Building the Directed Semantic Graph for Coherent Long Text Generation

Ziao Wang , Xiaofeng Zhang*, Hongwei Du
Harbin Institute of Technology, Shenzhen, China
20b351002@stu.hit.edu.cn, {zhangxiaofeng, hwdu}@hit.edu.cn

Abstract

Generating long text conditionally depending on the short input text has recently attracted more and more research efforts. Most existing approaches focus more on introducing extra knowledge to supplement the short input text, but ignore the coherence issue of the generated texts. To address aforementioned research issue, this paper proposes a novel two-stage approach to generate coherent long text. Particularly, we first build a document-level path for each output text with each sentence embedding as its node, and a revised self-organising map (SOM) is proposed to cluster similar nodes of a family of document-level paths to construct the directed semantic graph. Then, three subgraph alignment methods are proposed to extract the maximum matching paths or subgraphs. These directed subgraphs are considered to well preserve extra but relevant content to the short input text, and then they are decoded by the employed pre-trained model to generate coherent long text. Extensive experiments have been performed on three real-world datasets, and the promising results demonstrate that the proposed approach is superior to the state-of-the-art approaches w.r.t. a number of evaluation criteria.

1 Introduction

Recently, in the domain of natural language generation (NLG), the conditional long text generation (Shen et al., 2019) has been investigated with flourishing results (Wang et al., 2018; Tan et al., 2020; Hu et al., 2021; Ren et al., 2020). Essentially, this task aims to generate text in line with the main semantic meanings of the input text; and meanwhile, it requires that the length of the generated text is significantly longer than that of the short input text. The key to this challenging task is how to generate extra but relevant content that is well positioned to express the semantic meanings of the input text. To generate extra content, the pointer-generator network (See et al., 2017) is proposed to copy terms from an external corpus. Alternatively, various deep generative models (Bowman et al., 2016) have been proposed to generate previously unseen content by sampling from the learnt distribution of the latent variables and there also exist some other research attempts (Hu et al., 2021).

Although the aforementioned issues were partially alleviated by existing approaches, they obviously ignored the logic coherency issue between the generated texts. To simplify the problem, hereinafter, the logic coherency refers to the generated paragraphs, with each generated to represent the same semantic meanings, and to have the correct sequence order. Apparently, it is a non-trivial task to generate extra content having appropriate sequence order. First, it still remains an open challenge in the literature to generate extra content given the short input text, especially for some domain-specific applications, e.g., financial report generation (Ren et al., 2020), where there usually exist multiple underlying semantic meanings for each piece of short news. Second, even if the underlying semantic content could be well generated, how to properly determine the logical order of the generated texts is another challenging issue.

This paper is thus motivated to address aforementioned two research issues, i.e., logic coherency and extra content generation. To achieve this goal, we propose this two-stage approach which first constructs a directed semantic graph, and then generates coherent long text via a pre-trained model with the maximum matching subgraphs, extracted from the constructed semantic graph, as its input. Particularly, we first build a document-level path for each long text with each sentence embedding as its node and the sentence order as its directed edge. A revised self-organising map (SOM) is proposed to merge similar nodes belonging to different document-level paths into one to represent a coarse

*Corresponding author.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2563–2572
November 7–11, 2021.

level semantic meaning. The intuitive design behind this is that human writers may have their own writing styles, their sentences might be slightly different, and thus, by merging similar nodes the underlying semantic meanings could be acquired. After SOM merging step, the document level graphs are interwove with each other and thus a corpus-level graph is acquired which is called the directed semantic graph. For the text generation stage, each input text is matched with the top-K similar input texts (training data) and their output texts are then constructed into paths or subgraphs. These paths or subgraphs are aligned with the directed semantic graph, and the embeddings of the maximum matching paths or subgraphs are fed into the employed pre-trained model for text generation.

The major contributions of this paper are summarized as follows.

- To the best of our knowledge, this is the first attempt to address the logic coherency issue for the long text generation task.
- A hybrid method was proposed to build the directed semantic graph which could well capture the rich semantic meanings of paragraphs as well as their logic orders.
- Extensive experiments were conducted on three real-world datasets, which demonstrates the superior performance of our proposed approach over the state-of-the-art methods.

2 Related Work

Recently, long text generation (Shen et al., 2019) has received significant research attention which focuses more on generating previously unseen but relevant content (Hu et al., 2021; Wang et al., 2018). The existing approaches could be roughly categorized into three groups, e.g., *sequence-to-sequence based approaches* (Cho et al., 2014; Bahdanau et al., 2015), *generative model based approaches* (Bowman et al., 2016; tra) and *GAN based approaches* (Kusner and Hernández-Lobato, 2016; Yu et al., 2017). Recently, various pre-trained models (Song et al., 2019; Tan et al., 2020; Dong et al., 2019) have been proposed. For instance, (Song et al., 2019) employs a pre-trained module as the decoder to summarize text; while the GPT-based approaches (Tan et al., 2020) further improve the readability of the generated text. To generate relevant content, (Wang et al., 2019) proposed a variational autoencoder based approach which generates sentences sampled from the underlying topics learnt from the input data. Similarly, (Shen et al., 2019) proposed a multi-level variational autoencoder based approach where both high-level document features and low-level word features are respectively extracted and aggregated for the text generation. To learn external domain-specific knowledge, the CVAE-KD (Ren et al., 2020) proposed a knowledge distillation method that employs the teacher-student architecture. The teacher component is to learn external knowledge from the extracted sub set of financial reports and then guides the student component to generate relevant text w.r.t. the input news.

