Complex & Intelligent Systems (2023) 9:2019–2030
https://doi.org/10.1007/s40747-022-00898-0

ORIGINAL ARTICLE

Enriched entity representation of knowledge graph for text generation

Kaile Shi[1] · Xiaoyan Cai[1] · Libin Yang[1] · Jintao Zhao[1]

Received: 16 November 2021 / Accepted: 17 October 2022 / Published online: 4 November 2022

Abstract

Text generation is a key tool in natural language applications. Generating texts which could express rich ideas through several sentences needs a structured representation of their content. Many works utilize graph-based methods for graph-to-text generation, like knowledge-graph-to-text generation. However, generating texts from knowledge graph still faces problems, such as repetitions and the entity information is not fully utilized in the generated text. In this paper, we focus on knowledge-graph-to-text generation, and develop a multi-level entity fusion representation (MEFR) model to address the above problems, aiming to generate high-quality text from knowledge graph. Our model introduces a fusion mechanism, which is capable of aggregating node representations from word level and phrase level to obtain rich entity representations of the knowledge graph. Then, Graph Transformer is adopted to encode the graph and outputs contextualized node representations. Besides, we develop a vanilla beam search-based comparison mechanism during decoding procedure, which further considers similarity to reduce repetitive information of the generated text. Experimental results show that the proposed MEFR model could effectively improve generation performance, and outperform other baselines on AGENDA and WebNLG datasets. The results also demonstrate the importance to further explore information contained in knowledge graph.

Keywords Text generation · Knowledge graph · Multi-level entity fusion representation model · Comparison mechanism

Introduction

Natural text generation refers to the task of automatically producing texts from linguistic and non-linguistic input [1]. According to the style of input data, text generation can be categorized as text-to-text generation methods [2], data-to-text generation methods [3], and image-to-text generation methods [4].

In specific domains like medical or scientific area, it is hard to generate texts which express complex ideas of content with a reasonable and logical structure. Some researches address this issue with a structured representation of input, which can be benefit to understand the content [5]. They utilize rule-based methods or template-based methods for structured-data-to-text generation [6–8]. These methods are usually easy to guarantee correctness of the generated texts' content, due to their interpretability and controllability. However, they also face some limitations, i.e., high-quality template is hard to extract without manual process; generated content often meets with problems in terms of diversity, fluency, and consistency. Recent neural network-based generation methods are driven by data, and they do not require much manual intervention and mainly rely on representation learning, to select appropriate content and express grammatically [9]. Although structured input could provide more additional guidance for generation [10,11], neural network-based generation methods still have a variety of logical errors, like hallucinating statements which are not supported by the facts contained in the input, and confusing the output location of different information.

Therefore, researchers began to focus on graph-based neural network methods aiming to effectively capture global structure of the input, and preserve more original information to overcome the above issues [12–15]. For example, Koncel-Kedziorski et al. [16] proposed a Graph Transformer that extends Transformer [17] for encoding the input graph, built on graph attention network (GAT) [18] architecture. Although graphs could effectively capture both global and local structure of the input as well as further improving generation performance, the generated text are still affected by

✉ Xiaoyan Cai
xiaoyanc@nwpu.edu.cn

✉ Libin Yang
libiny@nwpu.edu.cn

[1] School of Automation, Northwestern Polytechnical University, Xi'an, Shaanxi, People's Republic of China

repetitions, and at the same time, entities which act as a key part of the graph are not fully covered in generated text.

In this paper, we focus on knowledge-graph-to-text generation, and propose a multi-level entity fusion representation (MEFR) model, which aims to address issues of repetitions and entity information is not fully covered in the generated text, further enhancing generation performance. First, we follow the similar procedure with previous work to pre-process the input knowledge graph, where a vertex denotes an entity node, or a relation node which is created for each edge relation between two entities, or a global node which connects all entity nodes. For the processed knowledge graph, we propose a fusion mechanism by aggregating node information from word level and phrase level to obtain entity representations in the graph. Then, we apply Graph Transformer [16] to encode the input knowledge graph and obtain the contextualized representation for each node. When decoding, vanilla beam search [19,20] is adopted, which is a global optimum-based search algorithm, that usually applying in text generation to select the results with top-k scores. To further reduce redundancy of the generated text, we develop a vanilla beam search-based comparison mechanism, which considers whether adding the generating word to the generated word sequence based on similarity. Experimental results show that our proposed MEFR model could effectively improve quality of the generated text. Three main contributions of this paper are:

- Multi-level fusion mechanisms are developed, i.e., sum fusion mechanism and selective mechanism, which aggregate information from word level and phrase level to obtain entity representations.
- A comparison mechanism during generation is proposed, which considers similarity between the generated sequence with and without the generating word, tackling the constraints of redundancy and enhancing the performance of generation.
- Thorough experimental studies are conducted to verify the effectiveness of the proposed model, and our proposed model which achieves great performance without pre-trained language models also illustrates the importance of further exploring the information contained in knowledge graph.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 explains the proposed MEFR model. Section 4 presents the experiments and evaluation results. Conclusions are presented in Sect. 5.

Related work

For structured-data-to-text generation, the core task is to generate a textual descriptions based on structural knowledge records. Some generation systems rely on rules and hand-engineered templates. Angeli et al. [21] constructed a domain-independent model, which manually designs a template to introduce knowledge of other domains for table to text generation. The model makes it easy to incorporate domain-specific knowledge which can improve generation performance. Kondadadi et al. [22] proposed a system which generates different content based on a specific domain. The system is also used statistical data of the text in addition, but it is restrained by requiring a lot of historical data. Howald et al. [23] presented a hybrid natural language generation system that utilizes Discourse Representation Structures (DRSs) for statistically learning syntactic templates. This model could generate acceptable texts for a number of different domains. Wiseman et al. [7] used hidden semi-Markov model (HSMM) to model text generation template and combined end-to-end methods with traditional template-based methods. On the other hand, many works focus on neural network-based end-to-end model in recent years. Mei et al. [3] used a neural encoder–decoder model to generate weather forecasts and soccer commentaries, and they also added an aligner to select important information based on end-to-end model. Juraska et al. [24] proposed a deep ensemble framework for text generation, which integrates sequence-to-sequence model based on bidirectional LSTM and CNN. This framework also used an automatic slot alignment-based reranking method which helps improve quality of the generated text. Gehrmann et al. [25] introduced multiple decoders to fit different data based on traditional encoder–decoder model. And in this way, the model could be used to generate different expressions for different types of text. Freitag et al. [26] interpreted structured data as a corrupt representation of the desired output, and used a denoising auto-encoder to reconstruct the sentence. The result shows that denoising auto-encoder could generalize to generate correct sentences when given structured data.

