Enhancing Text Generation via Multi-Level Knowledge Aware Reasoning

Feiteng Mu , **Wenjie Li**
The Department of Computing, The Hong Kong Polytechnic University, Hong Kong
{csfmu,cswjli}@comp.polyu.edu.hk

Abstract

How to generate high-quality textual content is a non-trivial task. Existing methods generally generate text by grounding on word-level knowledge. However, word-level knowledge cannot express multi-word text units, hence existing methods may generate low-quality and unreasonable text. In this paper, we leverage event-level knowledge to enhance text generation. However, event knowledge is very sparse. To solve this problem, we split a coarse-grained event into fine-grained word components to obtain the word-level knowledge among event components. The word-level knowledge models the interaction among event components, which makes it possible to reduce the sparsity of events. Based on the event-level and the word-level knowledge, we devise a multi-level knowledge aware reasoning framework. Specifically, we first utilize event knowledge to make event-based content planning, i.e., select reasonable event sketches conditioned by the input text. Then, we combine the selected event sketches with the word-level knowledge for text generation. We validate our method on two widely used datasets, experimental results demonstrate the effectiveness of our framework to text generation.

1 Introduction

Text generation aims to produce realistic and reasonable textual content that is indistinguishable from human-written text, which is helpful for question answering, story generation [Mostafazadeh *et al.*, 2016], etc.

Existing works generally ground the generative models on word-level knowledge [Zhou *et al.*, 2018; Ji *et al.*, 2020] and successfully generate informative text. However, these methods may produce unreasonable text, because word-level knowledge cannot express multi-word text units. For example, in Figure 1, the existing state-of-the-art (SoTA) model GRF [Ji *et al.*, 2020] utilizes the word-level knowledge to generate an unreasonable sequence. This is because the word-level knowledge cannot reflect the actual semantics of the multi-word expression "scaredy cat".

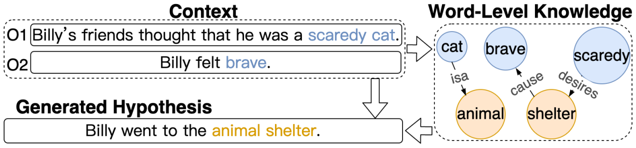


Figure 1: An unreasonable text is generated by GRF. The context comes from the task of αNLG.

Real-world knowledge exists at the word-level, the event-level, etc. It is inadequate for a text generation system to just ground on word-level knowledge. To make high-quality text generation, event-level knowledge also makes sense. An event is a tuple containing a subject, a verb, an object, and some additional token(s) [Radinsky *et al.*, 2012]. As the basic semantic unit of natural language, an event carries richer information than a single word, hence event knowledge might help the event-based content planning, and conditioned on event knowledge may help reasonable and diverse text generation. However, event knowledge is very sparse, which brings about the difficulty to make the reasonable content-planning. To solve this problem, we split a coarse-grained event into fine-grained word components to obtain the word-level knowledge among event components. The word-level knowledge models the interaction among event components, which makes it possible to reduce the sparsity of events.

Based on the event-level and the word-level knowledge, we devise a novel *Multi-Level Knowledge aware Reasoning* (MKR) framework for text generation. Our framework generates text through the following three steps. First, it uses a graph-based module to learn structure-aware embeddings for words and events (§ 3.2). This step models the interaction between event components, which makes it possible to reduce the sparsity of events. Following, it utilizes the event-level knowledge to make event-based content planning (§ 3.3), i.e., select reasonable event sketches conditioned by the input text. We use the selected event sketches as guidance for text generation. Finally, we combine the word-level knowledge with the selected event sketches (§ 3.4), to make full use of multi-level knowledge for text generation.

Our contributions can be summarized as follows: (1) We propose to split a coarse-grained event into fine-grained word components to obtain the multi-level knowledge. The multi-level knowledge models real-world knowledge in the event-

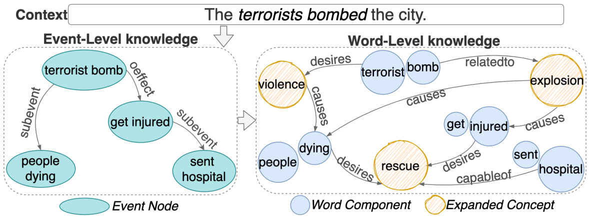


Figure 2: An example of constructing the multi-level knowledge graph for the input context.

level and the word-level, as well as models the interplay between event components; (2) We propose a novel multi-level knowledge aware reasoning framework for text generation. It utilizes both the event-level and the word-level knowledge, therefore it can generate high-quality text.