Notably, most existing approaches ignore the logic coherency issue during the process of the text generation. Alternative to the writing logic, (Yao et al., 2019) utilized a storyline to generate stories given a title. Similarly, (Hu et al., 2021) proposed the news-to-report generation task where the authors first generate the outline for the input news and then generate the financial reports guided by the outline. However, both the storyline and outline are essentially defined as multiple semantic topics for the text generator to generate more elaborating content, whereas the writing logic issue, raised in this paper, constrains the relationship among the generated texts. Similar to storyline, there also exist a number of document planning-based (Thomson et al., 2018; Shao et al., 2019; Hua and Wang, 2019; Kang and Hovy, 2020) text generation approaches. These approaches either pre-define a document plan for well-structured text (Biran and McKeown, 2015) or dynamically learn such plan by extracting a set of keywords (Thomson et al., 2018) from the training data. Although the extracted concepts or keywords (Fan et al., 2019) could well preserve word level contextual information, but apparently they ignore the underlying semantic coherence between the generated sentences or paragraphs whereas the proposed approach aims to capture such semantic coherence.

3 Preliminaries

The corpus set consists of (X, Y) pairs where $X = \{x_1, x_2, ..., x_n\}$ denotes the input text set, $Y = \{y_1, y_2, ..., y_n\}$ denotes the corresponding output long text set, and y_j is used to construct the directed semantic graph. Let $y_i = \{s_k^1, s_k^2, ...s_k^p\}$ with $e(s_k^p) = (a_1, a_2, ..., a_n)$ denoting the embed-

ding of a sentence s_k^p, where s_k^p denotes the $p-th$ sentence of the $k-th$ document. A document-level path is to be constructed for each input or output text with a sentence s_k^p as its node, and (s_k^i, s_k^j) is the directed edge between two nodes. Thus, a document-level path is defined as $p_i = (s_k^n, s_k^m)^p$ which starts from vertex s_k^n and ends at vertex s_k^m allowing repeated vertices. The directed semantic graph $SG = (V, E)$ is to be constructed with the centroid of a group of similar path nodes $\{s_i^j\}$ as its graph node, and the majority direction of the contained edges as its directed edge between two graph nodes. More details will be presented in Section 4.1.

4 The Proposed Approach

The proposed approach consists of two stages: (1) the directed semantic graph construction; (2) maximum subgraph matching for text generation. Details of each stage are illustrated in the following subsections.

4.1 Directed Semantic Graph Construction

In this subsection, a document-level path (or graph) was built for each input long text with each sentence embedding as its node and the sentence order as its directed edge. Then, we globally merge similar nodes into one via the revised self-organizing map (SOM) to represent a coarse level semantic meaning. After SOM merging step, document-level paths are interwoven with each other and thus a corpus-level graph is acquired which is called the directed semantic graph in this paper. Details of each step are illustrated as follows.

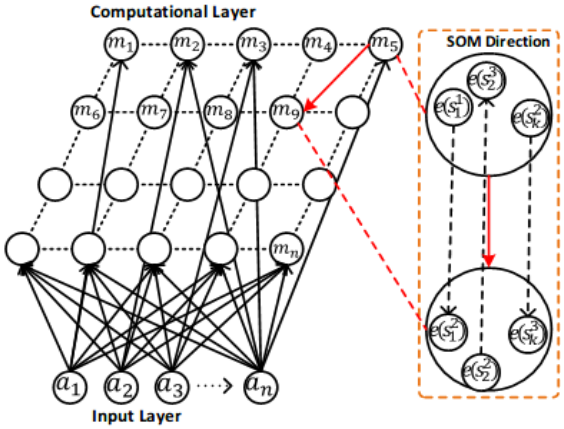
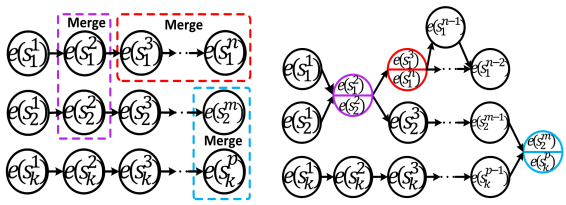


Figure 1: Merging similar nodes via the revised SOM.



(a) Before merging nodes. (b) After merging nodes.

Figure 2: The directed semantic graph construction process.

4.1.1 Building Document-level Path

To build a document-level path for each input document, sentences from each document (long text) of the corpus set were extracted and its embedding was treated as the node. To generate the sentence embedding, a pre-trained BERT encoder was employed and the sentence embeddings $e(s_k^p)$ are acquired by the 'CLS' token embedding (Devlin et al., 2019). The sentence order was used to generate the directed edge between two adjacent sentence nodes. Thus, we have K paths $P = \{p_1, p_2, ..., p_k\}$ if the corpus set contains K pairs of short-long text. These document-level paths will be used to construct the directed semantic graph in the following subsections.

