Modeling Structural Similarities between Documents for Coherence Assessment with Graph Convolutional Networks

Wei Liu[1], **Xiyan Fu**[2], **Michael Strube**[1]
[1]Heidelberg Institute for Theoretical Studies gGmbH
[2]Institute for Computational Linguistics, Heidelberg University
wei.liu@h-its.org, fu@cl.uni-heidelberg.de,
michael.strube@h-its.org

Abstract

Coherence is an important aspect of text quality, and various approaches have been applied to coherence modeling. However, existing methods solely focus on a single document's coherence patterns, ignoring the underlying correlation between documents. We investigate a GCN-based coherence model that is capable of capturing structural similarities between documents. Our model first creates a graph structure for each document, from where we mine different subgraph patterns. We then construct a heterogeneous graph for the training corpus, connecting documents based on their shared subgraphs. Finally, a GCN is applied to the heterogeneous graph to model the connectivity relationships. We evaluate our method on two tasks, assessing discourse coherence and automated essay scoring. Results show that our GCN-based model outperforms all baselines, achieving a new state-of-the-art on both tasks.

1 Introduction

Coherence describes the relationship between sentences that makes a group of sentences logically connected rather than just a random collection of them (Jurafsky and Martin, 2021). It is an important aspect of text quality (McNamara et al., 2010), and its modeling has been applied in many downstream tasks, including summarization (Parveen et al., 2015; Wu and Hu, 2018), dialogue generation (Mesgar et al., 2020; Xu et al., 2021), machine translation (Xiong et al., 2019; Tan et al., 2019) and document-level text generation (Wang et al., 2021; Diao et al., 2021). Given the importance of the task, there is a long line of methods proposed for coherence modeling.

Previous models leverage linguistic features to solve the problem. For example, entity grid-based methods (Barzilay and Lapata, 2005; Elsner and Charniak, 2011) capture the entity transition between adjacent sentences of a text to model local coherence; in contrast, graph-based models (Guinaudeau and Strube, 2013; Mesgar and Strube, 2015) measure coherence using the entity-graph of a document. Recently, neural network models (Li and Hovy, 2014; Li and Jurafsky, 2017; Mesgar and Strube, 2018; Xu et al., 2019b; Farag and Yannakoudakis, 2019; Moon et al., 2019; Jeon and Strube, 2020a,b; Mesgar et al., 2021) have been applied to the task due to their strength in representation learning and feature combination. Those models learn a document's representation from word embeddings or pre-trained language models, giving significantly better performance than previous statistical methods.

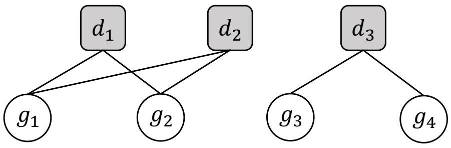


Figure 1: An example of structurally similar documents connected through subgraphs, in which d_2 is more similar in structure to d_1 than d_3. d_i denotes the i-th document and g_j denotes the j-th subgraph pattern, an edge will be built between them if the sentence graph of d_i contains subgraph g_j.

However, one drawback of existing neural-based methods is that they solely focus on extracting features within a single document, ignoring the underlying correlations between documents. Coherence describes how sentences of a text connect to each other (Reinhart, 1980; Foltz et al., 1998; Schwarz, 2001). Theoretically, documents with similar connection structures should tend to have similar degrees of coherence, which can be useful prior knowledge for coherence modeling. For example, a model is more likely to accurately assess a new document's coherence if it can refer to the labels of known documents with a similar organizational structure (see Appendix E.1 for an example).