Although structured input could provide more guidance and structural information for generation, it is still restrained by how to better make use of the structure. Many researches began to focus on graph-based methods which can better capture local and global structure of input. Xu et al. [12] proposed a graph-to-sequence neural network model, which illustrates the structured input information is important for text generation, solving the problem of structural information loss caused by traditional graph-to-text generation methods. Beck et al. [13] used an encoder based on Gated Graph Neural Network (GGNN) which can integrate the complete graph structure without losing information, and introduce graph transformation providing more information for the attention and decoding modules in the network. Li et al. [14] modeled news as a topic interaction graph, which better understands internal structure of the article and the connection between topics. Xu et al. [27] converted SQL into a directed graph and used a graph-to-sequence model to translate the graph into a sequence. Koncel-Kedziorski et al. [16] proposed a Graph Transformer model to encode the knowledge graph, used for generating a text that can express the content of the knowledge graph. Song et al. [28] leveraged richer training signals to guide the model to preserve original information, tackling the problem of messing up or even dropping the core structural information of input graphs during generation. Based on graph convolutional networks, Guo et al. [29] developed a novel network named DCGCNs, achieving advanced performance on AMR-to-text generation. Ribeiro et al. [15] presented four neural models, which could combine both local and global contextual information for graph encoding. Despite their success, how to effectively utilize more information within graph for text generation is still an open problem.

To obtain more information of the knowledge graph for generation, we develop an MEFR model to obtain entity representations in the graph, by proposing fusion mechanisms to incorporate information from word-level and phrase-level representations. The proposed fusion mechanisms could enrich entity representations based on the above two-level information as well as improving generation performance.

Multi-level entity fusion representation (MEFR) model

Figure 1 shows the framework of our proposed MEFR model. The input of the model is a knowledge graph corresponding to the document, and the title within the graph if it exists. We follow the previous work [15,16] to pre-process the input knowledge graph denoting as $G = (V, E)$. V denotes a vertex set containing three types of nodes: entity nodes, relation nodes which represent relations between two entity nodes, and a global node which connects all entity nodes. E is an adjacency matrix describing the directed edges. The input graph and the title are encoded using a Graph Transformer [16] and a bidirectional recurrent neural network [30], respectively. We treat the title as an additional node, and use both node representations within the graph and title representation for decoder. When decoding, we take attention-based RNN [31] as decoder and adopt copy mechanism [32] for generation. The final output of MEFR model is the generated descriptive text. Details of the model will be illustrated in this section.

Encoder

Node embeddings

There are three types of nodes in the graph, i.e., entity nodes, relation nodes and a global node. As each relation is represented as both a forward direction relation node and a backward direction relation node, we learn two embeddings per each relation node. We also learn an initial embedding for the global node. However, entities in scientific texts are often multi-word expressions, we use BiRNN to obtain the embedding of each entity based on word embeddings as

$$\mathbf{h}^{w}_{p_j} = \mathrm{BiRNN}\left(\mathbf{x}^{1}_{p_j}, \mathbf{x}^{2}_{p_j}, \ldots, \mathbf{x}^{i}_{p_j}, \ldots, \mathbf{x}^{t}_{p_j}\right), \qquad (1)$$

where $\mathbf{x}^{i}_{p_j}$ is i-th word embedding of entity p_j, t denotes the number of words in p_j. The last hidden state is used as the word-level representation of the entity p_j, denoted as $\mathbf{h}^{w}_{p_j}$.

Besides, there exist relationships among entities, such as sequential relationships and logical relationships. For example, the appearance position of entities in the input is chronological and some entities always appear before or behind other entities. Based on the above analysis, we aim to capture more information for entity representations based on relationships among entities.

Compared to word-level entity embeddings, we also apply BiRNN which is applied to each entity to capture the dependency, and obtain phrase-level representations for entities, as

$$\mathbf{h}^{p}_{p_j} = \mathrm{BiRNN}\left(\mathbf{h}^{w}_{p_1}, \ldots, \mathbf{h}^{w}_{p_m}\right), \qquad (2)$$

where m is the number of entities in the knowledge graph, and $\mathbf{h}^{p}_{p_j}$ is the phrase-level entity representation of p_j.

Then, we propose two fusion mechanisms, i.e., sum fusion mechanism and selective mechanism, to integrate information from word-level representation and phrase-level representation of each entity. Although word embeddings are the same for the above two-level representations, the choice of context is changed when information is fused.

- **Sum fusion mechanism**
  We develop two methods for sum fusion mechanism. We first use a sum operation to fuse the above word-level and phrase-level entity representations as

$$\mathbf{h}_{p_i} = \mathbf{h}^{w}_{p_i} + \mathbf{h}^{p}_{p_i}. \qquad (3)$$

As we think different level information may have different importance for entity representations, then we give different weights for the two entity representations in sum

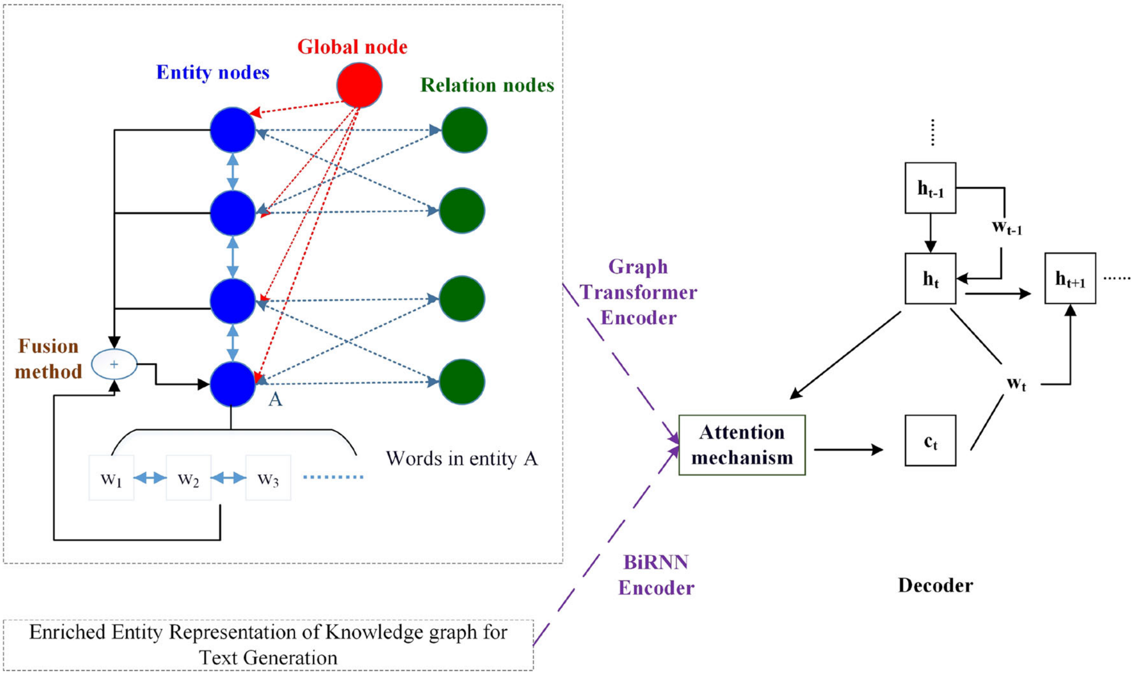

Fig. 1 Framework of MEFR model

operation, that is

$$\mathbf{h}_{p_i} = \alpha \mathbf{h}^{w}_{p_i} + (1-\alpha)\mathbf{h}^{p}_{p_i}, \tag{4}$$

where α is a weight balancing the word-level representations and phrase-level representations.