4.1.2 Merging Sentence Node via SOM

After acquiring all local document-level paths $P = \{p_1, p_2, ..., p_k\}$, we further merge similar sentence nodes into a single one and then globally build the directed semantic graph denoted as SG. To merge similar sentences, the self-organizing map (SOM) is adapted and its working process is illustrated in Figure 1. The revised SOM consists of two layers, i.e., the input layer and the computation layer. The input vector is given as $a = [a_1, a_2, ..., a_n]^T \in R^n$ which connects all the neurons in the computation layer. Each neuron in the computation layer has a weight vector denoted as $m_i = [m_{i1}, m_{i2}, ..., m_{in}]^T \in R^n$. The matching criterion of SOM is the minimum distance between input vector a and the neuron weight vector m_i which is calculated as

$$m_c = min_{i=1}^{n}\{\sum_{j=1}^{n}(a_j - m_{ij})^2\},$$

where neuron c is the Best Matching Unit (BMU) which is the nearest to current input vector a among all the neurons. Straightforwardly, the neighbor-

hood set N_c of cell c is updated as

$$m_i(t+1) = m_i(t) + h_{ci}(t)[a(t) - m_i(t)],$$

and h_{ci} is computed as

$$h_{ci} = h_0(t) \exp^{-||r_i - r_c||^2/\sigma(t)^2},$$

where r_i and r_c respectively denote the coordinates of cell i and c, $\sigma(t)$ is a coefficient decaying factor which is generally calculated as

$$\sigma(t) = \sigma_0 \exp^{-t/\tau_\sigma},$$

where τ_σ is a hyper-parameter to control the decaying speed of $\sigma(t)$.

4.1.3 Building Directed Edges

To recall that we have a number of independent paths $P = \{p_1, p_2, ..., p_k\}$ and these paths may interweave with each other after merging similar nodes. Note that after merging nodes, each SOM neuron contains several sentence nodes and thus for any two neurons, there exist two sets of sentence nodes. For example in Figure 1, we have $\{e(s_1^1), e(s_2^3), e(s_k^2)\}$ and $\{e(s_1^2), e(s_2^2), e(s_k^3)\}$, and there might exist directed edges between these two sets of sentence nodes. Assuming we have three directed edges $e(s_1^1) \rightarrow e(s_1^2)$, $e(s_2^3) \leftarrow e(s_2^2)$ and $e(s_k^2) \rightarrow e(s_k^3)$. Then the direction between two neurons m_i and m_j is simply determined by the direction of the majority nodes belonging to them, i.e. $m_5 \rightarrow m_9$ in this case.

After building directed edges between SOM neurons, the directed semantic graph is acquired as plotted in Figure 2. With this graph, we could effectively expand extra content, i.e., more paths or implicitly associated path nodes, to supplement the short input text, and meanwhile the coherence is guaranteed by the sequence order, i.e. the directed edge. For example, before merging path nodes, given starting node $e(s_1^1)$, the path can only lead us to as far as $e(s_1^n)$ where the semantic meaning is restricted to this path heavily as shown in Figure 2 (a). While after merging similar nodes, we have two paths starting from $e(s_1^1)$ and ending at node $e(s_1^{n-1})$ and $\frac{e(s_2^m)}{e(s_k^p)}$, respectively. Apparently, in this directed semantic graph, it is possible to generate more paths for the same input text and each extracted path contains more path nodes. Thus, the directed semantic graph is able to effectively generate extra but coherent content to express the semantic meanings of the short input text.

To further differentiate the importance of these directed edges, we have the following intuitive assumptions. If an edge is repeated many times, i.e., two sentences are frequently written together, they are more possible to be generated together. Therefore, the importance of each directed edge is calculated as

$$\omega = \sum_{\forall sg \in SG} |(u, v) \in sg|,$$

where s denotes sentences in output text set Y, sg denotes the matching subgraph in SG.

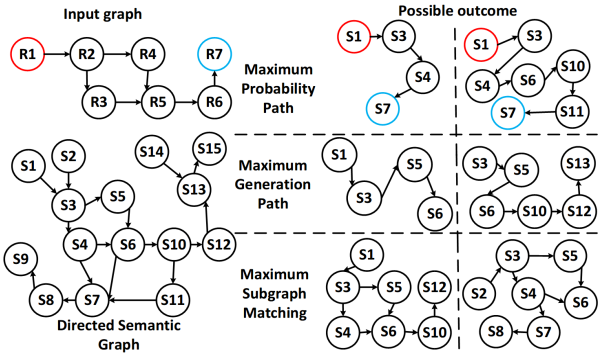


Figure 3: The proposed three kinds of matching methods.

4.2 Text Generation with Maximum Subgraph Matching

For text generation, a pre-trained model is employed to generate coherent text. To this end, we retrieve the top-K similar output texts from the training set for each input text. Then, these retrieved output texts are wrapped into paths or subgraphs using the method proposed in Section 4.1. The generated paths or sugraphs are aligned with the directed semantic graph to find the best matched subgraphs. Finally, these best matched subgraphs we believed they contain sufficient external semantic information will be fed into a pre-trained model for text-generation. For the alignment, we respectively propose three kinds of alignment methods, i.e., maximum probability path, maximum generation path and maximum subgraph matching, illustrated in the following subsections.