Experimental results on two widely used datasets demonstrate the effectiveness of our framework. We also make an in-depth analysis of event-level knowledge and word-level knowledge. The combination of two kinds of knowledge leads to the best generation results, which indicates that there is complementarity between the two kinds of knowledge.

2 Related Work

2.1 Knowledge-Aware Text Generation

Existing text generation methods focus on utilizing word-level knowledge to augment the limited textual information. [Zhou *et al.*, 2018] enriched the representations of the input text with neighbouring concepts on ConceptNet. [Zhang *et al.*, 2019a] explicitly models the conversation flow with the commonsense knowledge and guides the conversation flow in the latent concept space. [Ji *et al.*, 2020] integrates external knowledge into pretrained language models to generate more informative texts. Nevertheless, these methods just utilize word-level knowledge. Different from those works, we introduce multi-level knowledge to enhance text generation.

2.2 Text Generation with Content Planning

Text generation with content planning first identifies pertinent information to present, and then realizes the pertinent information into surface text [Holmes-Higgin, 1994]. [Puduppully *et al.*, 2019] incorporates the end-to-end content planning to data-to-text generation. [Xu *et al.*, 2020] enhanced the dialogue system by making the content planning grounded on narrative chains. [Goldfarb-Tarrant *et al.*, 2020] makes a plot-guided content planning and demonstrates its use in long-form story generation. In this paper, we make the content planning based on the event-level knowledge. In addition, we introduce the multi-level knowledge graph to reduce the sparsity of events, which is less studies in previous works.

3 Method

3.1 Preliminary

Building Multi-Level Knowledge Graph Given a context, we first construct a multi-level knowledge graph G for it. To obtain event-level knowledge for text generation, we use ATOMIC [Sap *et al.*, 2019] and COMeT [Hwang *et al.*, 2020] as event knowledge base. ATOMIC is a repository of inferential if-then knowledge. COMeT is a transformer model trained on ATOMIC that generates nine kinds of inferences of events in natural language. Given the input context, we first extract central events from the context and feed the central events into COMeT to generate one-hop events with corresponding relations. The one-hop events are then fed into COMeT to generate two-hop events. After that, we obtain many event paths. Several heuristic rules are applied to filter low-quality paths. For example, if an event in a path contains less than 2 words, the path will be discarded. There are still many event paths left, we randomly select at most 80 paths. We next split each event into word components. For each component, we expand at most three word-level relations from ConceptNet[1]. Finally, we look up word-level relations among all word pairs. In this way, originally unrelated events can interact through word-level relations between event components. Figure 2 illustrates the construction process. We heuristically use the word-overlap ratio to label words and events. That is, a word is labeled as positive if it is contained by the gold sequence. An event is labeled as positive if 70% of event components are contained by the gold sequence. The event labels are used as supervision for selecting event sketches (§ 3.3), and word labels are used as supervision for gate control (§ 3.4) in the decoding phase.

Task Definition The input $X = (x_1, x_2, \cdots, x_M)$ is a text sequence which may consist of several sentences. We aim to generate another sequence $Y = (y_1, y_2, \cdots, y_N)$. According to X, we extract a multi-level knowledge graph G. There are two types nodes in G: (1) word nodes $\mathcal{V}_w$; (2) event nodes $\mathcal{V}_e$. Each event $e_i \in \mathcal{V}_e$ contains several word components $e_i = \{w_{i1}, \cdots, w_{ik}\}, (w_{i1}, \cdots, w_{ik} \in \mathcal{V}_w)$. Also, There are two types of edges in G: (1) event-event relation (h_e, r_e, t_e) indicating the head event h_e has a relation r_e to the tail event t_e; (2) word-word relation (h_w, r_w, t_w) indicating the head word h_w has a relation r_w to the tail word t_w, where h_w and t_w may be the components of different events. We decompose the text generation task into two steps. The first step uses the event knowledge to make a event-based content planning, i.e., select reasonable event sketches E_k conditioned by the X. In the second step, we use word-level knowledge to generate the text sequence given X and E_k. Our model maximizes the following conditional probability:

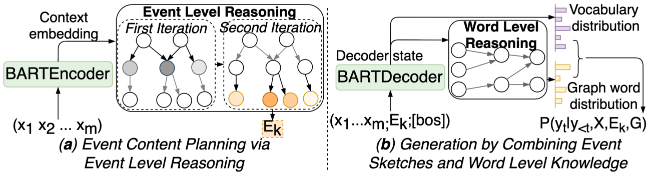
$$P(Y|X, G) = P(E_k|X, G) \cdot P(Y|X, G, E_k). \tag{1}$$

