentences.

We apply multi-level GCNs [17] on the graph to represent L-level GCNs graph.

$$G^L = (\nu^L, \xi^L),$$

where G^L is the set of the L-th level GCNs node, ν^L denotes the set of the set of the L-th level GCNs nodes and ξ^L is the set of the L-th level GCNs edges.

For the s-th sentence X_s with n words, all words can be represented $[h^{(s)}_{w1} \cdots h^{(s)}_{wn}]$. The node set ν^L in L-th level GCNs is

$$\nu^L = [h^{(s)}_{w1} \cdots h^{(s)}_{wn}].$$

For information delivery across different levels, We weigh each node using the attention mechanism in ν^L and sum them together as a new node $h^{(L)}_a$:

$$\beta = \text{softmax}(W_2 \nu^L + b_2),$$

$$h^{(L)}_a = \sum_{L=1}^{n} \beta \nu^L,$$

where W_2 and b_2 are trainable parameters.

For the $(s+1)$-th sentence X_{s+1}, we can build word embedding $[h^{(s+1)}_{w1} \cdots h^{(s+1)}_{wm}]$.

Then we combine $[h^{s+1}_{w1} \cdots h^{(s+1)}_{wm}]$ with $h^{(p)}_a$ as the nodes set ν^{L+1} of the $(L+1)$-th level GCNs:

$$\nu^{L+1} = [h^{(s+1)}_{w1} \cdots h_{wm^{(s+1)}}; h^1_a, \cdots, h^L_a].$$

Given a graph with $(m + L)$ nodes, for the graph structure G^{L+1}, the $(m + L) \times (m + L)$ adjacency matrix is used. For a target node i and a neighbour node $j \in \Psi(i)$ in the L_1-th level graph G^{L+1}, $\Psi(i)$ is the set of nodes neighbouring with node i. We calculate the correlation score λ_{ij} between node i and node j as follows:

$$\lambda_{ij} = w^T_3 \sigma(W_3[h_{Li}; h_{Lj}] + b_3),$$

where w_3, W_3, and b_3 are trainable parameters. σ is the non-linear activation function, and $[h_{Li}; h_{Lj}]$ denotes the concatenation operation. The weight ϕ can be calculated using the softmax function over the correlation score λ_{ij} as follows:

$$\phi = \frac{\exp(\lambda_{ij})}{\sum_{j \in \Psi_i} \exp(\lambda_{ij})}.$$

Table 1. Automatic evaluation and human evaluation. We bold our model and underline the best results.

Model	BLEU-1	BLEU-2	METEOR	ROUGE-1	ROUGE-2	ROUGE-L	Grammar	Logic
Seq2Seq [25]	18.5	5.9	12.1	20.3	2.5	21.2	2.57	1.41
Transformer [41]	17.4	6.0	11.9	19.8	2.3	20.9	2.54	1.62
GCN [45]	17.6	6.2	11.8	19.9	2.5	21.3	2.62	1.70
IE+MSA [15]	24.4	7.8	13.2	23.2	2.7	23.1	2.64	1.80
T-CVAE [42]	24.4	8.4	13.3	23.5	2.7	23.2	2.65	1.73
Plan&Write [46]	24.4	8.4	13.3	23.4	2.6	23.1	2.65	1.73
GPT2 [34]	23.0	7.3	13.1	22.9	2.6	22.8	2.69	1.85
KE-GPT2 [14]	26.5	9.4	16.1	25.7	2.9	26.8	2.65	1.92
MGCN-DP [17]	24.6	8.6	18.8	28.4	3.2	27.8	2.67	1.86
ChatGPT (https://chat.openai.com)	27.0	9.7	20.1	30.1	3.2	28.3	2.85	1.94
CCRGCN (ours)	**27.2**	**9.9**	**20.4**	**31.5**	**3.4**	**29.0**	**2.71**	**1.97**

The l-th representation of neighbour nodes $h_{Lj}^{(l)}$ are first transformed using a linear transformation layer W_4. Those transformed representations are gathered with the weight phi_{ij}, followed by a non-linear function σ. This propagation process is denoted as follows:

$$h_{Li}^{(l+1)} = \sigma(h_{Li}^{(l)} + \sum_{j \in \phi(i)} \phi_{ij}(W_4 h_{Lj}^{l+1} + b_4)),$$

where W_4 and b_4 are trainable parameters.