4.2.1 Maximum Probability Path

To align a path with the directed semantic graph SG, we first align the starting node and the ending node with graph nodes in SG where we simply

align nodes with the same SOM neuron representing similar semantic meaning. Let v_1 and v_n denote the matched starting and ending node in the directed semantic graph, respectively. Assuming there are m paths starting from v_1 to v_n, the best-matched path among m paths is calculated as

$$P(s) = \max_{j \in m} \{ \prod_{t-1}^{T} p_j(v_t | v_1, ..., v_{t-1}) \}. \quad (1)$$

This equation is to find the path having the highest joint probability. To calculate this equation, we fix the starting node and the ending node and enumerate the rest nodes residing in a path to find a sequence of nodes with the highest probability, the process is illustrated in Figure 3. This method simulates the situation how to organize the most appropriate content as well as their sequence order if the human writer wants to elaborate two given knowledge points.

4.2.2 Maximum Generation Path

Given a path $p_1 = (v_1, v_n)^{p_1}$, this method is to find a path $p_2 = (u_1, u_n)^{p_2}$ from SG that is more possible to be generated by p_1. Different from the maximum probability path, the starting and ending node are not fixed. This alignment problem is to calculate the probability that p_2 is generated by p_1, computed as

$$\begin{aligned} & P(u_1, ...u_n | v_1, ..., v_n) \\ = & \prod_{t=1}^{T} p(u_t | v_1, ...v_{t-1}, u_1, ..., u_{t-1}). \end{aligned} \quad (2)$$

The merit of this method lies in that we could generate a more flexible long text (no fixed starting and ending nodes) and the whole semantic meaning of the generated long text is optimized to be close enough to the input text, although its computation cost is a bit higher than that of the first method.

4.2.3 Maximum Subgraph Matching

If the input is a graph H, the target of maximum subgraph matching is to find the best matched subgraph $H^{'}$. For this purpose, the graph kernel method is designed where the kernel function is to measure the similarity between two subgraphs. We adopt the shortest path kernel to measure the similarities and the kernel function $k_{SP}(H, H^{'})$ is defined as,

$$\sum_{(u,v) \in V(H)^2} \sum_{(w,z) \in V(H')^2} k((u,v),(w,z)) \quad u \neq v \ , \ w \neq z$$

where distance measurement $k((u,v),(w,z))$ is calculated as,

$$k_L(l(u), l(w)) \cdot k_L(l(v), l(z)) \cdot k_D(d(u,v), d(w,z)),$$

where $d(u,v)$ is to calculate the shortest path between u and v, k_L is to measure whether two graph nodes are close enough or not and we calculate $l()$ using node embeddings, k_D is to determine whether the two shortest paths are close enough or not, and we have $k_D(d(u,v), d(w,z)) = 0$ if $d(u,v) = \infty$ or $d(w,z) = \infty$. The threshold ϕ is empirically fine-tuned in the experiment to align two subgraphs H and $H^{'}$, calculated as $k_{SP} \geq \phi$. Alternatively, this method relaxes the restrictions of the previous two methods by allowing the input could be a graph.

4.2.4 Text Generation

After acquiring the aligned paths or subgraphs for each input, these aligned results are fed into a pre-trained model for text-generation. The employed pre-trained model UNILIM (Dong et al., 2019) is fine-tuned by minimizing the loss between the generated output text and the ground-truth output text. For the testing stage, the pre-processing steps are similar to those of the training stage. Differently, the aligned paths or subgraphs for each testing input are directly decoded using the fine-tuned pre-trained model.

dataset	news_report	arXiv	ACL
#train	109,663	40,141	5,192
#valid	10,427	3,824	437
#test	10,427	3,824	437
avg len of target text	341	267	261
avg len of input text	28	12	12

Table 1: Statistics of experimental datasets

5 Experiments

In this section, we first briefly introduce the experimental datasets, evaluation criteria and baseline models. Then, we present how the experiments are prepared as well as the parameter settings. At last, extensive experiments were performed on three real-world datasets to answer the following research questions.

- RQ1: Whether the proposed approach outperforms the state-of-the-art long text generation methods or not?

- RQ2: How the proposed alignment methods affect the model performance.
- RQ3: How the model parameters and settings affect the model performance.
- RQ4: What is the quality of the generated long text?

5.1 Datasets and Evaluation Criteria

Three widely adopted real-world long text generation datasets were chosen in our experiments including arXiv (Clement et al., 2019) title-abstract dataset, ACL title-abstract dataset (Wang et al., 2018) and News-Report dataset (Hu et al., 2021). The input is respectively paper title and news data and the output is respectively abstract and financial report. The statistics of these experimental datasets are reported in Table 1.

For evaluation criteria, we adopted automatic evaluation and human evaluation. For automatic evaluation criteria, we chose the BLEU (Papineni et al., 2002), the ROUGE (Lin, 2004), the Meteor (Banerjee and Lavie, 2005), and the BERTScore (Zhang* et al., 2020). For human evaluation, we randomly select 900 generated output texts from 3 datasets and seek external human annotators to evaluate the coherence and the quality of the generated texts by different methods. The human score ranges from 1 to 5 and the higher the score the better the results.

5.2 Data Pre-processing and Parameters

The general pre-processing steps are performed on three datasets. Before embedding, each sentence will be inserted with the characters including SOS, EOS and UNK. Moreover, we restrict the length of the input data to 25 and each sentence is truncated or padded with a token (PAD) for short sentences. Similarly, the length of the output data is set to 300.