- **Selective mechanism**
  Inspired by highway networks with gating mechanism [33], which could fuse features by adopting two gating functions to scale and combine hidden states from two sources, and generate one representation, we develop a selective mechanism to dynamically control and indicate how much information are incorporated from the two-level entity representations, respectively. It can be illustrated as

$$\mathbf{s}_i = \sigma\left(\boldsymbol{\beta}_1 \mathbf{h}^{w}_{p_i} + \boldsymbol{\beta}_2 \mathbf{h}^{p}_{p_i} + c\right) \tag{5}$$

$$\mathbf{h}_{p_i} = \mathbf{s}_i \odot \mathbf{h}^{w}_{p_i} + (\mathbf{1} - \mathbf{s}_i) \odot \mathbf{h}^{p}_{p_i}, \tag{6}$$

where $\mathbf{s}_i$ is gate weight to control how much information incorporated from two levels, $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2$ are learnable parameters that model relations of parameters and c is the bias, σ denotes sigmoid function, and $\odot$ denotes element-wise multiplication.

To further validate the effectiveness of the selective mechanism, we also utilize two variants of it, which can be listed as

$$\mathbf{h}^{1}_{p_i} = \mathbf{s}_i \odot \mathbf{h}^{w}_{p_i} + \mathbf{h}^{p}_{p_i} \tag{7}$$

$$\mathbf{h}^{2}_{p_i} = \mathbf{h}^{w}_{p_i} + \mathbf{s}_i \odot \mathbf{h}^{p}_{p_i}. \tag{8}$$

Equations 7 and 8 represent removing selective mechanism of phrases and words, respectively.

Based on the above procedures, we obtain a d-dimensional representation of each node in the knowledge graph.

BiRNN encoder and graph transformer encoder

The input of the encoder is a knowledge graph and a corresponding title (if the graph contains a title). They are encoded by a Graph Transformer encoder [16] and a BiRNN encoder, respectively.

The title is also a short string, and we encode it using BiRNN to produce title embedding $\mathbf{T} = \mathrm{BiRNN}(\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_k)$, where $\mathbf{x}_i$ is i-th word embedding of the title.

We use Graph Transformer [16] to encode knowledge graph, which incorporates global structural information when contextualizing vertices in their local neighborhoods, and the resulting encodings are regarded as graph contextualized node encodings, i.e., $\mathbf{G} = \mathrm{GraphTransformer}(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n)$, $\mathbf{h}_i$ is i-th node embedding of the graph and n is the number of nodes in the graph.

Decoder

We adopt attention-based RNN [34] as the decoder of our model. At each decoding timestep t, we use the decoding hidden state $\mathbf{h}_t^{'}$ to calculate the context vectors $\mathbf{c}_k$ and $\mathbf{c}_r$ for knowledge graph and title, respectively. $\mathbf{c}_k$ is calculated by

$$\begin{aligned}
&\mathbf{c}_k = \mathbf{h}_t^{'} + \mathrm{Multihead}\left(\mathbf{h}_t^{'}, \mathbf{G}_j\right) \\
&\mathrm{Multihead}(\mathbf{Q}, \mathbf{K}) = \mathrm{concat}\left(\mathrm{head}_1, \ldots, \mathrm{head}_n\right) \\
&\mathrm{head}_i = \sum\nolimits_{j \in l} \mathit{Attention}\left(\mathbf{q_i}, \mathbf{k_j}\right) \mathbf{W}_G^n \mathbf{k}_j \\
&\mathrm{Attention}\left(\mathbf{q}_i, \mathbf{k}_j\right) = \frac{\exp\left(\left(\mathbf{W}_k \mathbf{k}_j\right)^{\mathrm{T}} \mathbf{W}_q \mathbf{q}_i\right)}{\sum\limits_{m \in l} \exp\left(\left(\mathbf{W}_k \mathbf{k}_m\right)^{\mathrm{T}} \mathbf{W}_q \mathbf{q}_i\right)} \cdot \frac{1}{\sqrt{d}},
\end{aligned} \tag{9}$$

where l denotes the neighborhood of the node q_i in graph, Attention() is the attention mechanism parameterized per head [16], $\mathbf{W}_G \in R^{d \times d}$ is a weight matrix, $(\mathbf{W}_q, \mathbf{W}_k) \in R^{d \times d}$ are learned independent transformations matrix of $\boldsymbol{q}$ and $\boldsymbol{k}$, respectively, $\frac{1}{\sqrt{d}}$ is a scaling factor to counteract the effect of gradient flow when dot products, $\mathrm{head}_1, \ldots, \mathrm{head}_n$ is n attention heads.

$\mathbf{c}_r$ is computed similarly using title encodings $\mathbf{T}$.

The final context vector $\mathbf{c}_t$ is obtained by concatenating $\mathbf{c}_k$ and $\mathbf{c}_r$, denoted as

$$\mathbf{c}_t = \mathrm{concat}\left(\mathbf{c}_k, \mathbf{c}_r\right). \tag{10}$$

Then, we use $\mathbf{c}_t$ and decoding state $\mathbf{h}_t^{'}$ as input for the next decoding step.

Copy mechanism

To enhance diversity of words and avoid out-of-vocabulary problem in generation, we compute a probability p_{gen} of copying from the input using $\mathbf{h}_t^{'}$ and $\mathbf{c}_t$ in a similar way with See et al. [32], as it allows copying words from vocabulary or knowledge graph. The probability $p_{\mathrm{gen}} \in [0, 1]$ for timestep t is calculated as

$$p_{\mathrm{gen}} = \sigma\left(\mathbf{W}_{\mathrm{copy}}\left[\mathbf{h}_t^{'} || \mathbf{c}_t\right] + b\right), \tag{11}$$

where $\mathbf{W}_{\mathrm{copy}}$ is a learnable parameter that transforms the concatenated vector, b is the bias, and σ is the sigmoid function.