Following the stacked l layer GCNs, the output of the encoder H_L is as follows:

$$H_L = h_{L_i}^{L+1}.$$

3.8 Decoder

We adopt the decoder of transformer [41] for decoding. The input of Multi-Head attention is D_{in},H_L, and H_L, FFN is two linear transformations with ReLU activation in between, and D_o is the middle output of the decoder. Decoding can be denoted as follows:

$$\tilde{D}_{\text{in}} = \text{MultiHead}(D_{\text{in}}, H_L, H_L),$$

$$D_o = \text{FFN}(\tilde{D}_{\text{in}})$$

We can predict the probability of a word using a linear transformation layer and softmax function to convert the output of the decoder. Let z denote the index of the ending sentence. At each time step t, the decoding process is represented as follows:

$$P(y_t|y < t, X) = \text{softmax}(W_z D_o + b_z),$$

where W_z and b_z are trainable parameters and $P(y_t)$ is the probability distribution over vocabulary.

4 Experiments

We evaluate our model using the ROCStories corpus [26]. The dataset contains 90,000 training stories, 4,081 validation stories, and 4,081 test stories. We use standard automatic language generation metric BLEU and human participant study.

4.1 Experimental Settings

GloVe.6B is used as word vectors. The vocabulary size is 10,000, and the word vector dimension is 300. The level of the stacked layer in GCNs is 16. The learning rate is 0.005. The batch size is 64. The head h of attention in the decoder is 6, d_k and d_v are 64. The level of the stacked layer of the decoder is 2. The dropout rate is 0.1. We train the model for 60 epochs.

4.2 Evaluation Metrics

4.2.1 Automatic Evaluation Metric

- BLEU [29] evaluates the n-gram overlap between generated endings and a reference. We report BLEUs with $n = 1, 2$. We calculate each BLEU-1 and BLEU-2 for stories in the test set and obtain the average prediction accuracy.
- METEOR [4] applies a weighted F-score using mapping unigrams and a penalty function for incorrect word order.
- ROUGE (Recall Oriented Understudy for Gisting Evaluation) [23] replies on recall. We calculate each ROUGE-1 and ROUGE-2. For n, we calculate the number of n-grams across all the gold ending text and count how many appear in the candidate gold ending from each model.
- ROUGE-L is based on the longest common subsequence (LCS). Rather than using only recall, it is calculated as the weighted harmonic mean of precision and recall.

4.2.2 Human Evaluation Metric

For human evaluation, we randomly sample 100 stories. We recruit five students who are a) fluent in English and b) demonstrate an understanding of story generation tasks. Human participants are given ten stories generated by GPT2, MCGN-DP, ChatGPT, and our model. The order of stories is randomized to avoid bias due to the ordering effect [27]. At least 3 participants see each story set (3 pairs). We ask them about the quality of Grammar and Logic for the outputs from each model. Grammar evaluates whether the generated story is fluent and natural, while Logic evaluates whether the generated story is reasonable and coherent with the context. We ask the participants to score 1/2/3, where 1 means bad, 2 means okay, and 3 means good.

4.3 Baselines

We adopt the experiment results from MGCN-DP [17], which includes 7 baselines plus MGCN-DP itself, and supplement this set of baselines with GPT2 [34] and ChatGPT [1]. GPT2 is a pre-trained model from web text such as Reddit and Wikipedia with a 1.5 billion parameter for story generation tasks. It has a Transformer architecture. We set the length of the generated story as the average length of the gold ending. We use our test set as inputs for GPT2. ChatGPT is fine-tuned on the top of GPT3.5 [28] using Reinforcement Learning from Human Feedback (RLHF). We use Free Research Preview ChatGPT and the query prompt "Can you predict the last sentence

[1] https://chat.openai.com

Table 2. Generated endings from different models. Bold words denote the keywords in the story. An improper story in endings is italic.