3.2 Multi-Level Knowledge Graph Encoding

We use Relational Graph Convolutional Network (RGCN) [Schlichtkrull *et al.*, 2018] to encode our multi-level knowledge graph to learn structure-aware embedding of words, events, and relationships. Given a RGCN with L layers, for each word $t_w \in \mathcal{V}_w$, we update its word embedding at the $l+1$-th layer by aggregating its local neighbours $\mathcal{N}(t_w)$ including pairs of the word h_w and the connected relation r_w:

$$\mathbf{h}_{t_w}^{l+1} = \sigma\Big(\frac{1}{Z_{t_w}} \sum_{(h_w, r_w) \in \mathcal{N}(t_w)} \mathbf{W}_a^l(\mathbf{h}_{h_w}^l - \mathbf{h}_{r_w}^l) + \mathbf{W}_s^l \mathbf{h}_{t_w}^l\Big), \tag{2}$$

[1] In ConceptNet, each relation is assigned with a weight. We choose 3 relations with the highest weight.



(*a*) *Event Content Planning via Event Level Reasoning*

(*b*) *Generation by Combining Event Sketches and Word Level Knowledge*

Figure 3: Our two-stage reasoning process. In (a), the event-level reasoning module iteratively computes the relevance scores (indicated by the color intensity) of one-hop events and two-hop events. In each iteration, the grey arrows denote unused edges, while the black arrows denote used edges. In (b), the word-level reasoning process is similar to the event-level reasoning process in (a).

where $\mathbf{h}^0_{t_w}$ is initialized by pretrained embedding and $\mathbf{h}^0_{r_w}$ by randomly, $Z_{t_w} = |\mathcal{N}(t_w)|$ is the number of neighbors of t_w, $\mathbf{W}^l_s$ and $\mathbf{W}^l_a$ are trainable specific to the l-th layer, σ is the RELU activation. We also update embeddings of word-word relations via another linear transformation $\mathbf{h}^{l+1}_{r_w} = \mathbf{W}^l_r \mathbf{h}^l_{r_w}$. After L iterations, words embeddings $\mathbf{h}^L_{h_w}$, $\mathbf{h}^L_{t_w}$ and relations embeddings $\mathbf{h}^L_{r_w}$ are obtained. Each event embedding is obtained through pooling on its word components:

$$\mathbf{h}_{e_i} = \text{max-pooling}(\mathbf{h}^L_{w_{i1}}, \cdots, \mathbf{h}^L_{w_{ik}}), \tag{3}$$

where $\mathbf{h}^L_{w_{i1}}, \cdots, \mathbf{h}^L_{w_{ik}}$ is the output of L-th layer of RGCN.

3.3 Event Content Planning via Event Reasoning

Given the context X and the graph G, we perform the event-based content planning to produce the event sketches.

We first encode $X = (x_1, x_2, \cdots, x_M)$ into the hidden state $\mathbf{h}_X \in \mathcal{R}^d$ (d is the hidden size):

$$\begin{aligned} \{\mathbf{h}_{x_m}\}_{m=1,\cdots,M} &= \text{BARTEncoder}(X) \\ \mathbf{h}_X &= \mathbf{W}_x(\text{max-pooling}_m(\mathbf{h}_{x_m})), \end{aligned} \tag{4}$$

where $\mathbf{h}_{x_m}$ is the vector of the m-th token, $\mathbf{W}_x$ is trainable.

Then we perform the event-level reasoning to select event sketches. Because how to realize reasoning is not the focus of this paper, we directly adopt the existing approach: multi-hop reasoning flow [Ji *et al.*, 2020]. We iteratively compute the relevance scores of one-hop events and two-hop events in accordance with the context, as shown in Figure 3. In each iteration, the module parallelly computes relevance scores of events which are in the same hop. For the event t_e, the score $cs(t_e)$ between t_e and X is computed by aggregating evidence from its neighbours $\mathcal{N}_{t_e}$ including pairs of the connected event h_e and the connected edge r_e:

$$cs(t_e) = \frac{1}{|\mathcal{N}_{t_e}|} \sum_{(h_e,r_e)\in\mathcal{N}_{t_e}} (\gamma \cdot cs(h_e) + R(h_e, r_e, t_e)), \tag{5}$$

where γ (0.5 by default) is a discount factor that controls the score flow from the previous hops. Initially, zero-hop events (central events) are given a score of 1, while other events are assigned with 0. $R(\cdot)$ is the relevance of the relation (h_e, r_e, t_e) under the context $\mathbf{h}_X$, which is calculated by

$$\begin{aligned} R(h_e, r_e, t_e) &= \text{sigmoid}(\tanh(\mathbf{h}_X^\top \mathbf{W}_{cs}) \cdot \mathbf{h}_{(h_e,r_e,t_e)}) \\ \mathbf{h}_{(h_e,r_e,t_e)} &= [\mathbf{h}_{h_e}; \mathbf{h}_{r_e}; \mathbf{h}_{t_e}], \end{aligned} \tag{6}$$

where $\mathbf{W}_{cs}$ is trainable.