To fill this gap, we investigate a graph-based approach to model the correlation between docu-

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7792–7808
July 9-14, 2023 ©2023 Association for Computational Linguistics

ments from the perspective of structural similarity. The main idea is to connect structurally similar documents through a graph and capture those connectivity relationships using Graph Convolutional Networks (GCN) (Kipf and Welling, 2017). In particular, inspired by Guinaudeau and Strube (2013), we first represent a document as a sentence graph, where nodes are sentences and two nodes will be connected if they contain semantically related nouns. Our method further converts each sentence graph into a subgraph set as it proves to be an efficient approach for measuring the topological similarity between graphs (Shervashidze et al., 2009; Kondor et al., 2009). Then, we construct a heterogeneous graph for the training corpus, containing document and subgraph nodes, based on subgraphs shared between documents. In this way, structurally-similar documents are explicitly linked through the subgraphs (shown in Figure 1). Finally, a GCN is applied to the heterogeneous graph to learn the representation of document nodes while considering the connections between them.

We evaluate our method on two benchmark tasks[1]: assessing discourse coherence and automatic essay scoring. Experimental results show that our method significantly outperforms a baseline model that does not consider structural similarities between documents, achieving a new state-of-the-art performance on both tasks. In addition, we provide a comprehensive comparison and detailed analysis, which empirically confirm that structural similarity information helps to mitigate the effects of uneven label distributions in datasets and improve the model's robustness across documents with different lengths.

2 Related Work

Our work is related to text coherence modeling and graph neural networks (GNN)-based methods for natural language processing (NLP).

Coherence Modeling. Inspired by Centering Theory (Grosz et al., 1995), Barzilay and Lapata (2005, 2008) propose an entity-based approach (the entity grid) to assess coherence by considering entity transitions between adjacent sentences of a text. The entity grid model has been improved by grouping entities based on their semantic relatedness (Filippova and Strube, 2007), incorporating entity-specific features (Elsner and Charniak, 2011), replacing grammatical roles with discourse roles (Lin et al., 2011). On the other hand, Guinaudeau and Strube (2013) propose an entity graph to capture entity transitions between not only adjacent sentences but also non-adjacent ones. Motivated by the functional sentence perspective of text coherence (Danes, 1974), Mesgar and Strube (2015, 2016) improve the entity graph with graph-based features extracted from text structures. Similarly, we also leverage the structural features of texts. However, instead of feeding individual documents' structure into the model as coherence patterns, we use them to capture the underlying correlation between documents.

With the advent of deep learning, neural networks have been applied to coherence modeling. Li and Hovy (2014); Xu et al. (2019b) learn to assess coherence by training a model to distinguish coherent texts from incoherent ones using different neural encoders. Tien Nguyen and Joty (2017); Joty et al. (2018) extend the entity grid model with a convolutional neural network. Moon et al. (2019) propose to enrich the coherence features of a document by considering discourse relations and syntactic patterns within it. Jeon and Strube (2020a) design structure-guided inter-sentence attention to learn a document's local and global coherence patterns. Inspired by human reading habits, Jeon and Strube (2020b) investigate a model to measure a document's coherence by incrementally interpreting sentences. Our work is in line with the above approaches to learning a coherence model based on neural networks. The main difference is that the above neural models focus on extracting features within a single document, whereas our graph-based approach aims to study the effectiveness of correlations between documents. Specifically, rooting on the linguistic definition of text coherence, we model the correlation from the perspective of structural similarities between documents.

GNN-based Methods for NLP. Graph neural networks are a family of neural networks that operate naturally on graphs. Many NLP problems can be expressed with a graph structure, so there is a surge of interest in applying GNNs for NLP tasks. Marcheggiani and Titov (2017) present a Syntactic GCN to learn latent feature representations of words in a sentence over dependency trees for Semantic Role Labeling. Yasunaga et al. (2017) propose a GCN-based multi-document summarization system that exploits the sentence relation information encoded in graph representations of document