Case 1	
Context	Lizzy's cousin died. She didn't have enough money to fly **home** for the funeral. She told her friend. He gave her the money.
GPT2	And then she *ran away.*
MGCN-DP	She was able to go to the .
ChatGPT	When Lizzy's cousin died and she didn't have enough money to fly home for the funeral, she was devastated and didn't know what to do.
CCRGCN	She went **home** .
Gold Ending	She was able to go **home** .
Case 2	
Context	The kid was in a **spelling contest**. He won a prize for most improved speller. He made it through seven rounds before getting **kicked out**. He was proud of what a good speller he was.
GPT2	*You can't go wrong with this guy.*
MGCN-DP	He **ended up** on the.
ChatGPT	Even though he didn't win the spelling contest, he was still proud of how much he had improved and promised to work harder for next year's competition.
CCRGCN	He **ended up** and **went home**.
Gold Ending	He **went home** and decided to practice more.

from the 4 sentences?" MGCN-DP is based on multi-level GCNs using dependency trees. The model mainly relies on internal context information. We use the same parameters for MGCN-DP and our model to compare results fairly. We give them the four context sentences as input per story.

4.4 Results and Analysis

4.4.1 Results

Table 1 shows that our model outperforms most baselines on automatic and human evaluation. Our model achieves significant improvements over other baselines. These results indicate that our model generates story endings that overlap with the gold endings from the generating story endings. Also, our model shows promising results from human evaluation. In particular, for the question regarding Grammar, our model shows the second highest score (2.71). As ChatGPT is a state-of-art pre-trained language generation model with human AI trainers using huge conversations with a chatbot, it shows better results. However, regarding Logic, our model obtains the highest score, 1.97.

4.4.2 Analysis

We present examples of the generated story endings. Table 2 shows the 4 given context sentences, endings generated from the models, and gold endings. From the given 4 sentences "Lizzy's cousin died. She didn't have enough money to fly home for the funeral. She told her friend. He gave her the money.", our model generates the best sentence compared to the gold ending and grammar. Our model generates a good ending, "She went home." with good grammar and the keyword $home$. However, MGCN-DP generates an ending "She was able to go to the" with bad grammar and missing a keyword $home$. GPT2 and ChatGPT generate good grammar endings, but they are very different from the gold ending.

Case 2 also shows that our model generates the best sentence, "He ended up and went home." considering the given context, "spelling contest" and "kicked out". Our model generates good sentences again in terms of grammar and meaning. However, the ending by MGCN-DP is "He ended up on the." It is poor grammar and missing critical information $home$. The sentence by GPT2 is, "You can't go wrong with this guy." It looks good in terms of grammar. However, it is out of the topic compared to the gold endings. Our model can generate the best sentences from the results by combining the external knowledge reasoning and capturing the internal long range distance.

Table 3. The results of human evaluation for ablation study

	Q1 (%)	Q2 (%)	Q3 (%)
w/o story augmentation	44	24	46
with story augmentation	**56**	**76**	**54**

4.5 Ablation Study

To investigate our proposed model's effectiveness, we conduct an ablation study. First, we compare our model with the baseline without story augmentation to address the effect of story augmentation. Next, we further explore how we achieve our model by varying the number of additional sentences. We conduct a human survey to quantify the effect of story argumentation to infer correct story endings. Next, we investigate the synergy effects between story augmentation and multi-level GCNs with different volumes of story augmentation. We conduct automated evaluation metrics for the cases with different numbers of additional sentences to find optimal values for story augmentation for multi-level GCNs.

4.5.1 Story Augmentation

We conduct an additional human evaluation to investigate the effects of story augmentation. We hire five students with the same qualification as previous human participants. We create a pair of stories. The first story set includes only the original four sentences and gold endings. The second story set consists of the original four sentences, other stories generated by our story augmentation, and gold endings. We ask the participants to select the better one between them. We use the questions proposed by previous work for multiple storytellings [33, 39, 2]. In particular, we use the following questions [1]:

- Q1: Which story is of higher quality?
- Q2: Which story is more enjoyable?
- Q3: Which story is better to predict the gold ending?

4.5.2 Results

We ask the participants to select which one has a higher score for each criterion. Table 3 shows the results. The second story set with story augmentation obtains better results from both questions. For the first question regarding quality, B (with story augmentation) obtains 56% against 44% (A: original four sentences and gold ending). For

Table 4. Examples of a story generated by story augmentation. Initial set sentences are in bold. From the two bold sentences, we generate additional sentences using causal common sense reasoning.

