InsertGNN: A Hierarchical Graph Neural Network for the TOEFL Sentence Insertion Problem

Fang WU
Stanford University, USA
fangwu97@stanford.edu

Stan Z. Li
Westlake University, China
stan.zq.li@westlake.edu.cn

Abstract

The integration of sentences poses an intriguing challenge within the realm of NLP, but it has not garnered the attention it deserves. Existing methods that focus on sentence arrangement, textual consistency, and question answering are inadequate in addressing this issue. To bridge this gap, we introduce InsertGNN which conceptualizes the problem as a graph, and employ a hierarchical Graph Neural Network (GNN) to comprehend the interplay between sentences. Our approach was rigorously evaluated on a TOEFL dataset, and its efficacy was further validated on the expansive arXiv dataset using cross-domain learning. Thorough experimentation unequivocally establishes InsertGNN's superiority over all comparative benchmarks, achieving an impressive 70% accuracy—a performance on par with average human test scores.

1 Introduction and Related Work

Sentence insertion (SI), initially introduced by Barzilay and Lapata (2008), stands as a crucial task for evaluating human linguistic prowess. However, with the advent of deep learning (DL), it has received scant attention over the past decade compared to other NLP domains like machine translation and text generation. As a result, the realm of DL lacks a standardized solution tailored to this particular challenge. To address this void, we curate a dataset sourced from the TOEFL exam, an English proficiency test that archives test-takers' comprehensive accuracy scores. Through this dataset, we investigate whether contemporary NLP techniques can surpass the performance of nonnative English speakers [1].

Sentence ordering (SO) and question answering (QA) constitute the two closely intertwined subdomains that share relevance with SI. On the one hand, SO initiates its approach by employing multi-layer perceptrons (MLPs) to facilitate pairwise order ranking (Chen et al., 2016) but can inadvertently propagate errors to a significant degree. To mitigate this concern, the pointer network (PN) (Gong et al., 2016) integrates an attention mechanism to enhance model capacity, building on subsequent advances such as the incorporation of attention and the deepening of the architecture (Logeswaran et al., 2016; Cui et al., 2018). However, the direct applicability of the PN framework to SI is hindered by the differing nature of its input. As a remedy, recent studies solve SO by plugging a coherence verifier (Jia et al., 2023) or through a non-autoregressive manner (Bin et al., 2023). Another line (Putra and Tokunaga, 2017) presents an unsupervised graph method that approaches the problem through the lens of sentence coherence and similarity within text. This method claims superiority when applied to the supervised entity grid and unsupervised Entity Graph scenarios. Yet, this approach, while effective, relies on a non-parametric structure and necessitates the construction of all potential graphs containing the extracted sentence. Meanwhile, QA provides a more universal pathway for our context. Notable works (Joshi et al., 2020; Abdel-Nabi et al., 2023) have established QA as a potent framework. In our scenario, the removed sentence and the paragraph at large can be likened to the question and context, respectively. These components are seamlessly amalgamated and channeled through a network to produce the sought-after position. However, this linear fusion poses challenges for cutting-edge models such as Transformers in comprehending the inherent logic connecting the concatenated paragraph and its designated slots.

In this paper, we represent SI as a directed graph, where each node represents a sentence, and the edge depicts their relative potential order. This pattern imitates the way people tackle SI, that is,

[1] Our compiled dataset is accessed at https://anonymous.4open.science/r/TOEFL-Sentence-Insertion-Dataset-899D/README.md

Findings of the Association for Computational Linguistics: EACL 2024, pages 173–180