For the parameters of the matching algorithm, we set $K = 3$ for the top-K search.The BERT-based (Devlin et al., 2019) model having 12 layers was chosen for the generator and the hidden size is 768 with 12 attention heads. We adopt UNILM (Dong et al., 2019) for the generation. We vary the length of training texts from 50 to 300 and fine-tune the model using the Adam (Kingma and Ba, 2015). As for the decoding, beam search decoding is adopted with the beam size set to 3.

5.3 Baseline Models

To evaluate the effectiveness of the proposed approach, we compare it with several baseline models as well as the SOTA approaches.

- The Seq2seq model (Cho et al., 2014) (**S2S** for short) is to generate text from text which achieves superior performance in various related tasks like machine translation.
- The **S2S-Sen** model is the S2S model implemented by us for fair comparison where sentence level feature embedding is performed using the same settings as our approach.
- The **S2S-Att** (Bahdanau et al., 2015) model extends the sequence to sequence model by integrating with attention mechanism.
- The **S2S-Att-Sen** model is implemented version of the S2S-att where sentence level feature embedding is performed using the same settings as our approach.
- The **VAE** (Bowman et al., 2016) model is one of the most widely adopted deep generative model for text generation, and is chosen as the baseline model in our experiments. Similarly, the **VAE-Sen** is the implemented version of VAE.
- The **ml-VAE-D** (Shen et al., 2019) adopts two latent variables (ml-VAE-D) to build a hierarchical variational autoencoder for long text generation task on arXiv.
- The **pointer-generator network** (**PG**) (See et al., 2017) model is proposed for text summarization task. It adopts a pointer to copy terms from the external corpus and retains the ability to generate new words. This model achieves the superior model performance.
- The **Writing-editing** (Wang et al., 2018) adopts Seq2seq model with attention mechanism and generates paper abstract given title as input. An abstract draft is first generated and then iteratively revised several times.
- The **Multiple-edit** (Hu et al., 2021) model is proposed to resolve the news-to-report generation problem. It first generates the outline from news and then generates report using both input news and the learnt outline. This

Model	News-Report							arXiv							ACL						
	BLEU			ROUGE		meteor	Berts	BLEU			ROUGE		meteor	Berts	BLEU			ROUGE		meteor	Berts
	2	3	4	2	L	-	-	2	3	4	2	L	-	-	2	3	4	2	L	-	-
S2S (Cho et al., 2014)	7.60	4.59	2.79	4.92	9.77	8.42	59.03	4.48	0.85	0.17	3.53	12.30	11.77	77.28	8.05	2.95	1.09	4.49	16.67	15.71	81.93
S2S-Sen	15.07	9.18	5.62	9.78	19.83	4.68	57.74	7.43	1.35	0.26	4.21	16.34	10.95	81.16	8.14	2.69	0.88	3.94	19.09	10.47	82.61
S2S-Att (Bahdanau et al., 2015)	11.81	7.27	4.49	7.92	15.80	11.06	62.54	7.85	2.12	0.46	5.08	16.99	15.80	81.43	9.91	3.74	1.37	5.44	20.20	18.80	83.89
S2S-Att-Sen	15.49	9.46	5.84	9.94	20.23	4.62	58.35	7.29	1.38	0.29	4.02	17.53	10.48	81.16	7.50	2.43	0.82	3.44	19.67	9.91	82.58
VAE (Bowman et al., 2016)	8.50	4.40	2.14	4.84	17.20	6.10	58.99	6.41	1.01	0.20	3.99	15.34	14.60	80.36	6.33	1.33	0.35	2.90	15.05	14.91	82.12
VAE-Sen	9.52	4.25	1.79	5.26	16.56	2.61	56.41	5.24	0.46	0.07	2.52	14.50	9.82	80.20	4.73	1.17	0.45	1.87	14.70	9.39	81.95
ml-VAE-D (Shen et al., 2019)	7.64	3.57	1.73	2.78	16.93	12.53	60.49	4.40	2.74	0.75	1.22	13.61	14.23	77.31	3.82	0.89	0.86	4.88	18.48	19.07	78.68
PG (See et al., 2017)	14.66	6.66	3.39	8.99	37.13	8.07	58.49	7.66	3.49	1.55	3.96	18.65	13.59	76.54	6.01	2.88	2.24	3.35	23.82	15.97	77.45
Writing-editing(Wang et al., 2018)	9.40	4.93	2.65	6.64	13.32	7.69	59.67	9.53	5.23	2.61	4.86	15.81	13.98	81.02	11.02	6.06	3.81	7.61	26.21	17.88	83.50
Multi-edit (Hu et al., 2021)	9.90	4.35	2.03	5.86	34.48	3.70	55.81	6.07	3.52	0.78	4.01	16.78	8.90	76.15	5.23	2.04	0.89	5.38	19.53	11.47	76.13
CVAE-KD (Ren et al., 2020)	14.72	6.46	3.30	8.60	26.42	9.21	58.07	9.62	5.85	2.30	7.70	34.71	14.22	75.70	9.14	5.44	1.99	5.51	28.76	17.19	77.39
UNILM (Dong et al., 2019)	16.04	10.34	6.92	10.71	19.43	14.06	64.03	6.37	2.16	0.82	4.28	14.53	13.49	81.81	8.33	3.42	1.20	4.50	16.56	16.32	82.94
Our Approach	**34.41**	**32.31**	**30.45**	**31.75**	**39.87**	**38.20**	**75.78**	**27.80**	**21.86**	**18.26**	**25.82**	**43.19**	**35.72**	**82.00**	**33.05**	**28.46**	**25.55**	**31.26**	**50.52**	**26.95**	**84.18**
w/o SOM	21.03	16.00	12.98	16.86	25.45	19.17	65.84	20.66	15.26	12.05	17.10	34.91	30.52	80.72	19.57	15.14	12.52	15.67	35.51	21.77	81.25
Subgraph Matching	32.84	30.46	28.48	29.34	37.86	36.39	75.17	26.02	20.10	16.62	23.68	42.37	35.46	81.63	30.48	25.93	23.16	28.22	42.46	25.47	83.27
Generation Path	32.29	29.89	27.91	28.87	38.40	36.27	75.50	24.96	19.12	15.70	22.37	42.13	34.58	81.69	30.05	25.45	22.64	27.60	43.76	24.75	83.32
Probability Path	32.38	29.92	27.92	28.69	37.58	35.96	75.23	24.97	19.16	15.76	22.51	42.07	34.50	81.63	29.66	25.01	22.19	27.27	45.44	25.12	83.32
Improvement	4.78%	6.08%	6.93%	8.18%	3.82%	4.97%	0.36%	6.86%	8.77%	9.89%	9.03%	1.93%	0.73%	0.23%	8.41%	9.77%	10.28%	10.80%	11.16%	5.79%	0.35%