Next, p_{gen} is used as a soft switch to choose selecting a word from the vocabulary by sampling from P_{vocab}, or copying entity from the input graph by sampling from the attention distribution P_{copy}. The probability distribution over the extended vocabulary, which is the union of the fixed vocabulary and input knowledge graph, is defined as

$$p_{\mathrm{gen}} * P_{\mathrm{copy}} + \left(1 - p_{\mathrm{gen}}\right) * P_{\mathrm{vocab}}, \tag{12}$$

where P_{copy} is calculated as $P_i^{\mathrm{copy}} = \mathrm{Attention}([\mathbf{h}_t^{'} || \mathbf{c}_t], \mathbf{x}_i)$ and $\mathbf{x}_i \in \mathbf{T} || \mathbf{G}$, P_{vocab} is computed by scaling $[\mathbf{h}_t^{'} || \mathbf{c}_t]$ to the vocabulary size and taking a softmax function.

Decoding algorithm

We use beam search algorithm during generation. As we found that there exists repetition problem in generated text, we develop a comparison mechanism based on vanilla beam search algorithm [19,20]. Our proposed comparison mechanism additionally calculates similarity between the word sequence adding current generating word and original word sequence, to update the score of word when beam search. The score of generating word is defined as

$$\mathrm{score}(y_t) = \delta \cdot \mathrm{score}(y_t) - (1 - \delta) \cdot \mathrm{comp}\left(s^* + y_t, s^*\right), \tag{13}$$

where y_t is the generating word at the timestep t, comp is the cosine similarity function calculating the similarity between two texts, s^* is the generated word sequence, and δ is the weighting factor. Based on the vanilla beam search, we add the second term in Eq. 13 to calculate the similarity between the sequences adding and without adding generating word, to punish the word which improves the similarity and reduce the redundancy.

Experiments

Dataset

We focus on generation task which generates corresponding text from knowledge graph in this paper. Therefore, we evaluate our model on two popular graph-to-text datasets: AGENDA [35] and WebNLG [36].

AGENDA (Abstract Generation Dataset), which consists of 40k paper titles and abstracts from Semantic Scholar Corpus taken from the proceedings of 12 top AI conferences. The average length of title and abstract are 9.9 words and 141.2 words, respectively. We follow the same procedure with Koncel-Kedziorski et al. [16] to create a knowledge graph for each abstract, and obtain a dataset of knowledge graphs paired with scientific abstracts. The average number of nodes and edges in the knowledge graph is 12.42 and 4.43, respectively. The dataset is split into a training/validation/test of 38720/1000/1000.We pre-process the dataset by replacing low-frequency words (words occurs fewer than 5 times) with <unk> tokens. In post-processing step, we delete repeated sentences and coordinated clauses.

WebNLG, which is also used for knowledge-graph-to-text generation task. Each instance in WebNLG contains a KG (knowledge graph) from DBPedia [37] and a corresponding text with one or several sentences describing the graph. The WebNLG dataset is split into 18,102, 872 and 971 instances for training, validation and test, respectively. Besides, graphs in AGENDA are automatically extracted, which leads to a high number of disconnected graph components. Compared with graphs in AGENDA, graphs in WebNLG are human-authored subgraphs of DBPedia. It means that the graph in WebNLG is more complete and more consistent with the content of corresponding target text. The relation types in WebNLG are 373, the average nodes are 34.9, and the average edges are 101. For WebNLG, we follow the previous work [36] to pre-process the knowledge graph. Besides, we refer [15] to deal with considerable number of edges and relations, avoiding parameter explosion, and create relation nodes to transform relational edges between entities which is similar to AGENDA.

Implementations

For AGENDA dataset, We employ LSTM [38] as Recurrent Neural Network, and apply a layer of bidirectional LSTM for title representation and each level of entity representations in the encoder–decoder framework, respectively. The dimension of the hidden vectors is set as 500. Models are trained for 20 epochs with early stopping [39] based on validation loss on an NVIDIA Tesla V100. Beam width is set as 4. The loss function is the negative log likelihood of generating text over the target text vocabulary and copied entity indices. Settings of SGD [40] optimization are applied to optimize the model parameters, and the related settings of Graph Transformer are set the same as in [16].

For WebNLG dataset, models are evaluated on the test set with seen categories. To implement our models, we employ two layers of bidirectional LSTM for each level of entity representations in encoder–decoder framework. We train our models with SGD optimizer for 100 epochs on WebNLG using an NVIDIA Tesla V100. The dimension of hidden encoder states is 256, and we train our models by minimizing negative log-likelihood loss function. The final results are generated by beam search, and beam width is set to 3.

Evaluation

We use BLEU [41] and ROUGE [42] as automatic evaluation metrics. Specifically, we use BLEU-n (n = 1, 2, 3, 4) in our experiments. And for ROUGE metric, we use ROUGE-1 and ROUGE-2 to assess informativeness, as well as ROUGE-L to assess fluency.

Parameter setting

In the first set of experiment, we examine and fix the values of parameters α in sum fusion mechanism and δ in comparison mechanism. We tune the values of α and δ from 0 to 1 with step size 0.1 when the model is trained.

Setting and analysis of parameter α in sum fusion mechanism

From Fig. 2, we can see that when $\alpha = 0$, the entity representation only contains phrase-level information. When increasing the value of α, the entity representation began to incorporate both word-level information and phrase-level information, which makes the model utilize rich entity information. The best ROUGE-2 score obtained at $\alpha = 0.8$ and we use it in the following experiments.

Setting and analysis of parameter δ in comparison mechanism

Then, we tune the parameter δ to obtain better performance of generation. We can see from Fig. 3 that when $\delta = 0$, the comparison mechanism is decided by comp function and quality of the generated text is not good enough. When changing δ value, the ROUGE-2 score changes fast at the beginning and gets best when $\delta = 0.4$, and then, it starts to decrease smoothly. when $\delta = 1$, the mechanism is based on vanilla beam search algorithm and the result is worse than the performance with $\delta = 0.4$. It illustrates that vanilla beam search algorithm could select important words, but it is restrained by redundancy. When we add comp function as second term to beam search algorithm, the comp function will consider similarity between the sequence with and without generating word to improve the quality of generated text. According to the result, the performance of the comparison mechanism is effectively improved when using a proper δ value. The opti-