Case 1
Nathan liked hanging out with his friends.
Nathan begins to go to the movies.
Nathan begins to go home.
Nathan starts to smoke.
They would sit around and smoke cigars.
Nathan starts to smoke another cigarette.
Nathan wants to smoke more.
Nathan wants to work.
Nathan starts to have money.
Nathan took a trip to Cuba.
Nathan tries to go to the airport.
Nathan wants to get in the car.
Nathan tries to go to the store.
Nathan starts to purchase cigars.
He bought a lot of cigars there to bring home.
Gold Ending : He couldn't wait to share them with his friends.

Case 2
Fred wanted to try the Paleo diet.
Fred begins to eat healthy food.
Fred starts to exercise.
Fred starts to work hard.
Fred tries to be in charge.
It was all the rage.
Fred tries to calm down.
Fred tries to rest.
Fred begins to think.
Fred begins to think about something.
He thought it would be good.
Fred starts to do something.
Fred begins to rest.
Fred wants to take a nap.
Fred tries to be tired.
But after one day he quit.
Gold Ending : It was too hard not to eat bread.

the second enjoyable question, B (with story augmentation) obtains 76% against 24%. Also, the story with story augmentation obtains a better score (54% vs 46%) for the question regarding effectiveness for generating endings. The study confirms that story augmentation using causal commonsense reasoning effectively generates high-quality and enjoyable stories. Also, it is helpful in generating story endings.

4.5.3 Case Study

Table 4 demonstrates the story augmentation effects using commonsense reasoning and soft causal relation. The initial four sentences are in bold. The sentences between the two bold sentences are generated sentences from story augmentation. The initially given sentences are "Nathan liked hanging out with his friend. They would sit around and smoke cigars. Nathan took a trip to Cuba. He bought a lot of cigars there to bring home." From the first two sentences our model generates additional three sentences. They show causal ordering based on everyday common sense. They use $begin$ and $start$ relation to add more causality between stories. In particular, they add context, "go to the movie" from "hang out". They add the preceding action "start to smoke" before "sit around and smoke cigars".

From the second and third sentences, "They would sit around and smoke cigars. Nathan took a trip to Cuba." Our story augmentation process adds new context using three sentences "Nathan starts to smoke another cigarette. Nathan wants to smoke more. Nathan wants to work. Nathan starts to have money." They use $start$ and $want$ relation to adopt causal commonsense reasoning. Our model reasons a new sentence from the previous sentence: "Nathan starts to smoke **another** cigarette. Our model generates the following story from everyday commonsense, meaning we need to make money for a trip overseas. From the third and fourth sentences, "Nathan took a trip to Cuba. He bought a lot of cigars there to bring home." Our model enriches more background information such as $airport\ car$, and $store$. These are possible from causal reasoning.

Similarly, also Case 2 demonstrates that our model is effective. We use $begin$, $start$, and try relation. From the internal context $diet$, we enrich the context using "health food" and "exercise". We add causal ordering. In particular, from "rage", our model generates "calm down", "rest", and "think". The human evaluation shows that we enrich the context by adding additional context from commonsense. Hence, we can make the story more enjoyable and close connection to the gold ending.

As shown in the results, our story augmentation process successfully adds more causality to the story following the reader's understanding. Also, our model using story augmentation and multi-level GCNs is beneficial to generate story endings.

Table 5. The effect of a different number of additional sentences ("as" denotes the number of additional sentences from each adjacent sentence).

Model	BLEU-1	METEOR	ROUGE-1	ROUGE-L
CCRGCN$_{as=0}$	24.6	18.8	28.4	27.8
CCRGCN$_{as=1}$	24.7	18.9	28.5	27.9
CCRGCN$_{as=2}$	26.3	19.3	30.7	28.5
CCRGCN$_{as=3}$	27.0	19.7	31.2	28.7
CCRGCN$_{as=4}$	**27.2**	**20.4**	**31.5**	**29.0**
CCRGCN$_{as=5}$	24.4	18.6	28.3	27.8

4.5.4 Number of Additional Sentences

Table 5 shows each case's automated evaluation metric results by varying the number of additional sentences. In this experiment, as means the number of additional sentences from story augmentation. For example, CCRGCN$_{as=0}$ means the results of multi-level GCNs without story augmentation. We find that the optimal number of additional sentences is 4. The model with 5 additional sentences fails to produce improved results. Since even one additional sentence of story augmentation increases all the evaluation measures, we conclude that story augmentation significantly (up to a certain number of additional sentences) contributes to maximizing the inference of story endings.

