Story Ending Generation with Multi-Level Graph Convolutional Networks over Dependency Trees

Qingbao Huang,[1,2] Linzhang Mo,[2] Pijian Li,[2] Yi Cai,[1,3 *]
Qingguang Liu,[2] Jielong Wei,[2] Qing Li,[4] Ho-fung Leung[5]
[1]School of Software Engineering, South China University of Technology, Guangzhou, China
[2]School of Electrical Engineering, Guangxi University, Nanning, China
[3]Key Laboratory of Big Data and Intelligent Robot (SCUT), MOE of China
[4]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[5]Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China
qbhuang@gxu.edu.cn, 1912302009@st.gxu.edu.cn, ycai@scut.edu.cn

Abstract

As an interesting and challenging task, story ending generation aims at generating a reasonable and coherent ending for a given story context. The key challenge of the task is to comprehend the context sufficiently and capture the hidden logic information effectively, which has not been well explored by most existing generative models. To tackle this issue, we propose a context-aware Multi-level Graph Convolutional Networks over Dependency Parse (MGCN-DP) trees to capture dependency relations and context clues more effectively. We utilize dependency parse trees to facilitate capturing relations and events in the context implicitly, and Multi-level Graph Convolutional Networks to update and deliver the representation crossing levels to obtain richer contextual information. Both automatic and manual evaluations show that our MGCN-DP can achieve comparable performance with state-of-the-art models. Our source code is available at https://github.com/VISLANG-Lab/MLGCN-DP.

Introduction

Story ending generation (SEG) is an interesting and challenging task in machine comprehension and natural language generation, which aims at completing the plot and concluding a story ending given a context. A recently proposed stories corpus named *ROCStories* (Mostafazadeh et al. 2016a) provides a suitable dataset for SEG. The original task is to select a correct story ending from two candidates, while the variational task SEG is to generate a reasonable ending. The latter is more challenging, because models must "understand" the story context, and then generate reasonable, coherent, and diverse endings according to the logic relation and causality information.

Previous works (Li, Ding, and Liu 2018; Gupta et al. 2019) are mainly based on Sequence-to-Sequence (Seq2Seq) model (Luong, Pham, and Manning 2015). Because of generating a sentence at a stroke in a left-to-right manner and training with Maximum Likelihood Estimate, they suffer from a well known issue of generating non-coherent and generic plots. Very recently, Guan et al. propose a model that uses an incremental encoding approach and conducts one hop reasoning over the ConceptNet graph to augment the representation of words in the context (Guan, Wang, and Huang 2019). Owing to incorporating external commonsense knowledge, the content of generated endings appears more abundant. In terms of coherence and reasonability, however, there is still a big gap between machines and human. One reason is that the proposed incremental encoding approach can build relationship of words in adjacent sentences incrementally, however, it cannot directly capture information from non-adjacent sentences, especially long-range sentences. The other possible reason is that an over reliance on exteral commonsense knowledge beyond context could lead to deviating from the main theme.

We consider clues hidden in the whole context are vital to high-quality ending generation, therefore it is necessary to seek an approach to better grasp the long-range dependency relations. Actually, there are many useful entities (e.g., *Tom, phone, biggest screen*) and events (refer to a verb or action here, e.g., *bought, change*) in the sentences (cf. Figure 1), which are beneficial to reveal the logical relationship hidden in the story context. Besides, we can find that the key word *phone* has some relations with other sentences, while the phrase *biggest screen* has a causal relationship with the fourth sentence (*biggest screen* $\rightarrow$ *pain to carry*). Furthermore, some sentences have causal relationships with others, from which the context clue ($bought\_phone \rightarrow got\_biggest\_screen \rightarrow be\_great\_look \rightarrow be\_pain\_carry \rightarrow change\_smaller\_one$) can be inferred . To understand story context adequately, models should try to capture relations and events hidden in the input sentences.

Dependency trees have been proven to be effective in extracting relations (Zhang, Qi, and Manning 2018; Guo, Zhang, and Lu 2019) and events (McClosky, Surdeanu, and Manning 2011) in text for the ability of capturing long-range syntactic relations. In Figure 1, we obtain the dependency relations of each sentence by the Stanford Syntactic Parsing tool (De Marneffe et al. 2014), and select the words

*Corresponding author: Yi Cai (ycai@scut.edu.cn)

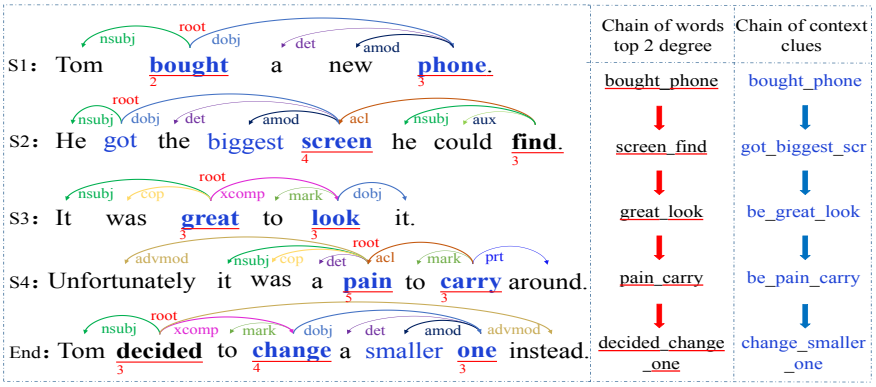

Figure 1: An example of SEG. The arcs with arrows represent the dependency relations between words. The abbreviations (e.g., *dobj*) are dependency relations. The numbers in red represent the degrees of the relations of corresponding words. We select the top-2-degree words in each dependency parse tree to form a top dependency relation chain. The chain in blue denotes the context clues analyzed by human. They look similar with each other.