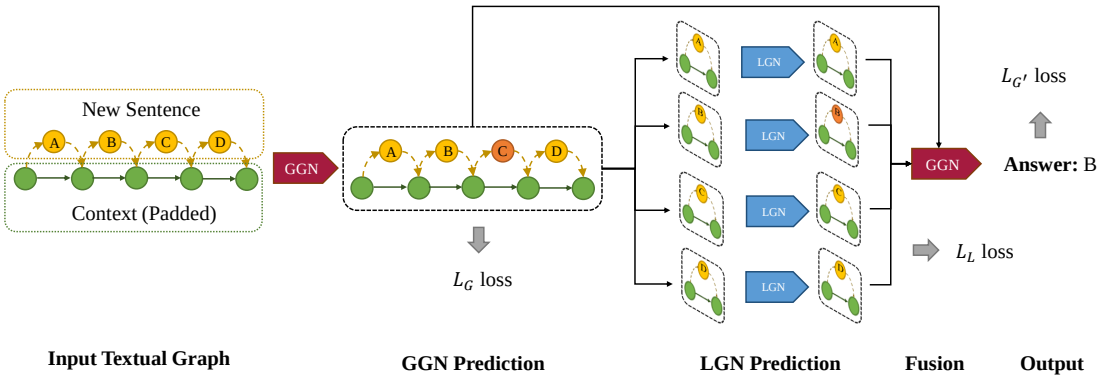

Figure 1: The architecture of our proposed InsertGNN.

putting the new sentence in each slot and checking the coherence. Then, we introduce a novel global-local fused GNN dubbed InsertGNN with delicately designed components to extract both global and local semantic information. Experiments on the TOEFL dataset with and without cross-domain learning convincingly show that InsertGNN surpasses all baselines and achieves competent performance compared to human examinees.

2 Methodology

2.1 Preliminary

As the TOEFL SI question has four options (A, B, C, D), we divide the input paragraph into five parts as $\{c_i\}_{i=1}^5$ to accommodate this setting. Then we pad the paragraph to make the graph complete if there is no leading sentence before the slot A or no ending sentence after the slot D. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (see Fig. 1) is built to describe all potential orders of the inserted sentence, where each node v_i represents a sentence, and the directed edge e_{ij} represents the relative order. If two sentences are connected, there is a directed edge between them. After that, we feed $\{c_i\}_{i=1}^5$ from the splitted context paragraph and the taken-out sentence q into a sentence encoder, obtaining representation vectors $\{\mathbf{S}_{c_i}\}_{i=1}^5$ and $\mathbf{S}_q$. These vectors serve as node features, where $\mathbf{S}_q$ corresponds to features of node $\{A, B, C, D\}$.

2.2 Global Graph Attention Networks

Contextual semantics play a vital role in the determination of insertion positions. To begin with, we apply an L-layer Global Graph Attention Network (GGN) (Veličković et al., 2017) to aggregate this global contextual information. The input feature is formulated as:

$$\mathbf{H}^0 = \{\underbrace{\mathbf{S}_{c1}, ..., \mathbf{S}_{c5}}_{\text{paragraph}}, \underbrace{\mathbf{S}_q, \mathbf{S}_q, \mathbf{S}_q, \mathbf{S}_q}_{\text{slots}}\}. \quad (1)$$

Then for a center node i and its neighbor j, the attention score at layer l is computed as:

$$e_{ij} = \rho(\alpha^T \cdot (\mathbf{W}\mathbf{H}_i^l \oplus \mathbf{W}\mathbf{H}_j^l)), \quad (2)$$

where ρ is the activation function and $\mathbf{W}$ is a trainable parameter. Attention weight is obtained by *softmax* as $\alpha_{ij} = \frac{\exp e_{ij}}{\Sigma_{k \in N_i}(\exp e_{ik})}$. After that, α_{ij} are used to update node features as:

$$\mathbf{H}_i^l = \sigma(\Sigma_{j \in N_i} \alpha_{ij} \mathbf{H}_j^{l-1} \mathbf{W}^l). \quad (3)$$

The representations for four slots in the last layer $\{\mathbf{H}_A^L, \mathbf{H}_B^L, \mathbf{H}_C^L, \mathbf{H}_D^L\}$ are fed into an MLP shared by the global-local fusion stage to obtain the prediction $\hat{y}$. Then for a batch with N samples, the binary cross-entropy (BCE) loss is calculated as:

$$L_G = -\frac{1}{N} \sum_{i=0}^{N} [\hat{y} \log y + (1-\hat{y}) \log (1-y)], \quad (4)$$

where $y \in \{0, 1\}$ is the ground truth label.