Table 2: Evaluation results of model performance comparison. The suffix "-Sen" means that the corresponding method only use sentence level feature embeddings and the 'Berts' refers to the BERTScore.

work is treated as the SOTA approach for comparison.

- The **CVAE-KD** (Ren et al., 2020) model is also proposed for the financial report generation task. It forces the latent variables to approximate the background knowledge distribution and adopts a teacher network to guide the generation of financial reports. This work is also considered as the SOTA approach for comparison.

- The **UNILM** (Dong et al., 2019) model is proposed to adopt a pre-trained BERT as both encoder and decoder for a sequence-to-sequence generation task, and is fine-tuned on our datasets for performance comparison.

5.4 Experimental Results

5.4.1 RQ1: Performance Comparison

We evaluate both the proposed approach as well as the compared models on three real-world datasets and the corresponding results are reported in Table 2. Note that the length of the generated text is 300. From this table, it is well observed that our approach outperforms both the baseline methods and the SOTA approaches. The performance of our approach against the second-best model is improved by 6.15% on average, and this verifies the effectiveness of the proposed approach. It is also noticed that, from the human evaluation results (Table 3), our approach is the best w.r.t. quality and coherence except for the quality score on ACL dataset. We check the human evaluation record and found that the quality of our generated text is actually comparable to that of UNILM (the reported best model w.r.t. the quality). However, the rating variance of the human annotator is quite large, and this might explain why the average quality value of our approach is a little bit lower than that of UNILM. We also observe that the results on ACL dataset are slightly better than those on arXiv and News-Report datasets from Table 2. The possible reason is that the underlying topics of the ACL dataset are focusing on NLP related content whereas the arXiv and the News-Report datasets obviously contains more diversified topics. Moreover, the average length of the News-Report dataset is much higher than that of the arXiv and ACL datasets which partially explains why the results on the News-Report data are also satisfying. This verifies the effectiveness of the proposed approach.

	News-Report		arXiv		ACL	
	coherence	quality	coherence	quality	coherence	quality
ml-VAE-D	2.90	2.86	3.02	2.90	3.44	3.62
Multi-edit	2.08	2.04	2.66	2.42	3.00	2.70
CVAE-KD	2.07	1.85	2.38	2.16	2.80	2.54
Writing-editing	3.62	2.98	3.72	3.52	3.74	3.60
UNILM	3.50	3.00	3.67	3.42	3.82	**3.62**
Our Approach	**4.18**	**3.42**	**4.14**	**3.66**	**4.09**	3.45

Table 3: Human evaluation results.

5.4.2 RQ2: Effect of Alignment Methods

To evaluate the effectiveness of the proposed three kinds of alignment methods, we respectively evaluate our approach by using only one kind of alignment method, denoted as "Subgraph Matching", "Generation Path" and "Probability Path", and the results are reported in Table 2. From these results, we notice that the model performance of "Subgraph Matching" is better than that of the rest two methods. This clearly shows that the matched subgraph contains more complex semantic meanings induced by multiple directed paths, i.e. the complex logic order. It is also observed that the model perfor-