of top-2-degree in each dependency tree to form a chain by chronological order ($bought\_phone \rightarrow screen\_find \rightarrow great\_look \rightarrow pain\_carry \rightarrow decided\_change\_one$). We observe that it is similar to the chain of the context clue summarized by human ($bought\_phone \rightarrow got\_biggest\_screen \rightarrow be\_great\_look \rightarrow be\_pain\_carry \rightarrow change\_smaller\_one$). Intuitively, we consider that dependency relations will facilitate the capture of context clues and eventually benefit for reasonable story plot generating.

For the strong ability of aggregate associated information from neighbor nodes, Graph Convolutional Networks (GCN) (Kipf and Welling 2017) and its variants, e.g., attention-based GCN (Yang et al. 2018) have been widely used for variety graph-based applications (Zhang, Qi, and Manning 2018; Huang et al. 2020; Guo, Zhang, and Lu 2019). Motivated by aforementioned observations, we put forward Multi-level Graph Convolutional Networks over dependency parse (MGCN-DP) trees to construct the dependency relations of input sentences. The key idea of our model is to encode the dependency structures over the input sentences with efficient graph convolution operations, then extract relations and events implicitly to obtain relation- and event-centric representations of the whole story context.

Our contributions can be summarized as follows:

- To grasp context information (including intra- and inter-sentence information) sufficiently, we propose Multi-level Graph Convolutional Networks to deliver representations by crossing levels. Our model can get an enhanced context representation which is conducive to the generation of more logical and reasonable story endings.
- To the best of our knowledge, this is the first endeavor to introduce dependency trees to the SEG task. By implicitly extracting relations and events, our model can capture logic relations and causality information hidden in the story context to some extent.
- Experiments show that our model can generate reasonable and coherent story endings, and achieve comparable performance on both automatic and manual evaluations with the state-of-the-art models. It also shows that injecting external symbolic representations could be helpful for SEG.

Related Work

Story Cloze Test (SCT) Mostafazadeh et al. define the SCT task to select a correct ending from two candidates for a given story (Mostafazadeh et al. 2016b). Previous methods can be roughly categorized into two lines: Feature-based approaches (Schwartz et al. 2017; Chaturvedi, Peng, and Roth 2017) measure the coherence between candidates and the given story context from aspects of topic and sentiment, while neural models (Mostafazadeh et al. 2016b; Cai, Tu, and Gimpel 2017; Peng, Chaturvedi, and Roth 2017; Chen, Chen, and Yu 2019; Cui et al. 2020) learn embeddings for the context and candidate endings, and select the right ending by computing the embeddings' similarity.

Story Generation Different from SCT, story generation (SG) is more challenging to generate a reasonable and logical self-consistent story plot. It can be classified into two groups: the restricted SG and the open-ended SG. The former is to generate a content-related story conditioning on various given contents, such as images(Huang et al. 2016), news (Liu et al. 2020a), and short descriptions (Jain et al. 2017). The latter attempts to generate an open-ended story with very limited leading information, such as a title (Yao et al. 2019; Li et al. 2019) and a sentence (Xu et al. 2018; Guan et al. 2020). A typical generation formulation is to firstly generate intermediate representations, e.g., key words (Yao et al. 2019), skeleton (Xu et al. 2018), events (Martin et al. 2018), prompts (Fan, Lewis, and Dauphin 2018), and characters (Liu et al. 2020b), then rewrite and enrich them to generate complete stories.

Story Ending Generation SEG (Zhao et al. 2018) is a specialization of SG. Specifically, a SEG model needs to deeply understand context information firstly, then generates a reasonable ending which accords with the logic thread of