5 Conclusion

Story ending generation is challenging due to the difficulty of capturing internal and external context. Also, the model should generate a make-sense conclusion. We propose a story augmentation with multi-level GCNs for generating story endings. At first, we enrich stories given four sentences using soft causal relation to track readers' everyday understanding when they read a story. To capture all the given information, we apply dependency tree parsing. Then we apply multi-level GCNs to train our story generation model. Our approach effectively generates story endings from both automatic and human evaluations. We also conduct an ablation study to address the effect of story argumentation with varying the number of additional sentences.

Acknowledgements

This research was supported by the Australian Government Research Training Program Scholarship.

References

[1] Prithviraj Ammanabrolu, Wesley Cheung, William Broniec, and Mark O. Riedl, 'Automated storytelling via causal, commonsense plot ordering', in *AAAI 2021*.

[2] Prithviraj Ammanabrolu, Ethan Tien, Wesley Cheung, Zhaochen Luo, William Ma, Lara Martin, and Mark Riedl, 'Story realization: Expanding plot events into sentences', in *AAAI 2020*.

[3] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning, 'Leveraging linguistic structure for open domain information extraction', in *ACL 2015*.

[4] Satanjeev Banerjee and Alon Lavie, 'METEOR: An automatic metric for MT evaluation with improved correlation with human judgments', in *ACL 2005*.

[5] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi, 'COMET: commonsense transformers for automatic knowledge graph construction', in *ACL 2019*.

[6] Daniel Cer, Marie-Catherine de Marneffe, Daniel Jurafsky, and Christopher D. Manning, 'Parsing to stanford dependencies: Trade-offs between speed and accuracy', in *LREC 2010*.

[7] Kevin Clark and Christopher D. Manning, 'Deep reinforcement learning for mention-ranking coreference models', in *EMNLP 2016*.

[8] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa, 'Natural language processing (almost) from scratch', *J. Mach. Learn. Res.*, **12**, (2011).

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 'BERT: Pre-training of deep bidirectional transformers for language understanding', in *NAACL 2019*.

[10] Chris Donahue, Mina Lee, and Percy Liang, 'Enabling language models to fill in the blanks', in *ACL 2020*.

[11] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang, 'GLM: general language model pretraining with autoregressive blank infilling', in *ACL 2022*.

[12] William Fedus, Ian Goodfellow, and Andrew Dai, 'Maskgan: Better text generation via filling in the', (2018).

[13] Pablo Gervás, Belén Díaz-Agudo, Federico Peinado, and Raquel Hervás, 'Story plot generation based on cbr', volume 18, (2005).

[14] Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang, 'A knowledge-enhanced pretraining model for commonsense story generation', *Transactions of the Association for Computational Linguistics*, **8**, (2020).

[15] Jian Guan, Yansen Wang, and Minlie Huang, 'Story ending generation with incremental encoding and commonsense knowledge', in *AAAI 2019*.

[16] Zhijiang Guo, Yan Zhang, and Wei Lu, 'Attention guided graph convolutional networks for relation extraction', in *ACL 2019*.

[17] Qingbao Huang, Linzhang Mo, Pijian Li, Yi Cai, Qingguang Liu, Jielong Wei, Qing Li, and Ho-fung Leung, 'Story ending generation with multi-level graph convolutional networks over dependency trees', in *AAAI 2021*.

[18] Qingbao Huang, Jielong Wei, Yi Cai, Changmeng Zheng, Junying Chen, Ho-fung Leung, and Qing Li, 'Aligned dual channel graph convolutional network for visual question answering', in *ACL 2020*.

[19] Daphne Ippolito, David Grangier, Chris Callison-Burch, and Douglas Eck, 'Unsupervised hierarchical story infilling', in *The First Workshop on Narrative Understanding*, (2019).

[20] Yangfeng Ji, Chenhao Tan, Sebastian Martschat, Yejin Choi, and Noah A. Smith, 'Dynamic entity representations in neural language models', in *EMNLP 2017*.

