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 Leung5

1School of Software Engineering, South China University of Technology, Guangzhou, China
2School of Electrical Engineering, Guangxi University, Nanning, China
3Key Laboratory of Big Data and Intelligent Robot (SCUT), MOE of China
4Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
5Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China
qib@stu.gxu.edu.cn, 1912302009@stu.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 the most existing generative models. To tackle this issue, we propose a context-aware Multi-level Graph Convolutional Networks over Dependency Parse (MGCN-DP) 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 external 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 → pain to carry). Furthermore, some sentences have causal relationships with others, from which the context clue (bought_phone → got_biggest_screen → be_great_look → be_pain_carry → 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...
of top-2-degree in each dependency tree to form a chain by chronological order (bought_phone → screen_find → great_look → pain_carry → decided_change_one). We observe that it is similar to the chain of the context clue summarized by human (bought_phone → got_biggest_screen → be_great_look → be_pain_carry → 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 node 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 $X = \{X_1, X_2, \cdots, X_n\}$, where $X_n = x_n^{(s)} \cdots x_n^{(s)}$ 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 Level 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 $h_a(L+1)$ 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 $X_s = x_1^{(s)} \cdots x_n^{(s)}$ 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 $h_{wk}^{(s)}$:

$$ e_k^{(s)} = \mathbf{e}^{w_k}(x_k^{(s)}), \quad (1) $$

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

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 $X_s = x_1^{(s)} \cdots x_n^{(s)}$, each word $x_j^{(s)}$ 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:

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

where $\mathcal{V}^s$ is the set of nodes and $\xi^s$ 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:

$$ G^p = (\mathcal{V}^p, \xi^p), \quad (4) $$

where $\mathcal{V}^p$ is the set of nodes of the pruned graph and $\xi^p$ 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 $h_{wj}$:

$$ w_{ij} = w_0^b \sigma (W_0[h_{wi}; h_{wj}] + b_0), \quad (5) $$

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:

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

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

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

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 $h_{n}^{(L)}$:

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

$$ h_{n}^{(L)} = \sum_{L=1}^{n} \beta \mathcal{V}^L, \quad (12) $$
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 $[h_w^{s+1}, \ldots, h_w^{s+1}]$. Then we combine $[h_w^{s+1}, \ldots, h_w^{s+1}]$ with $h_a^{(L)}$ as the nodes set $\gamma^{(L+1)}$ of the $(L+1)$-th level GCN:

$$\gamma^{(L+1)} = [h_w^{(s+1)}, \ldots, h_w^{(s+1)}; h_a^{(1)}, \ldots, h_a^{(L)}].$$

(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 $\lambda_{ij}$ between node $i$ and node $j$, we use a connected layer to learn the correlation between node features $h_{Li}$ and $h_{Lj}$:

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

(14)

where $w_3$, $W_3$, and $b_3$ are trainable parameters, $\sigma$ is the non-linear activation function, and $[h_{Li}; h_{Lj}]$ denotes the concatenation operation. We apply a softmax function over the correlation score $\lambda_{ij}$ to obtain the weight $\phi_{ij}$:

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

(15)

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

$$h_{Li}^{(i+1)} = \sigma(h_{Li}^{(i)} + \sum_{j \in \Psi(i)} \phi_{ij}(W_4h_{Lj}^{(i)} + b_4)),$$

(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:

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

(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:

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

(18)

$$D_0 = FFN(\hat{D}_{in}),$$

(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:

$$P(y_t | y < t, X) = \text{softmax}(W_5D_0 + b_5),$$

(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.
Following the work (Guan, Wang, and Huang 2019), we re-
n-gram overlap between a generated ending and a reference.
ert al. 2002) for our automatic evaluation. BLEU evaluates
Automatic Evaluation Metric We use BLEU (Papineni
Evaluation Metrics
Automatic Evaluation Metric We use BLEU (Papineni
t al. 2002) for our automatic evaluation. BLEU evaluates
Grammar and logicality (Logic) are used for manual evaluation. Gram is
HB EVALUATION METRIC
Human Evaluation Metric Grammaticality (Gram) and
Human Evaluation Metric Grammaticality (Gram) and
Natural, while Logic to evaluate whether the generated story is
Results and Analysis
Results and Analysis
Automatic Evaluation The results of the automatic evalu-
Automatic Evaluation The results of the automatic evalu-

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 re-
ources and time.

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 re-
ources and time.

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 re-
ources and time.

Evaluation Metrics
Evaluation Metrics
Human Evaluation Metric Grammaticality (Gram) and
Human Evaluation Metric Grammaticality (Gram) and

Results and Analysis
Results and Analysis
Automatic Evaluation The results of the automatic eval-
Automatic Evaluation The results of the automatic evalu-

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

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

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

Table 3: Ablation studies on dependency relations which ap-
pear frequently in the dataset.

Table 3: Ablation studies on dependency relations which ap-
pear 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, Trans-
fomer, 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). Un-
like 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% / 17.5% / 6.5%, respectively (all p-values < 0.001). On the Logic

d metric, our model is closest to the KE-GPT2 which bene-
fits from the large-scale corpus. It indicates that dependency
parsing and MGCN-DP on context are really beneficial to
grap 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 re-
moving 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

Table 3: Ablation studies on dependency relations which ap-
pear frequently in the dataset.

Table 3: Ablation studies on dependency relations which ap-
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, Trans-
fomer, 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). Un-
like 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% / 17.5% / 6.5%, respectively (all p-values < 0.001). On the Logic

d metric, our model is closest to the KE-GPT2 which bene-
fits from the large-scale corpus. It indicates that dependency
parsing and MGCN-DP on context are really beneficial to
grap 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 re-
moving 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

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

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

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

Table 3: Ablation studies on dependency relations which ap-
pear frequently in the dataset.
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 → already_stale → call_number → 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