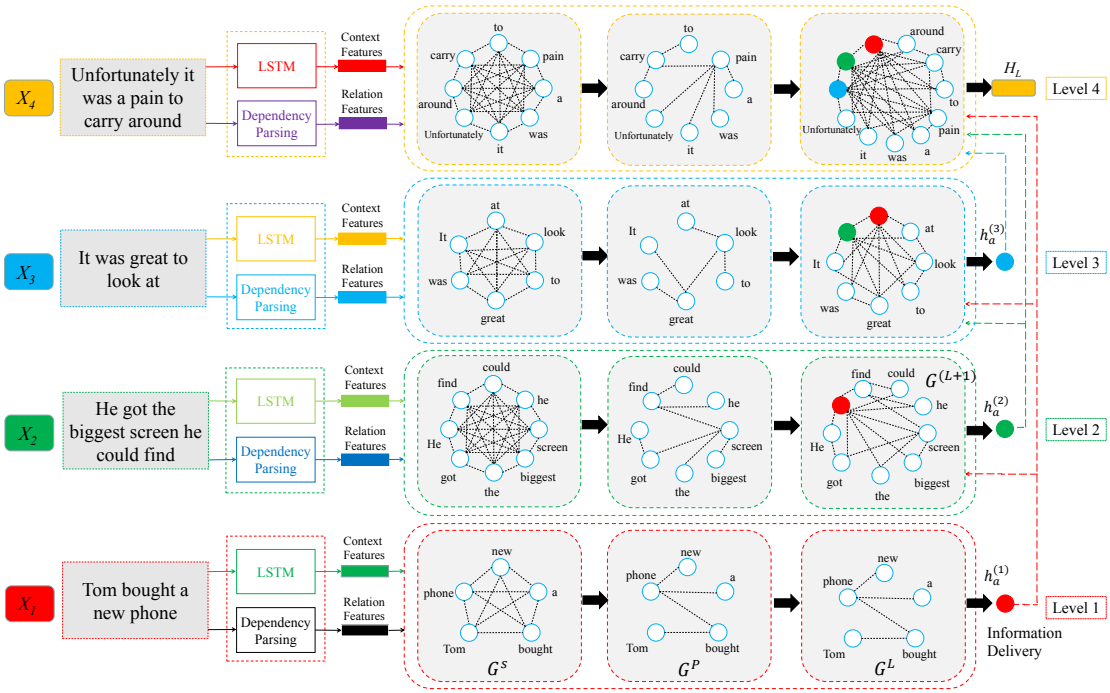

Figure 2: Illustration of the proposed multi-level graph convolutional networks over dependency parse trees for the SEG task. The model is equipped with four interrelated GCNs to capture intra- and inter-sentence information. The dependency parsing is used to construct the dependency relations between words. We use attention mechanism to weight each nodes and sum together as a new node (e.g., $h_a^{(1)}$, $h_a^{(2)}$, $h_a^{(3)}$) for the first three level GCNs. The model delivers representations of the preceding sentences to the last GCN. The final output H_L is obtained and served as the input of the decoder.

given context. Li et al. introduce a Seq2Seq model with adversarial training to generate reasonable and diversified endings (Li, Ding, and Liu 2018). Guan et al. propose a model using incremental encoding scheme and external structured commonsense knowledge to generate story endings (Guan, Wang, and Huang 2019). Wang et al. adopt a modified Transformer to capture the contextual clues and a conditioned variational autoencoder to improve the diversity and coherence (Wang and Wan 2019). Other works attempt to control sentiment (Peng et al. 2018; Luo et al. 2019) and attributes (Tu et al. 2019) to obtain diversified endings.

We endeavor to solve the issue of SEG from a perspective of neuro-symbolic. Specifically, we introduce dependency trees to implicitly extract relations and events, which has been proven effective in relation extraction (Zhang, Qi, and Manning 2018; Guo, Zhang, and Lu 2019) and event extraction (McClosky, Surdeanu, and Manning 2011; Björne and Salakoski 2018) tasks. Meanwhile, we propose Multi-level Graph Convolutional Networks to deliver relation- and event-centric representations crossing sentences.

Model

Overview

The SEG task can be formulated as follows: given a story context consisting of a sentence sequence $X = \{X_1, X_2, \cdots, X_\mu\}$, where $X_s = x_1^{(s)} x_2^{(s)} \cdots x_n^{(s)}$ contains n words in the s-th sentence, the goal of SEG is to generate a story ending Y related to the given context X.

Our MGCN-DP model is based on an encoder-decoder architecture (cf. Figure 2). We first use Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) to encode each input sentence and obtain representations of words to construct fully connected graphs. We apply the Standford Dependencies tool to parse dependency relations, then remove some unrelated edges and obtain the sparse graphs. We leverage an attention-based GCN (Yang et al. 2018) to update each node by aggregating information of its neighbor nodes, e.g., the node "phone" at *Leve1 1* in Figure 2. Aiming at handling the four input sentences to get enhanced inter-sentence representations, we devise a delivery mechanism to adjust GCNs of latter sentences self-adaptively. Specifically, we apply an attention mechanism to weight each node in the GCN of a sentence and sum them together as a new node $h_a^{(L)}$ for the GCN of the next level. In this way, the representation of the preceding sentences can be delivered to the last GCN. By this multi-level GCN encoding over dependency trees, we can get more information from the whole story context, which contains relation- and event-centric representations, thus help to generate a more reasonable and coherent ending.

Representation of Intra-Sentence Information

Word Representation Given a s-th input sentence $X_s = x_1^{(s)} x_2^{(s)} \cdots x_n^{(s)}$ with n words, we represent the k-th word $x_k^{(s)}$ in s-th sentence by Glove embedding (Pennington, Socher, and Manning 2014), following LSTM to obtain the word representation $h_{wk}^{(s)}$:

$$e_k^{(s)} = \mathbf{e}^w(x_k^{(s)}), \tag{1}$$

$$h_{wk}^{(s)} = LSTM(e_k^{(s)}), \tag{2}$$

where $\mathbf{e}^w$ denotes a word embedding lookup table and $e_k^{(s)}$ is the embedding vector of k-th word $x_k^{(s)}$ in the s-th sentence.

Fully-connected Graph Construction For s-th sentence $X_s = x_1^{(s)} x_2^{(s)} \cdots x_n^{(s)}$, each word $x_k^{(s)}$ is a node and its node feature is the corresponding word representation $h_{wk}^{(s)}$. Each edge represents a certain relation between two words. As shown in Figure 3(a), by treating each word in a sentence as a vertex, an intra-sentence graph G^s is constructed as follows:

$$G^s = (\mathcal{V}^s, \xi^s), \tag{3}$$

where $\mathcal{V}^s$ is the set of nodes and ξ^s is the set of edges connected with these nodes.