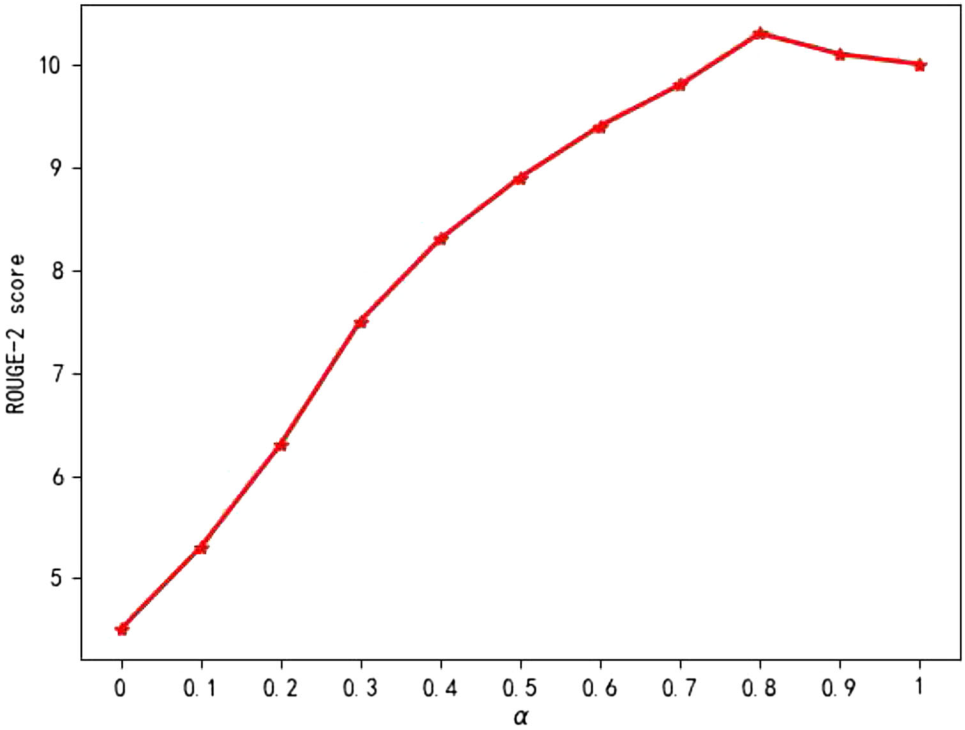

Fig. 2 ROUGE-2 scores vs. α on the test set

mal values of parameter α and δ on WebNLG dataset can be obtained in a similar way, and the best value is $\alpha = 0.6$, $\delta = 0.5$.

Ablation study

To explore effectiveness of MEFR model with different fusion mechanisms, we conduct experiments using different fusion mechanisms and their variants. For fair comparison, except the fusion methods, all the other processes involved remain the same.

Selective mechanism and its variants

Table 1 shows generation performance using selective mechanism and its two variants, i.e., Selective *w/o* p (removing selective mechanism of phrases) and Selective *w/o* w (removing selective mechanism of words). It indicates that the complete selective mechanism could incorporate both word-level and phrase-level information dynamically rather than just select information from one level. That is, information from the two levels can be fused through selective mechanism to jointly improve the generation performance.

Table 1 Results of selective mechanism and its variants on the test set

Methods	ROUGE-1	ROUGE-2	ROUGE-L
Selective *w/o* w	32.74	9.83	14.31
Selective *w/o* p	33.71	10.58	14.85
Selective mechanism	35.46	11.60	15.69

Comparison of different fusion mechanisms

Table 2 shows generation performance using different fusion mechanisms, including the sum fusion mechanism, i.e., direct sum (Sum_i) and weighted sum (Sum_e), as well as the selective fusion mechanism. The results show that though direct sum and weighted sum mechanisms could fuse information from the two levels, selective mechanism could better fuse word-level and phrase-level information dynamically, further enhancing generation performance. In the following experiments, we use selective mechanism as fusion mechanism of the model.

Table 2 Results of different fusion mechanism on the test set

Mechanisms	ROUGE-1	ROUGE-2	ROUGE-L
Sum_i	31.17	9.30	14.24
Sum_e	32.82	10.26	14.76
Selective mechanism	35.46	11.60	15.69

Comparison with other generation models

We first compare our proposed MEFR model with other generation models on AGENDA dataset:

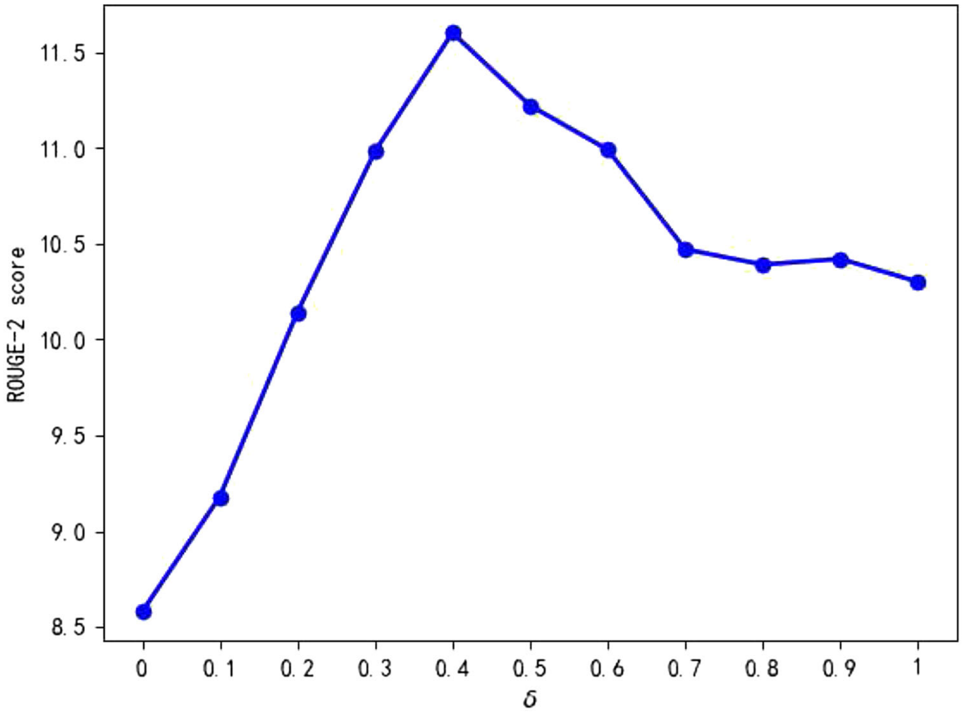

Fig. 3 ROUGE-2 scores vs. δ on the test set

(1) GAT [18], which is an Attention-Based Graph Neural Networks used for graph encoding.
(2) Graph Transformer [16], which encodes knowledge graph based on Transformer [17] and GAT [18].
(3) EntityWriter [16], which only uses entities and title for generation without considering graph relations.
(4) GCG [43], which is a graph convolutional networks-based model that explicitly considers the local node contexts within input structure.
(5) PGE [15], which is a fully parallel structure based on GAT for global and local node encoding.
(6) GT+RMA [44], which combines repulsive multi-head attention based on Graph Transformer [16] for text generation from knowledge graph.
(7) Graformer [45], which is an encoder–decoder architecture based on transformer used for graph-to-text generation.
(8) PGE-LW [15], which is a layer-wise parallel graph encoder based on GAT for node encoding.