[1] https://github.com/liuwei1206/StruSim

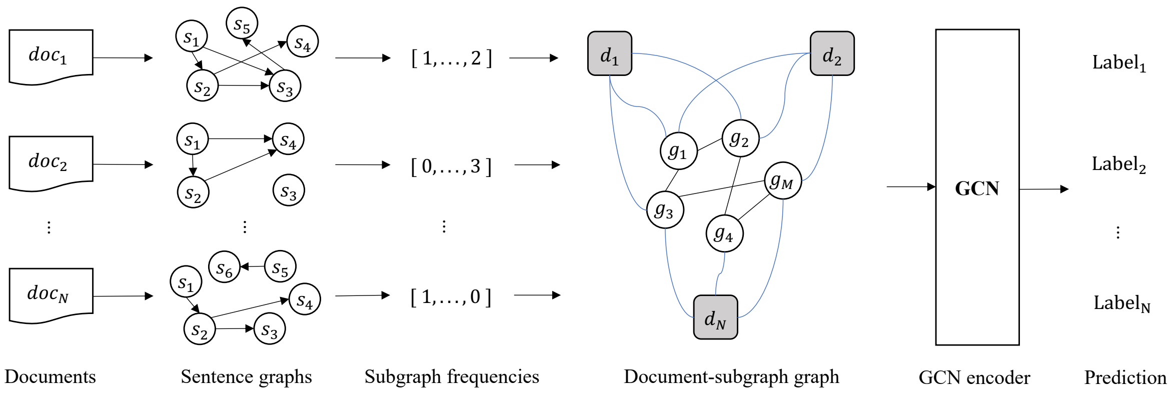


Figure 2: Overview of the proposed approach. Our method identifies a document's graph structure, converts the graph into a subgraph set, constructs a corpus-level graph based on the shared subgraphs between structurally similar documents, and finally encodes those connections with a GCN. For simplicity, we only show three documents and five subgraphs and limit the number of sentences in a document. s_u, d_i, and g_j denote the u-th sentence in a document, the i-th document in the training corpus, and the j-th defined subgraph, respectively.

clusters. Yao et al. (2019) build a graph containing documents and words as nodes and used the Text GCN to learn embeddings of words and documents. Lv et al. (2020) design a graph-based approach to encode structural information from ConceptNet for commonsense question answering. Compared with existing work, our graph-based method is different in both motivation and graph construction. For example, we specially design subgraph nodes to connect documents with a similar structure for capturing the structural correlations between samples.

3 Method

Figure 2 shows an overview of our proposed method. We describe step-by-step how to capture the structural similarities between documents, including i) identifying the structure of a document (Section 3.1); ii) representing the sentence graph of a document as a subgraph set (Section 3.2); iii) building a corpus-level heterogeneous graph to connect structurally similar documents based on the shared subgraphs (Section 3.3); iv) applying a GCN encoder to capture connectivity relationships between document nodes (Section 3.4).

3.1 Sentence Graph

To model the structural similarities between documents, we need to identify each document's structure. We follow Guinaudeau and Strube (2013) to represent a document as a directed sentence graph but with some modifications in graph construction. Specifically, in our implementation, two sentences are semantically connected if there are strong semantic relations between nouns in the two sentences. We use nouns instead of entities (Guinaudeau and Strube, 2013) because the former shows better performance than the latter in modeling semantic connection between sentences (Elsner and Charniak, 2011; Tien Nguyen and Joty, 2017).

Given a document, we use the Stanza toolkit (Qi et al., 2020) to segment it into sentences $\{s_1, s_2, ..., s_L\}$ and recognize all nouns in each sentence. For a pair of sentences s_u and s_v $(u < v)$, we compute the similarity score for each pair of nouns from them (one noun from s_u and the other from s_v) and use the maximum similarity score to measure their semantic connection. The score between two nouns is obtained by calculating the cosine value of their embedding. If the maximum similarity score is greater than the preset threshold δ, then the two sentences are considered semantically connected, and we add a directed edge between them (from s_u to s_v). After computing all combinations of s_u and s_v $(u < v)$ in the document, we can build a directed graph containing sentences as nodes (refer to Algorithm 1 in Appendix A).

3.2 Subgraph Set

After obtaining the graph structure of documents, we represent each sentence graph as a subgraph set. The subgraph set is an efficient way to compare topological similarities between graphs (Shervashidze et al., 2009), which we can employ to compare documents in terms of structure.