Pruned Graph with Dependency Tree

Pruned Graph with Dependency Relations Irrelevant relations between two words may bring noises. Therefore, we need to remove some unrelated relations to reduce the noises. By parsing the sentence, we obtain the dependency relations between words (cf. Figure 1). According to dependency relations, we remove some unrelated edges and obtain a sparse graph G^P, cf. Figure 3(b), which can be denoted as:

$$G^P = (\mathcal{V}^P, \xi^P), \tag{4}$$

where $\mathcal{V}^P$ is the set of nodes of the pruned graph and ξ^P is the set of edges connected with these nodes of the pruned graph. Then we perform the GCNs' node aggregation and updating on the sparse graph.

Aggregation and Updating of Nodes Following previous studies (Yang et al. 2018; Huang et al. 2020), we use an attention-based GCN to update the representations of inputs.

Given a graph with n nodes, each word in the sentence X_s is a node. We represent the graph structure with a $n \times n$ adjacency matrix, where the relations between nodes are reflected by a fully connected layer. For a target node i and its neighbor nodes set $\mathcal{N}(i)$, the representations of node i and node $j \in \mathcal{N}(i)$ are h_{wi} and h_{wj}, respectively. To obtain the correlation score w_{ij} between node i and node j, we learn a fully connected layer over concatenated node features h_{wi} and h_{wj}:

$$w_{ij} = w_0^T \sigma\left(W_0[h_{wi}; h_{wj}] + b_0\right), \tag{5}$$

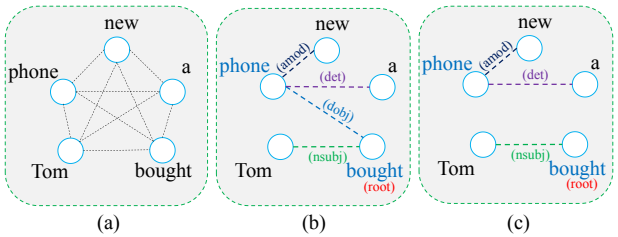

Figure 3: (a) A fully-connected graph. (b) Remove the unrelated edges to get a sparse graph. (c) Ablation study: The dependency relation $dobj$ between $bought$ and $phone$ is removed. (Will be analyzed in the ablation study section.)

where w_0, W_0, and b_0 are trainable parameters, σ is the non-linear activation function, and $[h_{wi}; h_{wj}]$ denotes the concatenation operation. We apply a softmax function over the correlation score w_{ij} to obtain the weight α_{ij}:

$$\alpha_{ij} = \frac{\exp(w_{ij})}{\sum_{j \in \mathcal{N}(i)} \exp(w_{ij})}. \tag{6}$$

The l-th representations of neighboring nodes $h_{wj}^{(l)}$ are first transformed via a learned linear transformation layer W_1. Those transformed representations are gathered with the weight α_{ij} followed by a non-linear function σ (e.g., ReLU). This propagation is denoted as:

$$h_{wi}^{(l+1)} = \sigma(h_{wi}^{(l)} + \sum_{j \in \mathcal{N}i} \alpha_{ij}(W_1 h_{wj}^{(l)} + b_1)), \tag{7}$$

where W_1 and b_1 are trainable parameters. Following the stacked l layer GCN, the output H_w of GCN is denoted as:

$$H_w = h_{wi}^{(l+1)}. \tag{8}$$

Multi-level GCN and Information Delivery

Multi-level Representation We adopt a multi-level GCN on sparse graphs to represent four sentences context. The sparse graph G^L can be denoted as:

$$G^L = (\mathcal{V}^L, \xi^L), \tag{9}$$

where G^L is the L-th level GCN graph, $\mathcal{V}^L$ is the set of the L-th level GCN nodes and ξ^L is the set of the L-th level GCN edges.

Information Delivery For the s-th sentence X_s with n words, the representations of all words in s-th sentence are $[h_{w1}^{(s)} \cdots h_{wn}^{(s)}]$. The node set $\mathcal{V}^L$ in L-th level GCN is:

$$\mathcal{V}^L = [h_{w1}^{(s)} \cdots h_{wn}^{(s)}]. \tag{10}$$

To set up information delivery between different level GCN, we use attention mechanism to weight each node in $\mathcal{V}^L$ and sum them together as a new node $h_a^{(L)}$:

$$\beta = softmax(W_2 \mathcal{V}^L + b_2), \tag{11}$$

$$h_a^{(L)} = \sum_{L=1}^{n} \beta \mathcal{V}^L, \tag{12}$$

where W_2 and b_2 are trainable parameters.

For the $(s+1)$-th sentence X_{s+1} consisting of m words, the representations of all words in $(s+1)$-th sentence are $[h_{w1}^{(s+1)} \cdots h_{wm}^{(s+1)}]$. Then we combine $[h_{w1}^{(s+1)} \cdots h_{wm}^{(s+1)}]$ with $h_a^{(L)}$ as the nodes set $\mathcal{V}^{(L+1)}$ of the $(L+1)$-th level GCN:

$$\mathcal{V}^{(L+1)} = [h_{w1}^{(s+1)} \cdots h_{wm}^{(s+1)}; h_a^{(1)}, \cdots, h_a^{(L)}]. \quad (13)$$

Given a graph with $(m+L)$ nodes, the $(m+L) \times (m+L)$ adjacency matrix is used to represent the graph structure $G^{(L+1)}$. For a target node *i* and a neighboring node $j \in \Psi(i)$ in the $(L+1)$-th level graph $G^{(L+1)}$, $\Psi(i)$ is the set of nodes neighboring with node *i*. The representations of node *i* and node *j* are h_{Li} and h_{Lj}, respectively. To obtain the correlation score λ_{ij} between node *i* and node *j*, we use a connected layer to learn the correlation between node features h_{Li} and h_{Lj}:

$$\lambda_{ij} = w_3^T \sigma(W_3[h_{Li}; h_{Lj}] + b_3), \quad (14)$$

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. We apply a softmax function over the correlation score λ_{ij} to obtain the weight ϕ_{ij}:

$$\phi_{ij} = \frac{\exp(\lambda_{ij})}{\sum_{j \in \Psi(i)} \exp(\lambda_{ij})}. \quad (15)$$

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

$$h_{Li}^{(l+1)} = \sigma(h_{Li}^{(l)} + \sum_{j \in \psi(i)} \phi_{ij}(W_4 h_{Lj}^{(l)} + b_4)), \quad (16)$$

where W_4 and b_4 are trainable parameters.