According to the scores of all events, we select the top-k events $E_k = \text{topk}_i(cs(e_i))$, which are used as guidance for text generation. We also explore the influence of k in the experiments.

3.4 Generation by Combining Event Sketches and Word Level Knowledge

We use the selected events E_k as guidance for text generation. To make full use of multi-level knowledge, we also use word knowledge in generation process. Specifically, taking the context X, the multi-level graph G and the event sketches E_k as input, the generation module first encodes X and E_k to obtain the guided context vectors $\mathbf{H}_C = \text{BARTEncoder}([X; E_k])$ where $[\cdot;\cdot]$ denotes concatenation operation, $\mathbf{H}_C \in \mathcal{R}^{c\times d}$ (c is the total length of $[X; E_k]$). The hidden state of t-th time step of the target sequence $\mathbf{h}_{y_t}$ is computed by:

$$\mathbf{h}_{y_t} = \text{BARTDecoder}(\mathbf{Y}_{\leq t}, \mathbf{H}_C). \tag{7}$$

The word distribution of t-th time-step over the standard vocabulary V is

$$P(y_t|Y_{<t}) = \text{softmax}_V(\mathbf{W}_v \mathbf{h}_{y_t} + \mathbf{b}). \tag{8}$$

The generation module then performs word-level reasoning to consider the word level knowledge according to the current decoder state $\mathbf{h}_{y_t}$. For the word t_w, the module computes the score $cs(t_w)$ between t_w and $[X; E_k]$ by aggregating evidence from its neighbours $\mathcal{N}_{t_w}$ including pairs of the connected word h_w and the connected edge r_w:

$$cs(t_w) = \frac{1}{|\mathcal{N}_{t_w}|} \sum_{(h_w,r_w)\in\mathcal{N}_{t_w}} (\gamma \cdot cs(h_w) + R(h_w, r_w, t_w)), \tag{9}$$

where $R(h_w, r_w, t_w)$ is computed by:

$$\begin{aligned} R(h_w, r_w, t_w) &= \text{sigmoid}(\mathbf{h}_{y_t}^\top \mathbf{W}_{wv} \mathbf{h}_{(h_w,r_w,t_w)}) \\ \mathbf{h}_{(h_w,r_w,t_w)} &= [\mathbf{h}^L_{h_w}; \mathbf{h}^L_{r_w}; \mathbf{h}^L_{t_w}]. \end{aligned} \tag{10}$$

Initially, the zero-hop words, which are existed in X, are given a score of 1, while other words are assigned with 0. After H iterations, the distribution over all words in G is:

$$P(y^w_t|Y_{<t}, G) = \text{softmax}_{\mathcal{V}_w}(cs(t_w)), \tag{11}$$

where $y^w_t \in \mathcal{V}_w$ is the selected word at the t-th time step.

The final token distribution combines the distribution over word nodes in $\mathcal{V}_w$ and the distribution over the standard vocabulary with a soft gate $g_t = \text{sigmoid}(\mathbf{W}_g \mathbf{h}_{y_t})$, which determines whether to copy words from G when generation:

$$\begin{aligned} P(y_t|Y_{<t}, X, G) = (1 - g_t) \cdot P(y^w_t|Y_{<t}, G) \\ + g_t \cdot P(y_t|Y_{<t}). \end{aligned} \tag{12}$$

3.5 Model Training

To train the event planning module, we minimize the cross-entropy loss of selecting positively-labeled events by:

$$J_P = \frac{1}{Z} \sum_i (-l_i \cdot log(p(e_i)) - (1 - l_i) \cdot log(1 - p(e_i))), \tag{13}$$

where $p(e_i) = \text{sigmoid}(cs(e_i))$ denotes the probability that the event e_i is selected, l_i is the label of e_i, Z is the number of events in G.