Graph g is a subgraph of graph G if the nodes in g can be mapped to the nodes in G and the connection relations within the two sets of nodes are the same. If the subgraph contains k nodes, we call it a k-node subgraph. In our method, we only consider subgraphs without backward edges. This

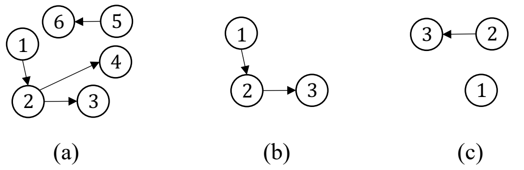


Figure 3: An example of subgraphs, in which graph (b) and graph (c) are 3-node subgraphs of graph (a).

is because when constructing the sentence graph, we process the document from left to right and never look back. We use weakly connected and disconnected subgraphs (shown in Figure 3) since we empirically find they both reflect the properties of a document in terms of coherence.

Given a sentence graph G_i of a document d_i, we first mine the contained k-node subgraphs by enumerating all combinations of k nodes and corresponding edges in G_i. Subgraphs with inter-sentence distances greater than w are filtered out because far-distant sentences are less likely to be related. It also reduces the search space when mining subgraphs. In the retained subgraphs, two can have the same structure but only differ in node IDs. We consider them as the same subgraph since they are isomorphic in graph theory. Then, we count the frequency of each k-node subgraph and identify the isomorphic subgraphs using the pynauty tool. Consequently, a sentence graph is represented as a k-node subgraph set (refer to Algorithm 2 in Appendix A).

3.3 Doc-subgraph Graph

A graph is an efficient way to model the correlation between items and has been used in different domains, such as knowledge graphs (Carlson et al., 2010) and social networks (Tang and Liu, 2009). We build a corpus-level undirected graph (on the training dataset), named doc-subgraph graph, to explicitly connect structurally similar documents through their shared subgraphs (shown in Figure 2). The graph contains document nodes and subgraph nodes, and the total number of nodes is the sum of the number of documents (N) and the number of k-node subgraph types (M) mined in Section 3.2. We design two types of edges in the graph, including edges between document and subgraph, and edges between subgraphs. We build the first type of edge if a document's subgraph set contains a subgraph, and set its weight as the product of the subgraph's normalized frequency in the subgraph set and the subgraph's inverse document frequency in the corpus. The definition of inverse document frequency is adopted from TF-IDF, but here it represents how common a subgraph is across subgraph sets of all documents. As for the second kind of edge, we construct it between two subgraphs that appear in the same subgraph set of a document, and its weight is the co-occurrence probability of these two subgraphs. We model the co-occurrence information between subgraphs because it has been shown helpful for comparing similar structures between graphs (Kondor et al., 2009).

Formally, we denote documents in a training corpus as $\mathbf{D} = \{d_1, d_2, ..., d_N\}$ and all types of k-node subgraphs mined from the corpus as $\mathbf{SubG} = \{g_1, g_2, ..., g_M\}$. We use G_i to denote the sentence graph of document d_i and $F_i = \{f_{i1}, f_{i2}, ..., f_{iM}\}$ to denote the k-node subgraph set mined from G_i, where f_{ij} denotes the frequency of subgraph g_j. We represent nodes in the doc-subgraph graph as $\mathbf{V} = \{v_1, ..., v_N, v_{N+1}, ..., v_{N+M}\}$, in which $\{v_1, ..., v_N\}$ are documents $\mathbf{D}$ and $\{v_{N+1}, ..., v_{N+M}\}$ are k-node subgraphs $\mathbf{SubG}$.