Following the stacked l layer GCN, the output of the encoder H_L is denoted as:

$$H_L = h_{Li}^{(l+1)}. \quad (17)$$

Decoder

We utilize the decoder of Transformer (Vaswani et al. 2017) to decoding. The inputs of Multi-Head Attention are D_{in}, H_L, and H_L. This process is denoted as:

$$\tilde{D}_{in} = MultiHead(D_{in}, H_L, H_L), \quad (18)$$

$$D_o = FFN(\tilde{D}_{in}), \quad (19)$$

where D_{in} is input of decoder, FFN is two linear transformations with a ReLU activation in between, and D_o is the middle output of decoder.

To predict word probabilities and generate words, we use a linear transformation layer and softmax function to convert the output of decoder. At each time step *t*, the decoding process is represented as :

$$P(y_t | y < t, X) = softmax(W_5 D_o + b_5), \quad (20)$$

where W_5 and b_5 are trainable parameters, $P(y_t)$ is the probability distribution over vocabulary.

Experiments

Dataset

We evaluate our model on the ROCStories corpus (Mostafazadeh et al. 2016a). The dataset contains 98,162 stories. We follow the work (Guan, Wang, and Huang 2019) to divide the dataset into the training, validation, and test set with 90,000 stories, 4,081 stories and 4,081 stories, respectively. All methods are evaluated on the test set, and the validation set could only be used for training purpose.

Baselines

We compare our model with the following models:

- **Seq2Seq:** A simple encoder-decoder model which concatenates four sentences to a long sentence with an attention mechanism (Luong, Pham, and Manning 2015).
- **Transformer:** The vanilla Transformer (Vaswani et al. 2017) is compared, which is an encoder-decoder model with multi-head attention and feed forward networks.
- **IE+MSA:** A model which uses the incremental encoding scheme and incorporate external knowledge for generating endings (Guan, Wang, and Huang 2019).
- **T-CVAE:** A Transformer-based conditional variational autoencoder model (Wang and Wan 2019) for the story completion task. We compare it with our model on SEG.
- **GCN:** To test our MGCN-DP, we also apply an attention-based GCN (Yang et al. 2018) to SEG for a baseline. We concatenate four sentences to a long sequence as the input of GCN. By treating each word in the sequence as a vertex, a fully-connected undirected graph is constructed.
- **KE-GPT2:** A knowledge-enhanced pre-training model for SG based on GPT-2 (Guan et al. 2020). It is fine-tuned with multi-task learning by distinguishing true or fake stories to capture causal and temporal dependencies between sentences. To adapt it to the SEG task and generate story endings, we give the first four sentences as inputs.
- **Plan&Write:** Yao et al. explore an open-domain SG task given a title (topic) as input (Yao et al. 2019). We use the static schema, which first plans a sequence of keywords, then generates a story based on them. We adapt it to generate a story ending by leveraging the first four sentences and the corresponding keywords which extracted by the RAKE algorithm (Rose et al. 2010).

Experimental Settings

We conduct experiments of baselines on their released codes without changing the best parameter settings. For the Transformer model, the head *h* of attention is 8, the level of the Transformer blocks is 6. The dropout rate is 0.1, the batch size is 64, and the learning rate is 0.005. The dimension of word embedding is 300. For our model, the level of stacked layer in GCN is 5, the learning rate is 0.005, the batch size is 64, the head *h* of attention in decoder is 6, d_k and d_v are 64, the level of stacked layer of decoder is 2, the dropout rate is 0.1. The GloVe.6B (Pennington, Socher, and Manning 2014) is used as word embedding and the dimension is 300. We train the model for 20 epochs.

Model	B1%	B2%	Gram	Logic
Seq2Seq	18.5	5.9	2.57	1.41
Transformer	17.4	6.0	2.54	1.62
GCN	17.6	6.2	2.62	1.70
IE+MSA	24.4	7.8	2.64	1.80
T-CVAE	24.3	7.7	2.58	1.71
Plan&Write	24.4	8.4	2.65	1.73
KE-GPT2*	**26.5**	**9.4**	2.65	**1.92**
MGCN-DP(ours)	24.6	8.6	**2.67**	1.86

Table 1: Automatic evaluation and human evaluation. In each column, we bold / underline the best and the second performance, respectively. The model with * is pretrained on a large-scale corpus, along with greater demands on resources and time.

Evaluation Metrics

Automatic Evaluation Metric We use BLEU (Papineni et al. 2002) for our automatic evaluation. BLEU evaluates n-gram overlap between a generated ending and a reference. Following the work (Guan, Wang, and Huang 2019), we report BLEUs with $n = 1, 2$ (i.e., B1, B2).