To train the text generator, we minimize the NLL loss of generating the gold sequence:

$$J_{\text{NLL}} = \sum_{t=1}^{N} -\log P(y_t^{\text{gold}} | Y_{<t}^{\text{gold}}, X, E_k, G), \qquad (14)$$

where E_k is the selected event sketches (§ 3.3). We additionally add the gate loss J_g to supervise the training of g_t, the loss takes the form of binary cross-entropy. The final loss of the generator is $J_G = J_{\text{NLL}} + \alpha J_g$ where α is set to 0.5 by default. The final loss is $J = J_P + J_G$. The event planner and the text generator are jointly training.

4 Experiments Settings

4.1 Datasets

Story Ending Generation (SEG) is to generate a reasonable ending given a four-sentence story context. The stories come from ROCStories [Mostafazadeh *et al.*, 2016] corpus. Following [Yao *et al.*, 2019], we randomly split the dataset into 8:1:1 for training, validating and testing.

Abductive NLG (αNLG) is to generate an explanatory hypothesis given two observations: O_1 as the cause and O_2 as the consequence. We use the official data split.

4.2 Baselines and Implementation Details

GPT2-FT is a GPT-2 model fine-tuned on the task-specific dataset with initialization from [Radford *et al.*, 2019]. **T5-FT** is a T5 model fine-tuned on the task-specific dataset with initialization from [Raffel *et al.*, 2019]. **BART-FT** is a BART model fine-tuned on the task-specific dataset with initialization from [Lewis *et al.*, 2019]. **GPT2-OMCS** is a commonsense-enhanced GPT-2 model first post-trained on the Open Mind Common Sense (OMCS) corpus5 from which the ConceptNet is constructed. The model is then fine-tuned on the task-specific dataset. **CALM-T5** [Zhou *et al.*, 2020], which utilizes the self-supervised contrastive learning to inject word-level (nouns and verbs) knowledge into pre-trained T5 model. **GRF-GPT2** [Ji *et al.*, 2020], the current SoTA model on the used datasets, is a GPT2 based model which generates text with multi-hop reasoning on structural knowledge graphs. For the sake of fairness, we also use the BART pretrained model to reproduce GRF and name it **GRF-BART**.

Our method employs the base version of the BART model, and a 2-layer RGCN module. To train the model, we use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-6}$ and linearly decrease learning rate to zero with no warmup. We search for the best hyper-parameters according to BLEU-2 on the development set of each dataset. At the inference stage, we adopt beam search decoding with a beam size of 3 for our model and all the baselines we produce.

5 Results and Analysis

5.1 Automatic Evaluation

Metrics include: *BLEU-n* [Papineni *et al.*, 2002], *METEOR* [Banerjee and Lavie, 2005], *BertScore* [Zhang *et al.*, 2019b] and *Distinct-n* [Li *et al.*, 2015] .

αNLG				
Models	BLEU-4	METEOR	D-2/D-3	BertS.
GPT2-FT	9.80	25.82	N/A	N/A
T5-FT	12.65	29.08	10.32/16.36	55.73
BART-FT	12.93	29.85	10.34/16.55	55.47
GPT2-OMCS	9.62	25.83	N/A	N/A
CALM-T5	12.91	29.18	10.41/16.37	55.77
GRF-GPT2	11.62	27.76	N/A	N/A
GRF-BART	14.42	31.29	10.43/16.14	56.04
MKR (ours)	**15.74**	**32.74**	**15.41/26.32**	**56.61**

SEG				
Models	BLEU-1/2	METEOR	D-2/D-3	BertS.
GPT2-FT	25.5/10.2	N/A	N/A	N/A
T5-FT	24.2/9.4	17.52	24.50/43.48	48.31
BART-FT	25.6/10.4	18.75	24.27/45.54	49.08
GPT2-OMCS	25.5/10.4	N/A	N/A	N/A
CALM-T5	24.2/9.2	17.37	23.71/42.16	48.28
GRF-GPT2	26.1/11.0	19.36	N/A	N/A
GRF-BART	26.1/11.2	19.64	29.02/52.01	49.76
MKR (ours)	**27.4/12.8**	**21.41**	**29.86/53.84**	**50.75**

Table 1: The result of automatic evaluation. D-2/D-3 denotes Distinct-2/3. BertS. denotes BertScore.

Our Method vs. Baselines

The result is shown in Table 1. In the used pretrained models, BART performs best, so we use BART as our backbone. Our implementation of GRF (GRF-BART) performs better than the original one (GRF-GPT2), the improvement is caused by the BART model. GRF-BART performs better than GPT2-OMCS and CALM-T5. This indicates that explicitly performing reasoning on structural knowledge graphs is more effective than implicitly injecting knowledge into PLMs. This coincides with [Ji *et al.*, 2020]. Compared with GRF-BART, our model achieves a notable improvement. This is because GRF-BART only utilizes the word-level knowledge, and our method utilizes the event-level knowledge as well. The distinct score of our method is more significant than baselines, which indicates that our method can generate more diverse text. This may be because the multi-level knowledge provides diverse background information for text generation. The overall result indicates that our framework is effective for improving the quality of text generation.