2.3 Local Graph Convolutional Networks

However, GGN is insensitive to local details (Zhang et al., 2020). Undeniably, the answer can sometimes be concluded by immediately reading the two sentences nearby the slot rather than the whole paragraph. Toward this goal, we utilize a Local Graph Network (LGN) to focus on local sentence interactions (see Fig. 2).

Concisely, we create four subgraphs with only the slot and its two surrounding sentences, whose features are the output of the previous GGN. Subgraphs are then fed into an M-layer parameter-shared GCN (Kipf and Welling, 2016), which is followed by a Weisfeiler-Lehman (WL) algorithm (Weisfeiler and Leman, 1968) to extract the multi-scale sub-tree features. The output of each layer $\mathbf{Z}^m$ is treated as WL's fingerprint. Similarly to DGCNN (Phan et al., 2018), we horizontally

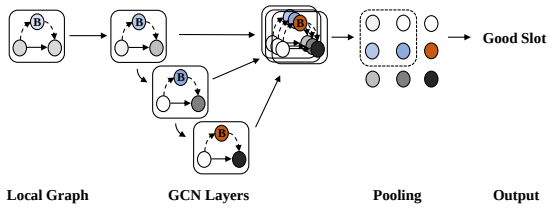

Figure 2: The illustration of LGN layers..

concatenate these fingerprints $\{\mathbf{Z}^1, ..., \mathbf{Z}^M\}$ rather than calculating the WL graph kernel. Then we apply a 4×4 pooling along with a fully-connected layer for subgraph classification. Finally, we compute the BCE loss L_L for these four subgraphs.

2.4 Global-local Fusion

At the fusion stage, $\mathbf{H}^L$ and $\mathbf{Z}^M$ are combined to integrate global and local information. We take a mean value of $\mathbf{Z}^M$ if node i is contained in more than one subgraph. The fused features $\mathbf{E} = \mathbf{H}^L + \text{mean}(\mathbf{Z}^M)$ go through another GGN, and the output for four slots is fed into the shared MLP to attain the final prediction $\hat{y}'$, which is used during the inference period. After that, a BCE loss $L_{G'}$ is calculated. The total loss is the sum of three BCE losses as $L = \alpha L_G + \beta L_L + \gamma L_{G'}$, where α, β and γ are loss weights of L_G, L_L and $L_{G'}$.

3 Experiment

3.1 Configurations and Datasets

Setups. We use the Sentence Transformer (Reimers and Gurevych, 2019) as the encoder to summarize the content of sentences, which makes sentences with similar meanings closer in vector space. It is first trained on Natural Language Inference (NLI) and then fine-tuned on the Semantic Textual Similarity benchmark (STSb) train set. Furthermore, we use BERT (Devlin et al., 2018) and its two variants DistillBERT (Sanh et al., 2019) and RoBERTa (Liu et al., 2019) for QA architecture. Notably, we neither fine-tune the baseline Transformers nor the Sentence Transformer, and only use them as an embedding layer. More details are in Appendix C.

TOEFL exams. TOEFL is one of the largest exams to test the English level of nonnative speakers hosted by the Educational Testing Service (ETS) globally. We chose it for two main reasons. First, all TOEFL questions are extracted from academic articles, designed by language experts and therefore are high quality. Second, ETS annually offers score data reports of examinees. According to the latest summary, the average precision in the reading section is 70.67%.