Human Evaluation Metric Grammaticality (Gram) and logicality (Logic) are used for manual evaluation. Gram is used to evaluate whether the generated story is fluent and natural, while Logic to evaluate whether the generated story is reasonable and coherent with the context. As a factor closely related to the story plot, compared with Gram, Logic is more important to evaluate the quality of the generated ending sentence. For fair comparison, score 1/2/3 is applied during annotation following (Wang and Wan 2019). 1 means bad, 2 means okay and 3 means good. We randomly pick 100 story endings generated by the baselines and our model on the test set, respectively, then distribute them to five well-educated annotators and obtain the averaged scores.

Results and Analysis

Automatic Evaluation The results of the automatic evaluation are shown in Table 1. It can be seen that our MGCN-DP model outperforms other non-pretrained baselines on B1 and B2. More specifically, our model achieves an improvement of 6.1% / 7.2% / 7% / 0.2% / 0.3% / 0.2% over the Seq2Seq / Transformer / GCN / IE+MSA / T-CVAE / Plan&Write model, respectively. As for B2, our model outperforms the Seq2Seq / Transformer / GCN / IE+MSA / T-CVAE / Plan&Write model by 2.7% / 2.6% / 2.4% / 0.8% / 0.9% / 0.2%, respectively, and ranks second only to KE-GPT2 (0.8% below). Owing to large-scale corpora and large-scale knowledge bases used in the pretrained process, the KE-GPT2 model indeed lead other models on B1 and B2 metrics by a big margin. Even so our model achieves good scores by sufficiently capturing story contextual information without external resources. In general, the results show that the story ending generated by our model has comparatively high overlaps with the reference ending.

Model	B1%	B2%	Gram	Logic
MGCN-DP	**24.6**	**8.6**	**2.67**	**1.86**
w/o DP	22.0	7.8	2.65	1.79
w/o ML	21.7	7.6	2.64	1.77
w/o DP, ML	17.6	6.2	2.62	1.70

Table 2: Ablation studies. DP denotes the dependency parsing module and ML denotes the multi-level information delivery module. w/o means removing corresponding module from the whole MGCN-DP model.

Relation	Description	B1%	B2%
MGCN-DP	full model	24.6	8.6
w/o $nsubj$	nominal subject	23.7	7.5
w/o $dobj$	direct object	23.7	7.5
w/o acl	claussal modifier	23.7	7.5
w/o $iobj$	indirect object	23.8	7.6
w/o dep	dependent	23.8	7.6
w/o $amod$	adjective modifier	23.8	7.7
w/o $case$	pre/post-positions	23.9	7.8
w/o $advmod$	adverbial modifier	23.9	7.9
w/o det	determiner	24.0	8.0
w/o cop	copula	24.0	8.1
w/o $mark$	marker	24.1	8.1
w/o $xcomp$	open clausal	24.2	8.2

Table 3: Ablation studies on dependency relations which appear frequently in the dataset.

Manual Evaluation The results of the human evaluation are also shown in Table 1. We discover that all the models have fairly good grammar scores. It shows that models can learn grammar well. Our model exceeds Seq2Seq, Transformer, GCN, IE+MSA, T-CVAE, Plan&Write, KE-GPT2 on grammar by 5% / 6.5% / 2.5% / 1.5 %/ 4.5% / 1% / 1%, respectively (Significance Test, all p-values < 0.001). Unlike grammar scores, logicality scores differ with each other remarkably, and the average score of them is far from full mark of 3. This phenomenon also illustrates the challenge of generating consistent story endings. Nevertheless our model has higher scores than other non-pretrained baselines. Specifically, our MGCN-DP model is significantly better than Seq2Seq, Transformer, GCN, IE+MSA, T-CVAE, and Plan&Write on logicality by 22.5% / 12% / 8% / 3% / 7.5% / 6.5%, respectively (all p-values < 0.001). On the Logic metric, our model is closest to the KE-GPT2 which benefits from the large-scale corpus. It indicates that dependency parsing and MGCN-DP on context are really beneficial to grasp the logic relations and causal clues. We will show some cases and detailed analysis in case study section.

Ablation study To investigate the effects of the multi-level information delivery module (ML) and the dependency parsing module (DP) in the MGCN-DP model, we perform ablation studies. The results are shown in Table 2. When removing the DP module, the performance of our model drops by 2.6% on B1, 0.8% on B2, 1% on Gram, and 3.5% on

Case 1	
Context	Denise was at a friend's **wedding**. Her daughter had **come along** with her. Denise's daughter **wasn't feeling well**. Her daughter began to **feel worse** after she **ate** at the **reception**.
Seq2Seq	Denise was *happy* with her ***purchase***.
Transformer	Denise had to **go to the hospital**.
IE+MSA	Denise was *glad* she had made her daughter to the **wedding**.
T-CVAE	Denise decided to never **go to the wedding**.
GCN	Denise had to **go to the hospital**.
Plan&Write	Denise's daughter was very sad.
KE-GPT2	Denise was **worried** for her daughter.
MGCN-DP	Denise had to **go to the hospital** with her **daughter**.
Gold Ending	Denise **took** her daughter **home** before she could **get sick** in **party**.
Case 2	
Context	Tom **bought** a bag of **chips**. When he **opened** them they were already **stale**. Tom **called** the **number** on the bag. He **complained** and **told** them what happened.
Seq2Seq	Tom had to **clean** it up.
Transformer	Tom had to **throw** them out.
IE+MSA	Tom **got** a new one.
T-CVAE	He had to **go to** the store to **get** a new **batch**.
GCN	Tom had to **throw** them away.
Plan&Write	Tom was furious.
KE-GPT2	They told him to **buy** another bag.
MGCN-DP	They **gave** him a **refund**.
Gold Ending	They **mailed** him **coupons** to **make up for** it.