Ablation Study

Settings To investigate the effectiveness of different components, we further conduct ablation study to compare our model with the following variants: (1) "w/o J_p" means we do not supervise the event content planning module; (2) "w/o MKE" means we remove the multi-level graph encoding module, that is we do not model interaction between event components; (3) "w/o ECP" means we do not utilize event knowledge and only use word knowledge when decoding stage; (4) "w/o GWK" means we do not utilize word knowledge when decoding stage.

α**NLG**				
Models	BLEU-4	METEOR	D-2/D-3	BertS.
MKR (full)	15.74	32.74	15.41/26.32	56.61
w/o ECP	14.34	31.63	10.19/15.74	56.19
w/o GWK	15.20	32.33	13.65/22.80	56.48
w/o MKE	15.55	32.55	13.91/23.28	56.42
w/o J_p	15.31	32.43	10.85/16.83	56.30

SEG				
Models	BLEU-1/2	METEOR	D-2/D-3	BertS.
MKR (full)	27.4/12.8	21.41	29.86/53.84	50.75
w/o ECP	25.8/10.9	19.22	27.28/48.75	49.44
w/o GWK	26.9/12.4	20.98	28.86/51.90	50.64
w/o MKE	27.0/12.5	21.09	29.38/52.66	50.65
w/o J_p	26.4/11.8	20.22	26.56/47.93	50.04

Table 2: Ablation study result. MKE denotes the multi-level graph encoding (§ 3.2). ECP denotes the event content planning (§ 3.3). GWK denotes text generation with word-level knowledge (§ 3.4).

Result The result is shown in Table 2. Each component contributes to text generation. "w/o MKE" leads to a performance drop, this is because the model does not model the interaction between event components, and the sparsity of events damages the event content planning. "w/o J_p" leads to a larger drop compared to "w/o MKE". This coincides with human intuition that the event labels help to make a more reasonable event sketches selection. In addition, we have the following observations. (1) Compared with word-level knowledge, event-level knowledge is more important for text generation. For example, the removal of event knowledge ("w/o ECP") leads to a huge performance drop. This makes sense, because event knowledge is naturally more diverse than word knowledge and carries richer information than word knowledge. Hence event guidance enables the model to generate more diverse and high-quality text. (2) Although not as important as event knowledge, word knowledge is also useful for text generation. The best results (our full model) are obtained by combining the word-level knowledge and the event -level knowledge. The reasons lie in two aspects: (1) word-level knowledge helps event representation learning, which is helpful for making reasonable event sketches planning; (2) Being aware of multi-level knowledge, a model has the ability to generate high-quality text.

5.2 Manual Evaluation

Criteria includes informativeness and reasonability. When evaluating informativeness, the annotators are required to assess whether the generated text produce unique and non-genetic information that is specific to the input context. When evaluating reasonability, annotators are asked to focus on evaluating the causal and temporal relevance of the generated results and the contexts. We carried out pairwise comparison with BART-FT, GRF-BART, and two ablated models "w/o ECP" and "w/o GWK". We randomly sample 100 cases from the two testsets respectively for each pair of models. Three annotators are recruited to make a preference among win, tie and lose given the input context and two outputs generated by our model and a baseline respectively. The annotators are research students from the field of text generation to make sure they have a fair judgement of used metrics. The result is shown in Table 3. Our full model outperforms all compared models. We calculate the Fleiss's kappa reliability as the inter-rater agreement. For αNLG, the agreement of informativeness and reasonability is 0.497 and 0.459, respectively. And for SEG, the two values are 0.424 and 0.451.

Datasets	Models	Informativeness		Reasonability	
		W(%)	L(%)	W(%)	L(%)
αNLG	vs. BART-FT	25.00	13.33	35.00	11.00
	vs. GRF-BART	25.67	11.33	31.00	11.33
	vs. w/o ECP	25.33	12.67	33.67	15.00
	vs. w/o GWK	24.00	10.33	28.00	13.00
SEG	vs. BART-FT	23.67	6.00	25.33	11.00
	vs. GRF-BART	20.33	7.33	21.67	11.00
	vs. w/o ECP	21.67	7.00	21.33	9.67
	vs. w/o GWK	16.00	6.00	14.00	7.00

Table 3: Manual evaluation results on two datasets. Scores indicate the percentage of Win (W) and Lose (L).