Table 3 Results of different generation models on AGENDA test set

Methods	Bleu-1	Bleu-2	Bleu-3	Bleu-4
EntityWriter	24.12	10.27	4.38	3.09
GCG	26.12	12.91	6.54	3.62
GAT	26.51	13.19	6.79	3.71
Graph Transformer	29.08	14.93	8.57	4.62
PGE	32.95	18.18	11.14	6.94
GT+RMA	30.42	15.86	9.05	4.89
Graformer	33.10	18.28	11.22	7.00
PGE-LW	32.78	18.05	10.98	6.79
MEFR model	33.19	18.34	11.26	7.05

Table 3 shows performance of different generation models on the AGENDA dataset. EntityWriter performs poorest among these models; this can be due to it does not consider the graph relations. GAT and GCG could model the input graph structure and learn node representations, but they are still restrained by considering more semantic information and node relations. While Graph Transformer allows for a more global contextualization of each vertex through the use of a transformer-style architecture, further improving performance of knowledge-graph-to-text generation. However, it still misses a few entity information in the generated text according to the experiments. PGE improves the performance with the parallel structure based on GAT, which indicates the advantage of considering richer graph information. PGE-LW which combines the encoder in a layer-wise fashion does not improve performance compared with PGE. To strengthen model's expression ability, GT+RMA introduces repulsive multi-head attention based on Graph Transformer, but it does not bring significant improvement of the performance compared with Graph Transformer. Graformer achieves competitive performance using a novel graph self-attention based on Transformer for graph encoding, which can be used to detect global patterns. It also indicates the importance of effectively considering relations between nodes in knowledge graph for node representations. Different from the above models, we note that the repetitions and uncovered problem of entities are existed in the generated text. Our proposed model could effectively model the entity in knowledge graph from different granularities, which is able to extract more information and richer relations of entities, and make full use of the information in knowledge graph for representation learning. Our proposed MEFR model outperforms other baselines in terms of Bleu metrics. This could be attributed to that MEFR not only takes richer entity representations of the knowledge graph into account, but also introduces comparison mechanism to improve quality of the generated text.

Besides, to further validate the effectiveness of our proposed model, we compare our model with several representative generation models on WebNLG dataset, which is also used for graph-to-text generation and graphs contained in WebNLG are more complete compared to AGENDA. The models used for comparison are listed as follows:

(1) UPF-FORGe [36]: a rule-based method which mostly focuses on using predicate–argument templates during sentence planning.
(2) Adapt [36]: a neural encoder–decoder based framework with utilizing sub-word representations and linearizing the input sequence.
(3) Melbourne [36]: which combines delexicalization and enrichment of the input sequence with attentional encoder–decoder model.
(4) Graph Conv [43]: which is a graph convolutional network-based encoder directly utilizing the input graph structure.
(5) E2EGRU [46]: which takes end-to-end architecture based on GRU for data-to-text generation without explicit intermediate representations.
(6) GTR-LSTM [47]: which is a sentence generation model with the novel graph-based triple encoder.
(7) SBS [48]: which proposes to split generation procedure into a symbolic text-planning stage and a neural generation stage.

We also use Graformer [45] as a comparison model.

Like the models we compare with, we report Bleu scores rather than Bleu-n on WebNLG, and the results for comparison are taken from their corresponding paper or obtained by running publicly released source code. The results are shown in Table 4.

Table 4 shows the results of different generation models on WebNLG test set with seen categories. The first three models are advanced competitors in WebNLG challenge with seen categories. Among them, we can see Adapt and Melbourne which are based on attentional encoder–decoder show greater performance; it indicates the advantages of neural network-based models compared with rule-based models. For the fourth to seventh models, Graph Conv directly utilizes the input graph structure with graph convolutional network-based encoder. E2EGRU uses an end-to-end data-to-text model based on GRU, to generate text without explicit intermediate representations. GTR-LSTM proposes a novel graph-based triple encoder to preserve more information from original data for data-to-text generation. And SBS further splits generation procedure into two stages for generating high-quality text. These models achieve good performance and show benefits of explicitly encoding the input graph structure. However, they are still restrained by effectively utilizing semantic information and node relations of the input graph. Transformer-based Graformer shows great performance which learns node representations not only relying on their neighbors, but also focusing on global patterns based on the novel graph attention. It indicates the advantages of effectively considering relations of nodes in the knowledge graph. Compared with Graformer, our proposed model could learn interactions of nodes with global patterns based on Graph Transformer. Besides, we especially focus on modeling relations of entities, and learning their representations from aggregating different granularity information to generate high-quality text. Our proposed model achieves best performance among the baselines, which proves that our model could obtain richer information of knowledge graph for entity representations, and utilize comparison mechanism to help improve quality of the generated text. Besides, graphs in WebNLG are more complete compared with AGENDA, which means that richer semantic information of entities is contained in the graph. And it can be effectively utilized by our model to enhance the performance. Moreover, our model could outperform other baselines without pre-trained language models, which also indicates the importance of further exploring the information contained in knowledge graph.

Table 4 Results of different generation models on WebNLG test set with seen categories

Methods	BLEU
UPF-FORGe	40.88
Adapt	60.59
Melbourne	54.52
Graph Conv	55.90
E2EGRU	57.20
GTR-LSTM	58.60
SBS	53.30
Graformer	61.15
MEFR model	62.06

Human evaluation and case study

We perform human evaluations to establish that the Bleu improvements of our proposed MEFR model are correlated

Table 5 Human evaluation results

Methods	Informativeness	Fluency	Redundancy
GAT	3.77	4.09	4.11
Graph Transformer	3.96	4.26	4.21
MEFR model	4.24	4.32	4.29