Every year, ETS releases merely a few articles in TOEFL Practice Online, which prevents us from building a large-scale dataset. We collected all questions since 2011 and got 156 samples with a relatively equal label distribution of 32%, 25%, 22%, and 21%.

ArXiv abstracts. We construct another dataset from arXiv to enrich the training samples and choose the abstract as the contextual paragraph, since it is independently readable and well-edited with strong logic clues. We abandon abstracts containing less than 5 sentences or 300 words to keep them more informative. Besides, we partially abandon categories that are not in the TOEFL scope. Moreover, several categories have tremendous mathematical formulations. These terms have no meaningful corresponding pretrained embeddings and should not be included in our supplementary dataset. 5965 abstracts remain after selection.

NLTK (Loper and Bird, 2002) is used to break the paragraph into sentences and randomly choose one as the sentence to be inserted. Then three other positions are selected to form a TOEFL-like problem. This operation can be repeated multiple times for each abstract since there are dozens of nonredundant combinations. The key statistics of these two datasets are listed in Appendix A.

3.2 Baselines

Unsupervised text coherence model. Putra and Tokunaga (2017) propose three algorithms to build coherence graphs. The main difference is the determination of the edges (see Appendix B).

In the Preceding Adjacent Vertex (PAV), a weighted directed edge is established from each sentence to the preceding adjacent sentence. Single Similarity Vertex (SSV) discards the constraints of precedence and adjacency. Multiple Similar Vertex (MSV) even relaxes the singular condition and allows multiple outgoing edges for each sentence as long as their corresponding similarity score exceeds a threshold θ. In the experiment, we use the same sentence encoder as our InsertGNN for the graph instead of GloVe (Pennington et al., 2014).

Topological sort. A research line (Prabhumoye et al., 2020; Sun et al., 2023) regard SI as a constraint learning problem. Sentences between two slots are represented as nodes with a known constraint between them. An MLP is used to predict

Method	Acc_TOEFL (%)
PAV	**41.66**
SSV	34.62
MSV	<u>37.18</u>

Table 1: Unsupervised learning accuracy. The best and second best are in bold and underlined, respectively.

the remaining constraints of the relative order between the taken-out sentence and other sentences. Lai et al. (2021) extend Sentence-Entity Graph Recurrent Network (SE-GRN) (Yin et al., 2019) and utilize two graph-based classifiers to iteratively make pairwise predictions for pairwise sentences.

QA methods. Here we formulate two types of QA architecture. The *P*-type (Plain) linearly combines the dependent sentence and the paragraph as the input and extracts the output vector of the [CLS] character as the final representation. The *A*-type (Altered) puts the new sentence in all four possible slots and classifies those differently filled paragraphs. The final prediction will be the one with the highest probability.

Large language models (LLMs). LLMs (Zhao et al., 2023) are posing a significant impact on the AI community. Here, we evaluate ChatGPT-3.5 with different prompts and report the best one. More details are listed in Appendix C.1.

3.3 Results

TOEFL. We test the unsupervised text coherence model first. PAV attains the highest accuracy of 41.66% on our TOEFL dataset (see Table 1), in accord with Putra and Tokunaga (2017)'s evaluation. It indicates that local cohesion is more important than long-distance cohesion, in line with our motivation for LGNs.

Next, we test the performance of supervised approaches with different proportions of validation. Experiments are repeated three times with different seeds, and we report the mean value. The result shows our InsertGNN observably improves upon other baselines including ChatGPT-3.5 in all cases. To be specific, InsertGNN reaches the highest accuracy of 71.5% with abundant training samples (see Table 2). It is comparable to the average human accuracy of 70.67%, suggesting that our model can do at least as well as non-native English speakers. We also conducted an ablation study to examine the contribution of each loss , and it can be discovered that all components are non-redundant (see Appendix D).