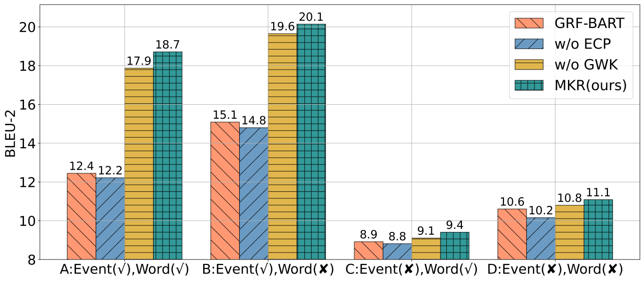


Figure 4: In group A, the multi-level graph of each sample contains at least a supporting event and a supporting word. Similarly, in group B, the multi-level graph of each sample contains at least a supporting event, but does not contain supporting words.

5.3 Deeper Analysis of Multi-Level Knowledge

We then investigate how multi-level knowledge helps text generation. The compared models include GRF-BART, our full method MKR, and two ablated versions: "w/o ECP" and "w/o GWK". We divide the test samples of the SEG task into four groups, according to whether the multi-level knowledge graph of a test sample contains at least a positively-labeled event or a positively-labeled word[2]. The number of samples in each group is (A) 374, (B) 1476, (C) 1369 and (D) 6596. In each group, we calculate the BLEU-2 score of generated text of different models. From Figure 4, we observes that:

(1) In group A and B, it is possible for "w/o GWK" to select a supporting event for generation, hence "w/o GWK" performs far better than "w/o ECP" and GRF-BART.

[2] For simplicity, we call positively-labeled events or words as supporting events or words.

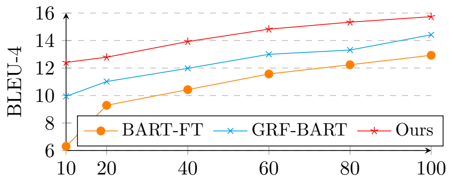


Figure 5: The impact of different size (%) of αNLG training data.

(2) In group C, the result gap between "w/o ECP" and "w/o GWK" was the smallest. This is because it is possible for "w/o ECP" to select the supporting words for generation, but it is impossible for "w/o GWK" to select any supporting event. This reduces the performance gap between the two variants. Note that event knowledge is more sparse than word knowledge. When a multi-level knowledge graph does not contain supporting events, it may contain supporting word-level knowledge, which is also useful for text generation.

(3) In all groups, the best result is obtained after combining the word-level with event-level knowledge. This is because the model is aware of multi-level knowledge, so it is more easier to generate high-quality text.

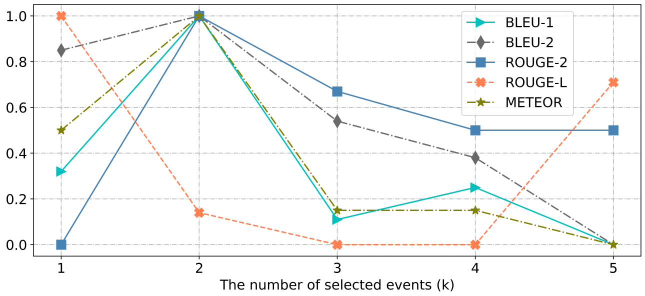


Figure 6: The result under the different number of selected events. We min-max-normalize the results of different metrics to 0-1.

5.4 Further Discussion

Impact of the training data size In Figure 5, we gradually reduce the training data size of αNLG, and test on the original testset. Our method achieve consistent performance gains, even when given extremely small training data (10%), but the result of GRF-BART drops more dramatically. This demonstrates the generalization ability of our model with the aid of multi-level knowledge.

Impact of the number of selected events We investigated the impact of the number of selected events on the generation quality on the αNLG testset, as shown in Figure 6. Most of metrics first increase, reach the maximum at $k = 2$, and then decrease. On the one hand, the greater the k, the more likely it is to select the support event. On the other hand, the more selected events, the more likely it is to select irrelevant events. When $k = 2$, the two aspects reach the good balance, so our method obtains the best performance. Therefore, we set $k = 2$ for all the main results of our model.