For any pair of document node v_i $(i \leq N)$ and subgraph node v_{N+j} $(j \leq M)$, we build an edge between them if g_j appears in the subgraph set of d_i, i.e. $f_{ij} > 0$, and define the edge's weight as:

$$A_{i,N+j} = \frac{f_{ij}}{\sum_{j'=1}^{M} f_{ij'}} \cdot \log \frac{N}{|d \in \mathbf{D} : g_j \in d|} \quad (1)$$

where the first term is the normalized frequency of subgraph g_j in subgraph set F_i, and the second term is an inverse document frequency factor, which diminishes the weight of subgraphs that occur frequently in subgraph sets and increases the weight of subgraphs that occur rarely. $|d \in \mathbf{D} : g_j \in d|$ represents the number of documents whose subgraph set contains subgraph g_j. A denotes the adjacency matrix of the doc-subgraph graph with shape $(N + M) \times (N + M)$ and is initialized as zero matrix. To make the graph symmetrical, we set the value of $A_{N+j,i}$ to be the same as $A_{i,N+j}$.

We also construct edges between any pair of subgraph nodes v_{N+j} and $v_{N+j'}$ $(j \leq M, j' \leq M, j \neq j')$ if g_j and $g_{j'}$ co-occur in the subgraph set of a document, i.e. $\exists\ d_i \in \mathbf{D} : f_{ij} > 0,\ f_{ij'} > 0$. The weight is set as the Pointwise Mutual Information (PMI) of these two subgraphs, which is a popular way (Ghazvininejad et al., 2016; Yao et al., 2019) to measure co-occurrence information:

$$A_{N+j,N+j'} = \log \frac{p(j, j')}{p(j)\, p(j')} \quad (2)$$

$$p(j) = \frac{|d \in \mathbf{D} : g_j \in d|}{N}$$
$$p(j, j') = \frac{|d \in \mathbf{D} : g_j \in d,\ g_{j'} \in d|}{N} \tag{3}$$

The PMI can be positive or negative, we follow previous work to clip negative PMI at 0 since this strategy works well across many tasks (Kiela and Clark, 2014; Milajevs et al., 2016; Salle and Villavicencio, 2019).

3.4 GCN Encoder

We adopt a GCN (Kipf and Welling, 2017) to encode the built doc-subgraph graph. GCN is a graph neural network which directly operates on graph-structured data. By integrating the normalized adjacency matrix, the GCN learns node representations based on both connectivity patterns and feature attributes of the graph (Li et al., 2018).

Formally, given the built graph with $(N + M)$ nodes, we represent the graph with an $(N + M) \times (N + M)$ adjacency matrix A. We first follow Kipf and Welling (2017) to add self-connections for each node:

$$\tilde{A} = A + I_{N+M} \tag{4}$$

where I_{N+M} is an identity matrix. Then, a two-layer GCN is applied on the graph, and the convolution computation at the l-th layer is defined as:

$$H^{(l)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l-1)}\mathbf{W}^{(l-1)}\right) \tag{5}$$

Here, $\tilde{D}$ is the degree matrix (i.e. $\tilde{D}_{i,i} = \sum_j \tilde{A}_{i,j}$) and $\mathbf{W}^{(l-1)}$ is a layer-specific trainable weight matrix. σ is an activation function, such as ReLU. $H^{(l)}$ is the output of l-th GCN layer; $H^{(0)} = X$, which is a matrix of node features. We use representations from the pre-trained model as features of document nodes due to its excellent performance on document-level tasks (Guo and Nguyen, 2020; Yin et al., 2021; Zhou et al., 2021). For subgraph nodes, since they have no textual contents, we set their features to zero vectors, which is a common setting in heterogeneous graphs (Ji et al., 2021). Finally, we feed the outputs of the two-layer GCN into a softmax classifier:

$$P = \mathrm{softmax}(H^{(2)}) \tag{6}$$

and train the model by minimizing the Cross-Entropy loss over document nodes:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{c=1}^{C} Y_{i,c} \cdot \log\left(P_{i,c}\right) \tag{7}$$

where Y_i is the label of document node v_i with a one-hot scheme, C is the number of classes.

While evaluating, for each document in the test corpus, we add it to the doc-subgraph graph, normalize the adjacent matrix of the new graph, and predict its label (refer to Appendix B).

4 Experiments

4.1 Datasets

We evaluate the proposed method on two benchmark tasks, assessing discourse coherence (ADC) and automated essay scoring (AES). The descriptive statistics of the dataset for each task are shown in Appendix C.