Method		Acc_TOEFL (%) (dev=0.05)	Acc_TOEFL (%) (dev=0.5)
BERT	*P*	42.86	38.46
	A	42.86	32.05
DistillBERT	*P*	57.14	35.89
	A	57.14	29.49
RoBERTa	*P*	42.86	35.89
	A	42.86	28.61
Topological Sort		57.14	34.62
SE-GRN		57.14	46.15
ChatGPT-3.5		<u>61.52</u>	<u>53.84</u>
InsertGNN		**71.43**	**55.12**

Table 2: Supervised learning accuracy under different validation split ratios of 0.05 and 0.5. *P* and *A* refers to the *P*-type and the *A*-type QA structure.

Method		Acc_arXiv (%)	Acc_TOEFL (%)
BERT	*P*	34.74	33.97
	A	32.63	30.13
DistillBERT	*P*	35.22	33.97
	A	31.33	30.76
RoBERTa	*P*	33.56	32.69
	A	3255	3141
Topological Sort		43.62	28.85
SE-GRN		<u>44.96</u>	36.53
ChatGPT-3.5		–	**60.76**
InsertGNN		**46.31**	<u>39.10</u>

Table 3: Cross-domain learning accuracy from the arXiv dataset to the TOEFL dataset. The left and right columns correspond to the accuracy in the arXiv test set and the whole TOEFL dataset, respectively.

From ArXiv to TOEFL. We further evaluate InsertGNN using cross-domain learning. Specifically, models are first trained on the arXiv dataset (source domain) with a validation splitting ratio of 0.05 and then directly tested on the TOEFL dataset (target domain). InsertGNN still stands out with an accuracy of 39.1% (see Table 3), outperforming all baselines except ChatGPT-3.5.

Moreover, the accuracy in TOEFL is generally lower than in arXiv because the contents of the two datasets are slightly different. ArXiv abstracts are a brief summarization and therefore very condensed. In contrast, the TOEFL paragraphs are an expanded narrative of a sub-point or a detailed explanation, which is more elaborate with a stronger inner logic. This manner of writing causes a decrease in accuracy when models are cross-domain.

4 Discussion

It is worth mentioning that the SI problem shares similarities with existing pretraining objectives like BERT's Next Sentence Prediction (NSP) and ALBERT's Sentence-Order Prediction (SOP). Specifically, NSP concatenates two masked sentences as inputs during pretraining. Sometimes they correspond to sentences that were next to each other in the original text, sometimes not. The model then has to predict if the two sentences were following each other or not. SOP primary focuses on inter-sentence coherence and is designed to address the ineffectiveness of the next sentence prediction (NSP) loss proposed in the original BERT.

5 Conclusion

In the paper, we propose a new sentence insertion framework called InsertGNN and build a TOEFL benchmark data set. Strong empirical evidence demonstrates the effectiveness of InsertGNN over existing language language models like ChatGPT in this interesting NLP task.

6 Limitations and Future Work

Despite the superior performance of our model, there are still some limitations left for future work. Most importantly, the TOEFL sentence insertion is small in size. Though we offer a compensatory dataset curated from the arXiv abstracts, more effort is needed to collect a larger and high-quality SI dataset. In addition, more powerful GPT models such as ChatGPT-4.0 are emerging, and it is worth evaluating these more advanced tools in our TOEFL sentence insertion task. Also, we believe more effective prompts can be mined to solve this specific task and we leave this for future study.

References

Heba Abdel-Nabi, Arafat Awajan, and Mostafa Z Ali. 2023. Deep learning-based question answering: a survey. *Knowledge and Information Systems*, 65(4):1399–1485.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Yi Bin, Wenhao Shi, Bin Ji, Jipeng Zhang, Yujuan Ding, and Yang Yang. 2023. Non-autoregressive sentence ordering. *arXiv preprint arXiv:2310.12640*.

Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2016. Neural sentence ordering. *arXiv preprint arXiv:1607.06952*.