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

[22] Zhongyang Li, Xiao Ding, and Ting Liu, 'Generating reasonable and diversified story ending using sequence to sequence model with adversarial training', in *ACL 2018*.

[23] Chin-Yew Lin, 'ROUGE: A package for automatic evaluation of summaries', in *ACL 2004*.

[24] Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan, 'A token-level reference-free hallucination detection benchmark for free-form text generation', in *ACL 2022*.

[25] Thang Luong, Hieu Pham, and Christopher D. Manning, 'Effective approaches to attention-based neural machine translation', in *EMNLP 2015*.

[26] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen, 'A corpus and cloze evaluation for deeper understanding of commonsense stories', in *NAACL HLT 2016*.

[27] Judith S. Olson and Wendy A. Kellogg, *Ways of Knowing in HCI*, Springer Publishing Company, Incorporated, 2014.

[28] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe, 'Training language models to follow instructions with human feedback'. arXiv, (2022).

[29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, 'Bleu: a method for automatic evaluation of machine translation', in *ACL 2002*.

[30] Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight, 'Towards controllable story generation', in *Proceedings of the First Workshop on Storytelling*, (2018).

[31] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, 'Glove: Global vectors for word representation', in *EMNLP 2014*.

[32] Julie Porteous and Marc Cavazza, 'Controlling narrative generation with planning trajectories: The role of constraints', (2009).

[33] Chris Purdy, Xinyu Wang, Larry He, and Mark O. Riedl, 'Predicting generated story quality with quantitative measures', in *AIIDE 2018*.

[34] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al., 'Language models are unsupervised multitask learners', *OpenAI blog*, **1**(8), 9, (2019).

[35] Mark Riedl and Robert Young, 'Narrative planning: Balancing plot and character', *J. Artif. Intell. Res. (JAIR)*, (01 2014).

[36] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi, 'Atomic: An atlas of machine commonsense for if-then reasoning', in *AAAI 2019*.

[37] Qing Sun, Stefan Lee, and Dhruv Batra, 'Bidirectional beam search: Forward-backward inference in neural sequence models for fill-in-the-blank image captioning', (2017).

[38] Kai Sheng Tai, Richard Socher, and Christopher D. Manning, 'Improved semantic representations from tree-structured long short-term memory networks', *CoRR*, (2015).

[39] Pradyumna Tambwekar, Murtaza Dhuliawala, Lara J. Martin, Animesh Mehta, Brent Harrison, and Mark O. Riedl, 'Controllable neural story plot generation via reward shaping', in *IJCAI 2019*.

[40] Wilson L. Taylor, '"cloze procedure": A new tool for measuring readability', *Journalism Quarterly*, **30**(4), 415–433, (1953).

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin, 'Attention is all you need', in *NIPS 2017*.

[42] Tianming Wang and Xiaojun Wan, 'T-cvae: Transformer-based conditioned variational autoencoder for story completion', in *IJCAI 2019*.

[43] Stephen G. Ware and R. Michael Young, 'Cpocl: A narrative planner supporting conflict', in *AIIDE 2011*.

[44] Yanling Xiao, Lemao Liu, Guoping Huang, Qu Cui, Shujian Huang, Shuming Shi, and Jiajun Chen, 'Bitiimt: A bilingual text-infilling method for interactive machine translation', in *ACL 2022*.

[45] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh, 'Graph r-cnn for scene graph generation', in *European Conference*, (2018).

[46] Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan, 'Plan-and-write: Towards better automatic storytelling', in *AAAI 2019*.

[47] Yan Zhao, Lu Liu, Chunhua Liu, Ruoyao Yang, and Dong Yu, 'From plots to endings: A reinforced pointer generator for story ending generation', in *Natural Language Processing and Chinese Computing*, eds., Min Zhang, Vincent Ng, Dongyan Zhao, Sujian Li, and Hongying Zan, pp. 51–63, Cham, (2018). Springer International Publishing.

[48] Yucheng Zhou, Tao Shen, Xiubo Geng, Guodong Long, and Daxin Jiang, 'ClarET: Pre-training a correlation-aware context-to-event transformer for event-centric generation and classification', in *ACL 2022*.