Assessing Discourse Coherence. ADC is the task of measuring the coherence of a given text. The benchmark dataset for this task is the Grammarly Corpus of Discourse Coherence (GCDC) dataset (Lai and Tetreault, 2018). Specifically, GCDC contains texts from four domains, including **Yahoo** online forum posts, emails from Hillary **Clinton**'s office, emails from **Enron**, and **Yelp** online business reviews. It is annotated by expert raters with a coherence score $\in \{1, 2, 3\}$, representing low, medium, and high levels of coherence, respectively.

Automated Essay Scoring. AES is a task to assign scores for essays, which has been used to evaluate coherence models (Burstein et al., 2010; Jeon and Strube, 2020b). We follow previous work (Jeon and Strube, 2020b) to employ the Test of English as a Foreign Language (TOEFL) dataset (Blanchard et al., 2014) in our experiments. The corpus contains essays from **eight prompts** along with score levels (low/medium/high) for each essay.

4.2 Experimental Settings

We implement our method based on the Pytorch library. The pre-trained embedding we use to calculate the similarity between nouns is GloVe (Pennington et al., 2014), and we set the similarity threshold δ to 0.65. For the subgraph set construction, we use 4-node subgraphs as basic units for the ADC task and 5-node subgraphs for the AES task, and limit the maximum sentence distance w to 8 for both tasks. The two-layer GCN is employed in our method, with ReLU as the activation function. We follow previous work (Jeon and Strube, 2020b) to use the representation from XLNet_{base} as document node features, and initialize XLNet using the pre-trained checkpoint from Huggingface. We use XLNet instead of other pre-trained models,

Model	Yahoo	Clinton	Enron	Yelp	Avg
Li and Jurafsky (2017)	53.50	61.00	54.40	49.10	54.50
Lai and Tetreault (2018)	54.90	60.20	53.20	54.40	55.70
Mesgar and Strube (2018)	47.30	57.70	50.60	54.60	52.55
Mesgar and Strube (2016)†	$61.30_{0.84}$	$64.60_{0.89}$	$55.74_{0.90}$	$56.70_{0.78}$	59.59
Moon et al. (2019)†	$56.80_{0.95}$	$60.65_{0.76}$	$54.10_{0.89}$	$55.85_{0.85}$	56.85
Jeon and Strube (2020a)†	$56.75_{0.83}$	$62.15_{0.88}$	$54.60_{0.97}$	$56.45_{0.97}$	57.49
Jeon and Strube (2020b)†	57.30	61.70	54.50	56.90	57.60
XLNet+DNN	$60.70_{1.03}$	$64.00_{1.36}$	$55.15_{1.14}$	$56.45_{0.94}$	59.10
Our Method	$\mathbf{63.65}_{0.74}$	$\mathbf{66.20}_{0.81}$	$\mathbf{57.00}_{0.81}$	$\mathbf{58.05}_{1.21}$	**61.23**

Table 1: Mean accuracy (std) results on GCDC.

such as BERT, because the TOEFL dataset contains long texts. For example, some essays have more than 800 words (maybe more than 1000 subwords). Autoencoding-based pre-trained models, such as BERT, limit input text length (usually 512 subwords), whereas XLNet can handle any input sequence length.

For the GCDC dataset, we follow the setting in Lai and Tetreault (2018) to perform 10-fold cross-validation over the training dataset. As for the TOEFL corpus, we conduct 5-fold cross-validation on the dataset of each prompt, which is a common setting for the AES task (Taghipour and Ng, 2016). Consistent with previous work (Lai and Tetreault, 2018; Jeon and Strube, 2020b), we use mean accuracy (%) as the evaluation metric. For more detailed settings and hyperparameters, please refer to Appendix D.