Baiyun Cui, Yingming Li, Ming Chen, and Zhongfei Zhang. 2018. Deep attentive sentence ordering network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4340–4349.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jingjing Gong, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2016. End-to-end neural sentence ordering using pointer network. *arXiv preprint arXiv:1611.04953*.

Sainan Jia, Wei Song, Jiefu Gong, Shijin Wang, and Ting Liu. 2023. Sentence ordering with a coherence verifier. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9301–9314.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Shaopeng Lai, Ante Wang, Fandong Meng, Jie Zhou, Yubin Ge, Jiali Zeng, Junfeng Yao, Degen Huang, and Jinsong Su. 2021. Improving graph-based sentence ordering with iteratively predicted pairwise orderings. *arXiv preprint arXiv:2110.06446*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lajanugen Logeswaran, Honglak Lee, and Dragomir Radev. 2016. Sentence ordering and coherence modeling using recurrent neural networks. *arXiv preprint arXiv:1611.02654*.

Edward Loper and Steven Bird. 2002. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Anh Viet Phan, Minh Le Nguyen, Yen Lam Hoang Nguyen, and Lam Thu Bui. 2018. Dgcnn: A convolutional neural network over large-scale labeled graphs. *Neural Networks*, 108:533–543.

Shrimai Prabhumoye, Ruslan Salakhutdinov, and Alan W Black. 2020. Topological sort for sentence ordering. *arXiv preprint arXiv:2005.00432*.

Jan Wira Gotama Putra and Takenobu Tokunaga. 2017. Evaluating text coherence based on semantic similarity graph. In *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*, pages 76–85.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Peng Sun, Tomohiro Ohno, and Shigeki Matsubara. 2023. Japanese word reordering based on topological sort. In *ICAART (3)*, pages 768–775.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

B Weisfeiler and A Leman. 1968. The reduction of a graph to canonical form and the algebgra which appears therein. *NTI, Series*, 2.

Yongjing Yin, Linfeng Song, Jinsong Su, Jiali Zeng, Chulun Zhou, and Jiebo Luo. 2019. Graph-based neural sentence ordering. *arXiv preprint arXiv:1912.07225*.

Yaobin Zhang, Weihong Deng, Mei Wang, Jiani Hu, Xian Li, Dongyue Zhao, and Dongchao Wen. 2020. Global-local gcn: Large-scale label noise cleansing for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7731–7740.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A Details of Datasets

There we give an example of the SI problem in the TOEFL efficiency test (see Table 4).

Context Paragraph	**[A]** The age of rock art in Australia has been revised several times, with earlier dates suggested recently after new discoveries. **[B]** Accurate scientific proof has dated the first appearance of surface rock in Australia to approximately 30,000 to 50,000 years ago. **[C]** This lengthy and astounding history of rock art in Australia makes it the oldest art tradition known today in the world. **[D]**
New Sentence	Thanks to radiocarbon dating and technological development in studying evidence, researchers can now give a more precise age on this type of art.
Answer	**[B]** is the correct answer. Here, we need to match back "this type of art" to what it is referencing in order to correctly place the prompt sentence. It's referencing rock art, named in the first sentence, making the answer **[B]**.

Table 4: An example of TOEFL sentence insertion problem. It is extracted from questions of the third reading passage named "Rock Art of the Australia Aborigines" from TOEFL Practice Online 23.

In the experiments, we use two sorts of datasets, one is the collected TOEFL dataset, and the other is the arXiv abstract dataset. We summarize their statistics of them in Table 5.

Dataset	Size	Sentences	Words	Topics
TOEFL	156	7.31	133.94	Anthropology, Architecture, Astronomy, Economics, Biology, Chemistry, Communication, North America, Physics, Political Science, Psychology Sociology, World History
arXiv	5965	7.24	121.36	Astrophysics, Computer Vision and Pattern Recognition, Cryptography, Economics, General Relativity High Energy Physics-Theory, Information Theory, Networking and Internet Architecture, Quantum Physics

Table 5: Dataset statistics, including the sample size, average number of sentences inside each paragraph, the average number of words of each paragraph, and their main topics (categories).