Table 4: Generated endings from different models. Bold words denote the key entities, events, or key words in the story. Improper words in ending are italic.

Logic. When removing the ML module, the performance of our model drops by 2.9% on B1, 1% on B2, 1.5% on Gram, and 4.5% on Logic. When removing the DP and ML module together, the performance of our model drops by 7% on B1, 2.4% on B2, 2.5% on Gram, and 8% on Logic. All of these show that the DP module and ML module can help to generate more reasonable and coherent endings.

To further explore the contribution of every dependency relation, we conduct the ablation studies for the dependency relations on automatic evaluation. We remove the edges of a certain dependency relation (e.g., $dobj$ between $bought$ and $phone$ in Figure 3(b)) from the pruned sparse graph to get the representation without the $dobj$ (direct object) relation (cf. Figure 3(c)). As shown in Table 3, we just list a part of commonly-used dependency relations parsed by the Stanford dependency parsing tool and their corresponding results on B1 and B2. The results show that, among all the relations, $dobj$, $nsubj$, acl, $iobj$ and dep make the greatest contribution to the performance. More specifically, when removing the $nsubj$ / $dobj$ / acl / $iobj$ / dep relation, the performance of our model drops by 0.9% / 0.9% / 0.9% / 0.8% / 0.8% on B1 and 1.1% / 1.1% / 1.0% / 1.0% / 1.0% on B2, respectively. It is evident that the dependency parsing exerts a measure of influence over the B1 and B2 scores. In Figure 1, We can find that the dependency relations of the words consisting of the top-2-degree chain are $dobj$ of $bought\_phone$, acl of $screen\_find$, $xcomp$ of $great\_look$, acl of $pain\_carry$, $dobj$ of $decided\_change\_one$, respectively, and most of them have high scores in Table3. The results show, with extracting relations and events implicitly, dependency trees can facilitate the capture of context clues and benefit eventually for generating reasonable story plots.

Case Study We present some examples of generated story endings in Table 4.

In Case 1, we discover that all eight endings generated with good grammar. The Seq2Seq model misses the story context and gets a completely illogical result. The Transformer and GCN get the same generic output. It sounds reasonable, but does not provide any information about *her daughter*. The IE+MSA outputs the ending with the content about *Denise* and *her daughter*, but it misses the context clue *feel worse* and leads to unreasonable sentiment. A possible explanation may be that an over reliance on commonsense knowledge (e.g. Wedding is happy) beyond the context could lead to unreasonable generations. Compared with the phrase *wasn't feeling well* and *feel worse*, the generated ending by Plan&Write makes the plot barely developing. The main entities and events in input sentences are *Denise, friend's wedding, daughter* and *ate, feel worse*, respectively. Our MGCN-DP understands the context clues effectively and generates a more reasonable ending, which is even better than the gold ending.

In Case 2, to generate a reasonable and coherent ending, SEG models should understand the clue $bought\_chips \rightarrow already\_stale \rightarrow call\_number \rightarrow complained$. To some extent, our model captures this clue and generates a reasonable ending: *They gave him a refund*. It is most semantically similar to the gold ending. However, some baselines have not generated a contextual ending, they generate the unrelated or less relevant ending (e.g., Seq2Seq), generic ending (e.g., IE+MSA), or safe ending (e.g., Plan&Write). The generated endings of Transformer, GCN, and T-CVAE are acceptable, but compared with ones generated by KE-GPT2 and our MGCN-DP, they seem insipid for without the subject transformation and role-interaction. Furthermore, our generation more conforms to the actual situation in comparison with the one generated by KE-GPT2.

Conclusion

To improve the coherence and rationality of generated endings of SEG task, we devise a multi-layer GCN model over dependency trees to enhance the ability of capturing logic relations and causal clues hidden in the whole story context. We parse dependency relations of the input sentences to aid the GCNs for grasping implicitly context information and relations of intra- and inter-sentence. Experiments show that our model achieves the comparable performance with the state-of-the-art models. We shall explore the explicit reasoning and interpretability on SEG in the future.

Acknowledgments

We thank the anonymous reviewers for valuable comments and thoughtful suggestions.

This work was supported by National Natural Science Foundation of China (62076100, 51767005), the National Key Research and Development Program of China, and the collaborative research grants from the Fundamental Research Funds for the Central Universities, SCUT (No. D2182480), the Science and Technology Planning Project of Guangdong Province (No.2017B050506004), the Science and Technology Programs of Guangzhou (No.201704030076, 201707010223, 201802010027, 201902010046), and the Hong Kong Research Grants Council, China (project no. PolyU1121417 and project no. C1031-18G), and an internal research grant from the Hong Kong Polytechnic University, China (project 1.9B0V).

References

Björne, J.; and Salakoski, T. 2018. Biomedical Event Extraction Using Convolutional Neural Networks and Dependency Parsing. In Demner-Fushman, D.; Cohen, K. B.; Ananiadou, S.; and Tsujii, J., eds., *Proceedings of the BioNLP 2018 workshop, Melbourne, Australia, July 19, 2018*, 98–108. Association for Computational Linguistics.

Cai, Z.; Tu, L.; and Gimpel, K. 2017. Pay Attention to the Ending:Strong Neural Baselines for the ROC Story Cloze Task. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 616–622.