Baselines. To investigate the effectiveness of structural similarities between documents for coherence modeling, we empirically compare our method with a baseline model that does not use this knowledge. We call this baseline XLNet+DNN, which inputs document representations from XLNet as features, learns document embeddings with a two-layer deep neural network (DNN), and uses a softmax layer as the classifier. The only difference between the XLNet+DNN baseline and our method in terms of mathematical form is whether the regularized adjacency matrix $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ is applied (Li et al., 2018). We configure this baseline to have the same number of parameters as our method for a fair comparison.

We also compare with Mesgar and Strube (2016), which feeds subgraphs as input features. For a fair comparison, we input document representations from XLNet to this model, equip it with a two-layer DNN and softmax layer for feature extraction and classification. Furthermore, we compare our method against existing state-of-the-art models for each task to evaluate the effectiveness of our approach.

4.3 Overall Results

Assessing Discourse Coherence. Table 1 shows the experimental results on GCDC dataset[2]. The first three rows (Li and Jurafsky, 2017; Mesgar and Strube, 2018; Lai and Tetreault, 2018) in the first block show the performance of embedding-based models, and the last four rows (Mesgar and Strube, 2016; Moon et al., 2019; Jeon and Strube, 2020a,b) in the same block are the state-of-the-art models based on XLNet. With the pre-trained model as the encoder, the latter four models outperform embedding-based methods by a large margin.

We present the performance of the XLNet+DNN baseline and our method in the last two blocks of Table 1. As shown in the table, structural similarity information between documents is helpful for coherence assessment, which improves the average accuracy from 59.10% of the XLNet+DNN baseline to 61.23% of our method. Subgraphs as input features (Mesgar and Strube, 2016) can also enhance performance, but the improvement is much smaller than our method. We speculate that simply concatenating subgraph features cannot efficiently capture structural similarities between documents. By contrast, our method explicitly connects structurally similar documents via a graph, thereby fully utilizing this information. Surprisingly, our simple baseline outperforms previous state-of-the-art models, which are also built on XLNet. This is likely because the GCDC dataset mainly contains short and informal texts, whereas previous sota models were designed to handle long and well-formed documents. By contrast, our method works well on the corpus, achieving the best performance.

Automated Essay Scoring. As mentioned in Section 4.1, AES is a task for scoring the quality of essays and has been used to evaluate coherence

[2] In Tables 1, 2, † denotes that the same XLNet as our method is employed in the model.

Model	Prompt								Avg
	1	2	3	4	5	6	7	8	
Dong et al. (2017)	69.30	66.47	65.84	66.38	68.89	64.20	67.11	65.73	66.74
Mesgar and Strube (2016)†	75.31$_{0.77}$	74.90$_{0.94}$	73.42$_{0.81}$	74.35$_{1.18}$	76.10$_{0.74}$	75.42$_{0.68}$	72.48$_{0.83}$	72.31$_{0.65}$	74.29
Moon et al. (2019)†	73.84$_{0.81}$	72.54$_{0.87}$	72.32$_{1.27}$	73.26$_{0.67}$	75.34$_{0.72}$	74.72$_{0.78}$	71.97$_{0.71}$	72.14$_{0.93}$	73.27
Jeon and Strube (2020a)†	75.10$_{0.74}$	73.35$_{0.92}$	74.75$_{0.61}$	74.18$_{1.07}$	76.38$_{0.91}$	74.30$_{1.13}$	73.61$_{0.72}$	73.44$_{1.15}$	74.39
Jeon and Strube (2020b)†	75.60	73.40	75.00	73.50	76.80	75.20	73.50	72.80	74.48
XLNet+DNN	74.70$_{0.88}$	74.46$_{0.97}$	73.07$_{0.92}$	74.09$_{1.04}$	75.45$_{0.83}$	75.21$_{0.94}$	71.17$_{0.76}$	71.95$_{0.81}$	73.84
Our Method	**75.97**$_{1.14}$	**76.25**$_{1.07}$	**74.14**$_{1.18}$	**75.81**$_{0.71}$	**77.01**$_{0.94}$	**77.08**$_{1.14}$	**73.55**$_{0.80}$	**72.91**$_{0.66}$	**75.34**

Table 2: Mean accuracy (std) results on TOEFL.