Input News	The effect of the "policy by city" regulation continued, and third-tier cities benefited from this week. The growth rate of real estate sales in large and medium-sized cities declined month-on-month. The sales area of first-tier, second-tier, and third-tier cities decreased week-on-month, while sales area in third-tier cities increased month-on-month.
Target Report	In terms of property market transaction, the data of commercial real estate transaction in 30 large and medium-sized cities show that the current property market transactions still maintain a relatively deserted situation that began in October last year. Comprehensively speaking, relevant policies for the prevention and control of the property market can be said to be effective. The possible negative impact has attracted greater attention. We pointed out in the special report: As the current real estate stock projects account for a relatively large proportion, the overall policy tendency of increasing land supply in second-tier cities will also affect the volume of new projects. Certain support, on the whole, we expect that the growth rate of real estate investment should generally show a slight downward trend. In 2017, its year-on-year growth rate was about 3.6%. However, although the current tightening of real estate related policies has exerted a strong restraint on further speculation in the property market At the same time, it will not cause the economy to fall into the dilemma of a hard landing. In the short term, the economy can still basically maintain a steady downward new normal state, but this does not mean that the property market risks are dissipated.
CVAE-KD	In the western region, the number of new housing starts increased compared with the same period of the year in the western region. The monthly existing home sales situation in the United States failed to continue the substantial growth in the month and the month-on-month decline. The sales situation of the new housing market also showed no significant bright spots. The economic recovery focuses on the background of gradually strengthening progress, and the negative impact of a series of stimulus policies expiring caused by the decline in demand may be the real reason. The US government's stimulus policy for the real estate market is the most important in the year. But here The government has not made a clear decision on whether to continue the tightening of the tax rebate policy for house purchases.
UNILM	Affected by the continuous regulation and control of real estate, the growth rate of real estate investment in 2012 will drop significantly, with a growth rate of only about 15%; subject to the high debt constraints of local governments, the growth rate of infrastructure investment will also be difficult to increase significantly; and the European debt crisis will deepen The impact of the second decline in the global economy triggered by the second decline in export growth will also see a significant slowdown; the slowdown in the growth of several major sectors will cause a significant decline in China's economy in the first half of the year, and the quarterly GDP growth rate may exceed the 8 steady growth measures. With the gradual relaxation of real estate control measures, inflation has clearly fallen behind the strengthening of market endogenous growth forces, and the worst period of the European debt crisis.
Our	According to the transaction data of commercial housing in 30 large and medium-sized cities, the transaction data of commercial housing in the 30 large and medium-sized cities shows that the current real estate transaction still maintains a relatively deserted situation that began in October last year. Comprehensively speaking, the relevant policies for the prevention and control of the real estate market can be said to be effective. The significant cooling of the country's economy has caused greater concern about the possible negative impact on the economy. We pointed out in the special report: Due to the relatively large proportion of current real estate projects, the overall policy tendency of increasing land supply in second-tier cities will also affect the new The increase in the number of projects has provided some support. It seems that the overall real estate investment growth rate should show a slight downward trend. However, we still adhere to the view that the RMB exchange rate will rise steadily, but the Fed's monetary policy tightening will be significantly greater than the European Central Bank. However, the decline in money market interest rates and the short-term improvement in liquidity may be due to the slowdown of foreign exchange growth and capital outflows.

Table 4: A case study for the subjective evaluation and the correctly generated texts are highlighted in blue.

mance of "Generation Path" is better than that of "Probability Path", and the reason is that "Generation Path" aligns with other paths by considering its complete path information whereas "Probability Path" only considers the start and the end nodes' information. If we only use the best-matched results of these three kinds of alignments, the model performance is further enhanced as shown in the results of "Our Approach".

5.4.3 RQ3: Parameter Sensitivity Analysis

Parameter Effect. In this experiment, we first vary the length of the generated text from 50 to 300 to evaluate how it affects the model performance, and the results, i.e., BLEU, ROUGE, METEOR and BERTScore, are plotted in Figure 4. From this figure, it is well noticed that the model performance gradually decreases with the length of the generated text except for the BERTScore due to its similarity-based but not the hit-based calculation method. We then evaluate how the parameter of the top-K search affect the model performance. We vary K from 1 to 5 and plot the results in Figure 5. From the figures, we noticed that the model performance is the best with K=3 and thus we fix this value in the rest experiments.

Ablation Study. We remove the revised SOM to evaluate effectiveness of the directed semantic graph, and the results denoted as "w/o SOM" are

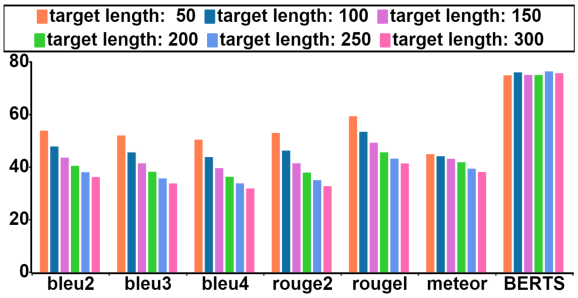
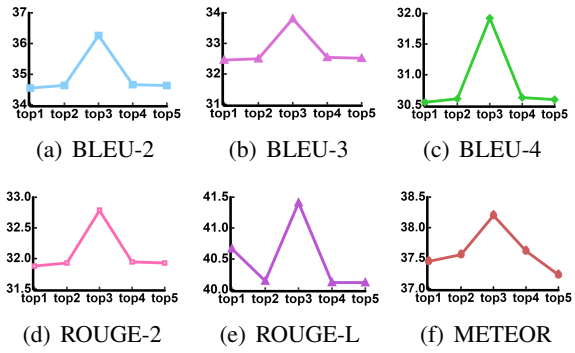


Figure 4: Evaluation results of the generated text with different lengths.



(a) BLEU-2 (b) BLEU-3 (c) BLEU-4

(d) ROUGE-2 (e) ROUGE-L (f) METEOR

Figure 5: Effect of parameter K.

reported in Table 2. From the results, it is observed that the model performance dramatically decreases after removing the SOM which verifies the effectiveness of the proposed SOM.