B Some Baseline Models

There we provide a visualization of how the unsupervised text coherence method (Putra and Tokunaga, 2017) processes the SI problem in Fig. 3.

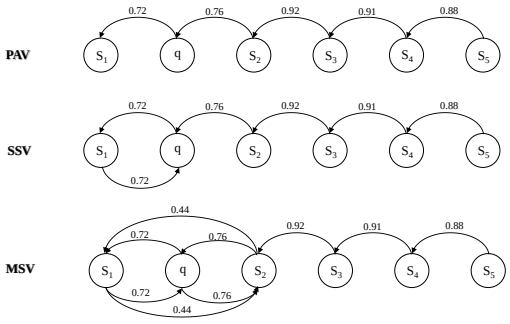

Figure 3: Three graph construction algorithms. PAV only allows edges between a sentence and its preceding sentence. SSV allows for edges between a sentence and any other sentence. MSV allows multiple edges.

C Experimental Details

There we explicitly describe the experimental configurations. All learnable models are trained on a single A100 GPU. Regarding MSV, we choose a threshold θ of 0.3. For both GGNs and LGNs, we utilize leaky Relu and Relu as the activation function, respectively, and include the dropout mechanism between layers with a dropout rate of 0.5. They both have one hidden layer and four hidden units. For GGN, we choose 16 hidden attention heads and four output attention heads with an attention dropout of 0.6 and a 0.2 negative slope of leaky Relu. They all have a residual connection. For all MLPs, we utilize Tanh as the activation function with no dropout. We adopt an Adam optimizer with a weight decay rate of 0.0005 and a random seed of 1234. After a grid search algorithm, we set $\alpha = 0.2$, $\beta = 0.2$ and $\gamma = 1.0$, where $L_{G'}$ is given more weights so that the model can focus more on the global-local fusion information. We train 100 epochs for InsertGNN with a learning rate of 0.0001 and 200 epochs for BERT-based models with a learning rate of 0.01.

C.1 Implementation of ChatGPT-3.5

For the powerful toolkit ChatGPT [2], prompt engineering is a key factor. We tried three different prompts to obtain the answer, which are listed below. We also provide the response template provided by ChatGPT-3.5. The empirical result shows that performance varies significantly according to different prompt inputs. The precision of prompts 1, 2, and 3 are 53.82%, 61.52%, and 30.76%, respectively, and we can find that prompt 2 is optimal.

Prompt 1. Please do the following sentence insertion problem, which has four choices denoted as A/B/C/D. The context paragraph is *xxx* and the sentence to be inserted is *xxx*. What is the answer?

Prompt 2. Given a sentence *xxx*, which place should it be inserted in the following paragraph *xxx*?

Prompt 3. Which paragraph is the most fluent? *xxx*, *xxx*, *xxx*, or *xxx*.

ChatGPT-3.5's answer for prompt 1. The most appropriate insertion point for the given sentence is before sentence [A/B/C/D]. Here is the revised paragraph: *xxx*.

ChatGPT-3.5's answer for prompt 2. The sentence *xxx* fits best after the sentence [A / B / C / D] in the paragraph. Here is the revised paragraph: *xxx*.

ChatGPT-3.5's answer for prompt 3. The most fluent paragraph is the first/second/third/fourth one: *xxx*.

D Additional Results

We implement ablation studies in the TOEFL data set with a 0.5 validation split ratio to verify the contributions of each constituent of our InsertGNN in Table 6. It can be observed that the LGN significantly promotes the model performance.

	L_G	L_L	$L_{G'}$	Accuracy
1	✓	-	-	0.4487
2	✓	✓	-	0.5256
3	✓	✓	✓	**0.5512**

Table 6: Ablation studies.

[2] https://openai.com/blog/chatgpt/