O_1	Tim wanted to quit smoking.
O_2	Soon Tim was smoke-free!
BART-FT	Tim started smoking regularly.
GRF-BART	Tim went to the doctor.
w/o ECP	Tim went to the doctor.
w/o GWK	Tim got a nicotine patch.
MKR (ours)	Tim went to the doctor and got prescribed nicotine patches.
O_1	It was a bright, warm day.
O_2	Joe regret going outside.
BART-FT	Joe went outside and it started to rain.
GRF-BART	Joe went outside to play.
w/o ECP	Joe decided to go outside.
w/o GWK	Joe forgot to put on his jacket.
MKR (ours)	Joe got sunburned in the sun.

Table 4: Cases with the generated text of compared models.

5.5 Case Study

We provide two cases which are from the αNLG testset, and present the generated sentences of corresponding models in Table 4. We find that: (1) Baseline models sometimes fail to generate reasonable sentences, i.e., the generated text of BART-FT in the first case. In contrast, our method generates reasonable sentences which are causally relevant to the context. (2) By utilizing the multi-level knowledge, our full model generates more informative sentences than the model "w/o GWK" which only utilizes the event-level knowledge.

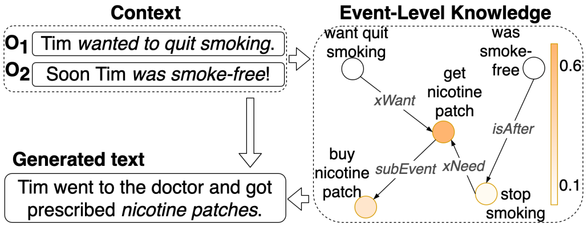


Figure 7: The extracted event level subgraph for the first case. The darker orange indicates the higher relevance score (Equation 5).

We visualize a part of the event-level knowledge of the first case, as shown in Figure 7. The event "*get nicotine patch*" receives the highest score, follows by the event "*buy nicotine patch*". The two events are selected as event sketches and are used for generation.

6 Conclusion

We present a multi-level knowledge aware reasoning framework for text generation. It is a two-stage reasoning framework which utilizes both the event-level and the word-level knowledge. Experiments demonstrate that word-level knowledge and event-level knowledge complement each other, and both contribute to text generation.

Acknowledgements

This work is supported by Research Grants Council of Hong Kong(PolyU/15207920, PolyU/15207821), National Natural Science Foundation of China (62076212) and PolyU internal grants (ZVQ0, ZVVX).

References

[Banerjee and Lavie, 2005] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. *Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, page 65, 2005.

[Goldfarb-Tarrant *et al.*, 2020] Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. Content planning for neural story generation with aristotelian rescoring. *arXiv:2009.09870*, 2020.

[Holmes-Higgin, 1994] Paul Holmes-Higgin. Text generation—using discourse strategies and focus constraints to generate natural language text by kathleen r. mckeown, cambridge university press, 1992, pp 246,£ 13.95, isbn 0-521-43802-0. *The Knowledge Engineering Review*, 9(4):421–422, 1994.

[Hwang *et al.*, 2020] Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. *arXiv preprint arXiv:2010.05953*, 2020.

[Ji *et al.*, 2020] Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. Language generation with multi-hop reasoning on commonsense knowledge graph. *arXiv preprint arXiv:2009.11692*, 2020.

[Lewis *et al.*, 2019] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[Li *et al.*, 2015] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

[Mostafazadeh *et al.*, 2016] Nasrin Mostafazadeh, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. pages 839–849, 2016.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[Puduppully *et al.*, 2019] Ratish Puduppully, Li Dong, and Mirella Lapata. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6908–6915, 2019.

[Radford *et al.*, 2019] 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.

[Radinsky *et al.*, 2012] Kira Radinsky, Sagie Davidovich, and Shaul Markovitch. Learning causality for news events prediction. In *Proceedings of the 21st international conference on World Wide Web*, pages 909–918, 2012.

[Raffel *et al.*, 2019] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[Sap *et al.*, 2019] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035, 2019.

[Schlichtkrull *et al.*, 2018] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.

[Xu *et al.*, 2020] Jun Xu, Zeyang Lei, Haifeng Wang, Zheng-Yu Niu, Hua Wu, and Wanxiang Che. Enhancing dialog coherence with event graph grounded content planning. In *IJCAI*, pages 3941–3947, 2020.

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

[Zhang *et al.*, 2019a] Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. Grounded conversation generation as guided traverses in commonsense knowledge graphs. *arXiv:1911.02707*, 2019.

[Zhang *et al.*, 2019b] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[Zhou *et al.*, 2018] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629, 2018.

[Zhou *et al.*, 2020] Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, Bill Yuchen Lin, and Xiang Ren. Pre-training text-to-text transformers for concept-centric common sense. *arXiv preprint arXiv:2011.07956*, 2020.