Table 6 Examples of generated texts

Title: “Towards Internet-scale multi-view stereo.”
Gold: this paper introduces an approach for enabling existing **multi-view stereo methods** to operate on extremely large **unstructured photo collections**. the main idea is to decompose the collection into a set of overlapping sets of photos that can be processed in parallel, and to merge the resulting reconstructions. this **overlapping clustering problem** is formulated as a **constrained optimization** and solved iteratively. the **merging algorithm**, designed to be parallel and out-of-core, incorporates robust **filtering steps** to eliminate **low-quality reconstructions** and enforce **global visibility constraints**. the approach has been tested on several large datasets downloaded from **flickr.com**, including one with over ten thousand images, yielding a **3d reconstruction**…
Graph Transformer: in this paper, we propose a new method for *3d reconstruction* in *unstructured photo collections*. in the proposed method, a *merging algorithm* is used to solve the *overlapping clustering problem*. the method is applied to the problem of 3d reconstruction in *flickr.com*…
GAT: in this paper, we present a new approach for *3d reconstruction* in *unstructured photo collections*. and *merging algorithm* can be used for *overlapping clustering problem* ***with merging algorithm***. the presented approach has been used for 3d reconstruction in *flickr.com*…
MEFR model: in this paper, we address the problem of *3d reconstruction* in *unstructured photo collections* . we propose a novel *merging algorithm* based on *constrained optimization* of the *overlapping clustering problem*. unlike *multi-view stereo methods*, we are able to 3d reconstruction in *flickr.com* with *global visibility constraints*. furthermore, we show that global visibility constraints can be very useful for overlapping clustering problem in flickr.com…

The bold words are entity information contained in the Gold. The italic words in generated texts are included entities. The bold and italicized words are repeated parts

with human judgments. We randomly select 40 samples from test set and compare the text generated by our method with the text generated by GAT and Graph Transformer. We ask three volunteers to rate these samples on a scale of 5 (very good) to 1 (very poor), in terms of informativeness, fluency, and redundancy of each text. The three volunteers are specialists (including a professor, two associate professors) from School of International Studies, Shaanxi Normal University. The average results are listed in Table 5. Informativeness represents that the generated text should include rich information, fluency represents that sentences in the text should be expressed fluently and logically, and redundancy represents that the text should contain few repeated information.

Table 5 shows that our proposed MEFR model outperforms the other two models on three aspects, especially in informativeness. Compared with Graph Transformer, the text generated by MEFR is more informativeness, indicating the advantages of fusion methods.

Besides, we show an example of generated text by the three models in Table 6. Compared with GAT, Graph Transformer could generate a more fluent and informative text with the help of global contextualization. It is not surprising to find that our proposed MEFR model gets best scores of informativeness obviously, and the generated text contains more details description as well as entity information, which makes the text more complete and readable compared with the other methods. It indicates that by integrating information from different levels of entity, our proposed MEFR model could generate text containing more information, and better utilize the information of knowledge graph to produce rich description which is different from the textual expressions produced by other two models.

Conclusion and future work

In this paper, we focus on knowledge-graph-to-text generation task which generates corresponding descriptive text from knowledge graph. However, the generated text often suffers from problems, such as redundancy and not fully utilizing entity information, which leads low quality of the generated text. Therefore, we propose an MEFR model to solve the above issues, aiming to generate the text with rich description (covering information contained in knowledge graph as much as possible) and low redundancy (containing less repetitive information). Our proposed MEFR model effectively incorporates information from different levels for obtaining entity representations in knowledge graph. Besides, the proposed comparison mechanism in decoding procedure is used to reduce redundancy of the generated text based on similarity. According to the results on the two popular graph-to-text generation datasets, our proposed model could achieve advanced performance and improve quality of the generated text. At the same time, our model which does not use pre-trained language models shows great performance compared with other generation models. It also means the importance of further exploring information contained in the knowledge graph. And for our proposed model which combines multi-granularity information can make more effective use of the original input for representation.

In the future, we will continue exploring how to better utilize information from different granularities in complex networks, to further improve the performance of text generation. Besides, pre-trained language models show great performance on natural language generation, and we will explore to enrich node representations in knowledge graph with pre-trained language models for generation. In addition to improving performance of the generation model, the dataset used for knowledge-graph-to-text generation is still worth to focus on. And we will try to make the dataset of knowledge graphs paired with texts in specific fields, to further study the effect of fusion representation in graph-to-text generation.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Nos. 61872296, 61772429, U20B2065), MOE (Ministry of Education in China) Project of Humanities and Social Sciences (18YJC870001), and Fundamental Research Funds for the Central Universities (3102019DHKY04).

Declarations We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

1. Reiter E, Dale R (2000) Building natural language generation systems, studies in natural language processing. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511519857
2. Hu Y, Wan X (2014) Automatic generation of related work sections in scientific papers: an optimization approach. In: Proceedings of the 2014 Conference on empirical methods in natural language processing (EMNLP), Association for Computational Linguistics, pp 1624–1633. https://doi.org/10.3115/v1/D14-1170
3. Mei H, Bansal M, Walter MR (2016) What to talk about and how? Selective generation using lstms with coarse-to-fine alignment. arXiv:1509.00838
4. Vinyals O, Toshev A, Bengio S, Erhan D (2015) Show and tell: a neural image caption generator. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE Computer Society, pp 3156–3164
5. Iqbal T, Qureshi S (2020) The survey: text generation models in deep learning. J King Saud Univ Comput Inf Sci. https://doi.org/10.1016/j.jksuci.2020.04.001
6. Nie F, Wang J, Yao J-G, Pan R, Lin C-Y (2018) Operation-guided neural networks for high fidelity data-to-text generation. In: Proceedings of the 2018 Conference on empirical methods in natural language processing, association for computational linguistics, pp 3879–3889. https://doi.org/10.18653/v1/D18-1422
7. Wiseman S, Shieber S, Rush A (2018) Learning neural templates for text generation. In: Proceedings of the 2018 Conference on empirical methods in natural language processing, Association for Computational Linguistics, pp 3174–3187. https://doi.org/10.18653/v1/D18-1356
8. Li L, Wan X (2018) Point precisely: towards ensuring the precision of data in generated texts using delayed copy mechanism. In: Proceedings of the 27th International Conference on computational linguistics, Association for Computational Linguistics, pp 1044–1055
9. Puduppully R, Dong L, Lapata M (2019) Data-to-text generation with content selection and planning. In: Proceedings of the 33rd AAAI Conference on artificial intelligence, pp 6908–6915. https://doi.org/10.1609/aaai.v33i01.33016908
10. Mohan A, Pramod KV (2021) Temporal network embedding using graph attention network. Complex Intell Syst 15:10. https://doi.org/10.1007/s40747-021-00332-x
11. Huang Z, Xie Z (2021) A patent keywords extraction method using textrank model with prior public knowledge. Complex Intell Syst 15:10. https://doi.org/10.1007/s40747-021-00343-8
12. Xu K, Wu L, Wang Z, Feng Y, Witbrock M, Sheinin V (2018) Graph2seq: graph to sequence learning with attention-based neural networks. arXiv:1804.00823
13. Beck D, Haffari G, Cohn T (2018) Graph-to-sequence learning using gated graph neural networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, pp 273–283. https://doi.org/10.18653/v1/P18-1026
14. Li W, Xu J, He Y, Yan S, Wu Y, Sun X (2019) Coherent comments generation for Chinese articles with a graph-to-sequence model. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp 4843–4852. https://doi.org/10.18653/v1/P19-1479
15. Ribeiro LF, Zhang Y, Gardent C, Gurevych I (2020) Modeling global and local node contexts for text generation from knowledge graphs. Trans Assoc Comput Linguist 8:589–604
16. Koncel-Kedziorski R, Bekal D, Luan Y, Lapata M, Hajishirzi H (2019) Text Generation from Knowledge Graphs with Graph Transformers, In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, pp 2284–2293. https://doi.org/10.18653/v1/N19-1238
17. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Lu, Polosukhin I (2017) Attention is all you need. In: Proceedings of the 31st international conference on neural information processing systems, vol 30. Curran Associates, Inc, pp 6000–6010
18. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y (2018) Graph attention networks. arXiv:1710.10903
19. Graves A (2012) Sequence transduction with recurrent neural networks. arXiv:1211.3711
20. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Proceedings of the 27th International Conference on Neural information processing systems 2:3104–3112
21. Angeli G, Liang P, Klein D (2010) A simple domain-independent probabilistic approach to generation. In: Proceedings of the 2010 Conference on empirical methods in natural language processing, Association for Computational Linguistics, pp 502–512
22. Kondadadi R, Howald B, Schilder F (2013) A statistical NLG framework for aggregated planning and realization. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, pp 1406–1415
23. Howald B, Kondadadi R, Schilder F (2013) Domain adaptable semantic clustering in statistical NLG. In: Proceedings of the 10th International Conference on computational semantics (IWCS 2013)—Long Papers, Association for Computational Linguistics, pp 143–154
24. Juraska J, Karagiannis P, Bowden K, Walker M (2018) A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1