Effect of the Directed Semantic Graph. To evaluate the effect of the constructed directed semantic graph, we visualize the sentence order as well as the aligned results in Figure 6. In this figure, the sentences as well as their order (i.e., $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow ... \rightarrow 14$) are plotted in red, and their aligned clusters of the SOM graph are plotted in colored regions assigned with different numbers. Thus, the sequence order of the aligned SOM nodes is $27 \rightarrow 18 \rightarrow 27 \rightarrow 6 \rightarrow ... \rightarrow$. Notably, node 6 and 27 are aligned several times which indicates that the semantic meanings of these two nodes are important to generate the corresponding output text. Then, we observe that the main content of the input news is about "the central bank's liquidity policy". By checking the term frequency within cluster (semantic node) 6 and 27, we found that the highest frequent term of cluster 6 is "liquidity" and the highest frequent term of cluster 27 is "central bank". This demonstrates that the relevant semantic meanings could be well retrieved through the directed semantic graph.

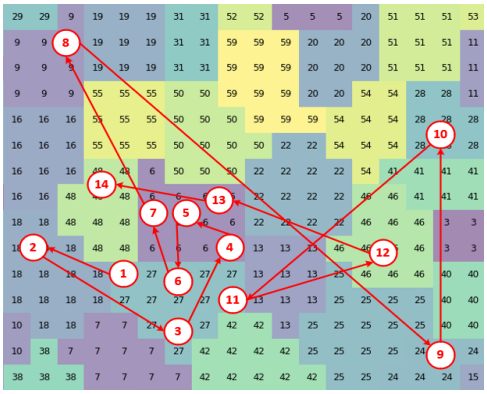


Figure 6: Effect of the directed semantic graph.

5.5 RQ4: A Case Study

To evaluate whether the generated text is logically coherent or not, we report the generated texts on the challenging News-Report dataset by CVAE-KD, UNILM and our approach in Table 4. First, the correctly generated texts are highlighted in blue and apparently, our approach achieves better results. Second, the input news is about "policy by city", i.e., the control policy for real estates of different cities. The generated texts of our approach contain a sequence of contents like "control of the real estate effective", "significant cooling of economy", "increasing land supply in second-tier cities" and "real estate investment growth rate should show a slight downward". Apparently, these generated sentences well reflect the causal effect for "policy by city". Unfortunately, the generated texts by the compared methods do not have such logic coherent content. This further verifies that the proposed approach could generate more coherent texts.

6 Conclusion

In this paper, we proposed a two-stage approach to generate coherent long text given the short input text. Particularly, we first build a document-level path and then construct the directed semantic graph. Three kinds of matching methods are proposed to extract the best matched paths or subgraphs for the later text generation. Extensive experiments are evaluated on three real-world datasets and the promising experimental results demonstrated that the proposed approach outperforms both baseline and the SOTA approaches.

7 Acknowledgement

This work was supported in part by the Shenzhen Science and Technology Program under Grant No. JCYJ20200109113201726 and the National Natural Science Foundation of China under Grant No. 61872108.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL Workshop*, pages 65–72.

Or Biran and Kathleen McKeown. 2015. Discourse planning with an n-gram model of relations. In *EMNLP-IJCNLP*, pages 1973–1977.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *CoNLL*, pages 10–21.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder

for statistical machine translation. In *EMNLP*, pages 1724–1734.

Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. 2019. On the use of arxiv as a dataset.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *NeurIPS*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for structuring story generation. In *ACL*, pages 2650–2660.

Wenxin Hu, Yunpeng Ren, and Xiaofeng Zhang. 2021. Generating financial reports from macro news via multiple edits neural networks. In *ECML*, pages 667–682.

Xinyu Hua and Lu Wang. 2019. Sentence-level content planning and style specification for neural text generation. In *EMNLP-IJCNLP*, pages 591–602.

Dongyeop Kang and Eduard Hovy. 2020. Plan ahead: Self-supervised text planning for paragraph completion task. In *EMNLP*, pages 6533–6543.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Matt J Kusner and José Miguel Hernández-Lobato. 2016. Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Yunpeng Ren, Ziao Wang, Yiyuan Wang, and Xiaofeng Zhang. 2020. Generating long financial report using conditional variational autoencoders with knowledge distillation. *arXiv preprint arXiv:2010.12188*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*, pages 1073–1083.

Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. 2019. Long and diverse text generation with planning-based hierarchical variational model. In *EMNLP-IJCNLP*, pages 3257–3268.

Dinghan Shen, Asli Celikyilmaz, Yizhe Zhang, Liqun Chen, Xin Wang, Jianfeng Gao, and Lawrence Carin. 2019. Towards generating long and coherent text with multi-level latent variable models. In *ACL*, pages 2079–2089.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *ICML*, pages 5926–5936.

Bowen Tan, Zichao Yang, Maruan AI-Shedivat, Eric P Xing, and Zhiting Hu. 2020. Progressive generation of long text. *arXiv preprint arXiv:2006.15720*.

Craig Thomson, Ehud Reiter, and Somayajulu Sripada. 2018. Comprehension driven document planning in natural language generation systems. In *INLG*, pages 371–380.

Qingyun Wang, Zhihao Zhou, Lifu Huang, Spencer Whitehead, Boliang Zhang, Heng Ji, and Kevin Knight. 2018. Paper abstract writing through editing mechanism. In *ACL*, pages 260–265.

Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Topic-guided variational auto-encoder for text generation. In *NAACL*, pages 166–177.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *AAAI*, pages 7378–7385.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, page 2852–2858.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *ICLR*.