Chaturvedi, S.; Peng, H.; and Roth, D. 2017. Story comprehension for predicting what happens next. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1603–1614.

Chen, J.; Chen, J.; and Yu, Z. 2019. Incorporating structured commonsense knowledge in story completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6244–6251.

Cui, Y.; Che, W.; Zhang, W.; Liu, T.; Wang, S.; and Hu, G. 2020. Discriminative Sentence Modeling for Story Ending Prediction. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, 7602–7609.

De Marneffe, M.-C.; Dozat, T.; Silveira, N.; Haverinen, K.; Ginter, F.; Nivre, J.; and Manning, C. D. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *LREC*, volume 14, 4585–4592.

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.

Guan, J.; Huang, F.; Huang, M.; Zhao, Z.; and Zhu, X. 2020. A Knowledge-Enhanced Pretraining Model for Commonsense Story Generation. *Trans. Assoc. Comput. Linguistics* 8: 93–108.

Guan, J.; Wang, Y.; and Huang, M. 2019. Story ending generation with incremental encoding and commonsense knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6473–6480.

Guo, Z.; Zhang, Y.; and Lu, W. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 241–251. Florence, Italy.

Gupta, P.; Kumar, V. B.; Bhutani, M.; and Black, A. W. 2019. WriterForcing: Generating more interesting story endings. *arXiv preprint arXiv:1907.08259* .

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

Huang, Q.; Wei, J.; Cai, Y.; Zheng, C.; Chen, J.; Leung, H.; and Li, Q. 2020. Aligned Dual Channel Graph Convolutional Network for Visual Question Answering. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 7166–7176. Association for Computational Linguistics.

Huang, T. K.; Ferraro, F.; Mostafazadeh, N.; Misra, I.; Agrawal, A.; Devlin, J.; Girshick, R. B.; He, X.; Kohli, P.; Batra, D.; Zitnick, C. L.; Parikh, D.; Vanderwende, L.; Galley, M.; and Mitchell, M. 2016. Visual Storytelling. In Knight, K.; Nenkova, A.; and Rambow, O., eds., *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, 1233–1239. The Association for Computational Linguistics.

Jain, P.; Agrawal, P.; Mishra, A.; Sukhwani, M.; Laha, A.; and Sankaranarayanan, K. 2017. Story Generation from Sequence of Independent Short Descriptions. *CoRR* abs/1707.05501. URL http://arxiv.org/abs/1707.05501.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representation (ICLR)* .

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 *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, USA, January 27 - February 1, 2019*, 1715–1722. AAAI Press.

Li, Z.; Ding, X.; and Liu, T. 2018. Generating reasonable and diversified story ending using sequence to sequence model with adversarial training. In *Proceedings of the 27th International Conference on Computational Linguistics*, 1033–1043.

Liu, B.; Han, F. X.; Niu, D.; Kong, L.; Lai, K.; and Xu, Y. 2020a. Story Forest: Extracting Events and Telling Stories from Breaking News. *ACM Trans. Knowl. Discov. Data* 14(3): 31:1–31:28.

Liu, D.; Li, J.; Yu, M.; Huang, Z.; Liu, G.; Zhao, D.; and Yan, R. 2020b. A Character-Centric Neural Model for Automated Story Generation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, 1725–1732. AAAI Press.

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

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. *In Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP)* 1412–1421.

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 McIlraith, S. A.; and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 868–875. AAAI Press.

McClosky, D.; Surdeanu, M.; and Manning, C. D. 2011. Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 1626–1635.

Mostafazadeh, N.; Chambers, N.; He, X.; Parikh, D.; Batra, D.; Vanderwende, L.; Kohli, P.; and Allen, J. 2016a. A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 839–849. San Diego, California.

Mostafazadeh, N.; Vanderwende, L.; Yih, W.-t.; Kohli, P.; and Allen, J. 2016b. Story cloze evaluator: Vector space representation evaluation by predicting what happens next. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, 24–29.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 311–318.

Peng, H.; Chaturvedi, S.; and Roth, D. 2017. A Joint Model for Semantic Sequences: Frames, Entities, Sentiments. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, 173–183.

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

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

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

Schwartz, R.; Sap, M.; Konstas, I.; Zilles, L.; Choi, Y.; and Smith, N. A. 2017. Story cloze task: Uw nlp system. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, 52–55.

Tu, L.; Ding, X.; Yu, D.; and Gimpel, K. 2019. Generating Diverse Story Continuations with Controllable Semantics. *Proceedings of the 3rd Workshop on Neural Generation and Translation (WNGT 2019), Hong Kong, China, November 4* 44–58.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, T.; and Wan, X. 2019. T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 5233–5239.

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 (EMNLP)* 4306–4315.

Yang, J.; Lu, J.; Lee, S.; Batra, D.; and Parikh, D. 2018. Graph R-CNN for Scene Graph Generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Yao, L.; Peng, N.; Weischedel, R. M.; Knight, K.; Zhao, D.; and Yan, R. 2019. Plan-and-Write: Towards Better Automatic Storytelling. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, USA, January 27 - February 1, 2019*, 7378–7385.

Zhang, Y.; Qi, P.; and Manning, C. D. 2018. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2205–2215.

Zhao, Y.; Liu, L.; Liu, C.; Yang, R.; and Yu, D. 2018. From Plots to Endings: A Reinforced Pointer Generator for Story Ending Generation. In *CCF International Conference on Natural Language Processing and Chinese Computing*, 51–63.