(Long Papers), Association for Computational Linguistics, pp 152–162. https://doi.org/10.18653/v1/N18-1014

25. Gehrmann S, Dai F, Elder H, Rush A (2018) End-to-end content and plan selection for data-to-text generation. In: Proceedings of the 11th International Conference on natural language generation, Association for Computational Linguistics pp 46–56. https://doi.org/10.18653/v1/W18-6505
26. Freitag M, Roy S (2018) Unsupervised natural language generation with denoising autoencoders. In: Proceedings of the 2018 Conference on empirical methods in natural language processing, Association for Computational Linguistics, pp 3922–3929. https://doi.org/10.18653/v1/D18-1426
27. Xu K, Wu L, Wang Z, Feng Y, Sheinin V (2018) SQL-to-text generation with graph-to-sequence model. In: Proceedings of the 2018 Conference on empirical methods in natural language processing, Association for Computational Linguistics, pp 931–936. https://doi.org/10.18653/v1/D18-1112
28. Song L, Wang A, Su J, Zhang Y, Xu K, Ge Y, Yu D (2018) Structural information preserving for graph-to-text generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp 7987–7998. https://doi.org/10.18653/v1/2020.acl-main.712
29. Guo Z, Zhang Y, Teng Z, Lu W (2019) Densely connected graph convolutional networks for graph-to-sequence learning. arXiv:1908.05957
30. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45(11):2673–2681. https://doi.org/10.1109/78.650093
31. Luong T, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on empirical methods in natural language processing, Association for Computational Linguistics, pp 1412–1421. https://doi.org/10.18653/v1/D15-1166
32. See A, Liu PJ, Manning CD (2017) Get to the point: Summarization with pointer-generator networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, pp 1073–1083. https://doi.org/10.18653/v1/P17-1099
33. Srivastava RK, Greff K, Schmidhuber J (2015) Highway networks. arXiv:1505.00387
34. Luong M-T, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on empirical methods in natural language processing, pp 1412–1421
35. Ammar W, Groeneveld D, Bhagavatula C, Beltagy I, Crawford M, Downey D, Dunkelberger J, Elgohary A, Feldman S, Ha V, Kinney R, Kohlmeier S, Lo K, Murray T, Ooi H-H, Peters M, Power J, Skjonsberg S, Wang L, Wilhelm C, Yuan Z, van Zuylen M, Etzioni O (2018) Construction of the literature graph in semantic scholar. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), Association for Computational Linguistics, pp 84–91. https://doi.org/10.18653/v1/N18-3011
36. Gardent C, Shimorina A, Narayan S, Perez-Beltrachini L (2017) The webnlg challenge: Generating text from rdf data. In: Proceedings of the 10th International Conference on natural language generation, pp 124–133
37. Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z (2007) Dbpedia: a nucleus for a web of open data. In: Proceedings of the 6th international the semantic web and 2nd Asian conference on Asian semantic web conference. Springer, pp 722–73
38. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput. https://doi.org/10.1162/neco.1997.9.8.1735
39. Prechelt L (1998) Early stopping-but when? In: Neural networks: tricks of the trade, this book is an Outgrowth of a 1996 NIPS Workshop, Springer-Verlag, pp 55–69
40. Qian N (1999) On the momentum term in gradient descent learning algorithms. Neural Netw. https://doi.org/10.1016/S0893-6080(98)00116-6
41. Papineni K, Roukos S, Ward T, 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, Association for Computational Linguistics, pp 311–318. https://doi.org/10.3115/1073083.1073135
42. Lin C (2004) Rouge: a package for automatic evaluation of summaries. In: Proceedings of workshop on text summarization branches out, post conference workshop of ACL 2004. Association for Computational Linguistics, pp 74–81
43. Marcheggiani D, Perez-Beltrachini L (2018) Deep graph convolutional encoders for structured data to text generation. arXiv:1810.09995
44. An B, Dong X, Chen C (2019) Repulsive Bayesian sampling for diversified attention modeling. In: 4th workshop on Bayesian deep learning (NeurIPS 2019), pp 1–10
45. Schmitt M, Ribeiro LF, Dufter P, Gurevych I, Schütze H (2020) Modeling graph structure via relative position for text generation from knowledge graphs. arXiv preprint arXiv:2006.09242
46. Ferreira TC, van der Lee C, Van Miltenburg E, Krahmer E (2019) Neural data-to-text generation: a comparison between pipeline and end-to-end architectures. In: Proceedings of the 2019 Conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), vol 1, pp 552–562
47. Distiawan B, Qi J, Zhang R, Wang W (2018) Gtr-lstm: a triple encoder for sentence generation from rdf data. In: Proceedings of the 56th annual meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 1627–1637
48. Moryossef A, Goldberg Y, Dagan I (2019) Step-by-step: separating planning from realization in neural data-to-text generation. In: Proceedings of the 2019 conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp 2267–2277