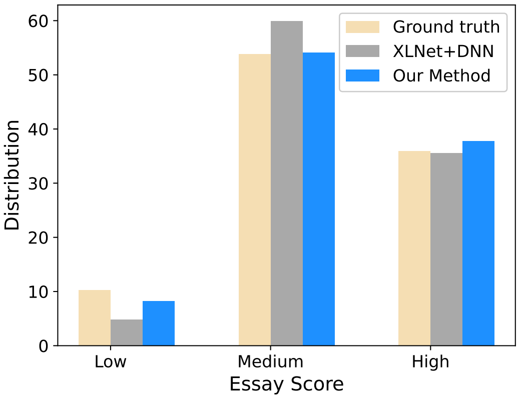


Figure 4: Predicted label distribution.

models. Hence, to better illustrate the effectiveness of our approach, we report the performance of both existing coherence models (Mesgar and Strube, 2018; Moon et al., 2019; Jeon and Strube, 2020a,b) and models designed to solve the AES task. For the latter, we report the result of Dong et al. (2017), which is a state-of-the-art method for the AES task.

Results on the TOEFL dataset are shown in Table 2. Previous coherence models and the XLNet+DNN baseline give significantly better performance than the AES model in Dong et al. (2017). Similar to the results on the GCDC dataset, subgraphs as input features can slightly improve the performance. However, the XLNet+DNN baseline can not beat the state-of-the-art coherence models on the TOEFL dataset. The results are reasonable because those coherence models are not only based on XLNet but also consider the characteristics of long documents. Consistent with observations on GCDC, our method, considering the structural similarities between documents, outperforms the XLNet+DNN baseline on the TOEFL dataset, giving state-of-the-art results.

4.4 Performance Analysis

To understand how structural similarity works for coherence modeling, we compare our model with the XLNet+DNN baseline in terms of the predicted label distribution and document length.

Model	Yahoo	Clinton	Enron	Yelp	Avg
XLNet+DNN	47.32$_{1.56}$	46.16$_{1.77}$	42.86$_{1.85}$	39.32$_{1.73}$	43.91
Our Method	**51.92**$_{1.06}$	**48.49**$_{1.61}$	**45.67**$_{1.57}$	**44.18**$_{1.10}$	**47.66**

Table 3: Mean F1-Macro results (std) on the GCDC.

Predicted Label Distribution. Figure 4 shows the distributions of predicted essay scores from the XLNet+DNN baseline and our model on the TOEFL P1 dataset. The XLNet+DNN's predictions are strongly biased, with about 60% of essays predicted as medium scores. We speculate this is caused by the uneven label distribution in the TOEFL P1 dataset (10.3%/53.8%/35.9% of low/medium/high-scoring essays). By contrast, our model is less affected by the uneven distribution, making more low and high score predictions. We also collect the prediction accuracy of the two models for each essay score. The prediction accuracy of the XLNet+DNN model for low, medium, high scores is 35.29%, 83.71%, 76.47%, and that of our method is 50.00%, 82.02%, 84.87%. XLNet+DNN mainly predicts medium scores, so this label's recall value is high. Compared with the baseline, our method makes relatively accurate predictions for all essay scores, suggesting that capturing structural similarities between essays helps mitigate the effects of uneven label distribution and thus focuses on learning coherence patterns.

To better verify this, we report the performance of the XLNet+DNN baseline and our method using F1-Macro as the evaluation metric. F1-Macro computes the accuracy for each class independently and then takes the average at the class level. Intuitively, if our model's predictions are more uniform and accurate, the F1-Macro performance gap between our method and the XLNet+DNN baseline should be no smaller than the gap in terms of accuracy. Table 3 shows the F1-Macro results of the XLNet+DNN baseline and our model on the GCDC dataset. Our method achieves much better F1-Macro results than the XLNet+DNN baseline, and the gap between the two models in F1-Macro is larger than the gap in accuracy, which further demonstrates that our

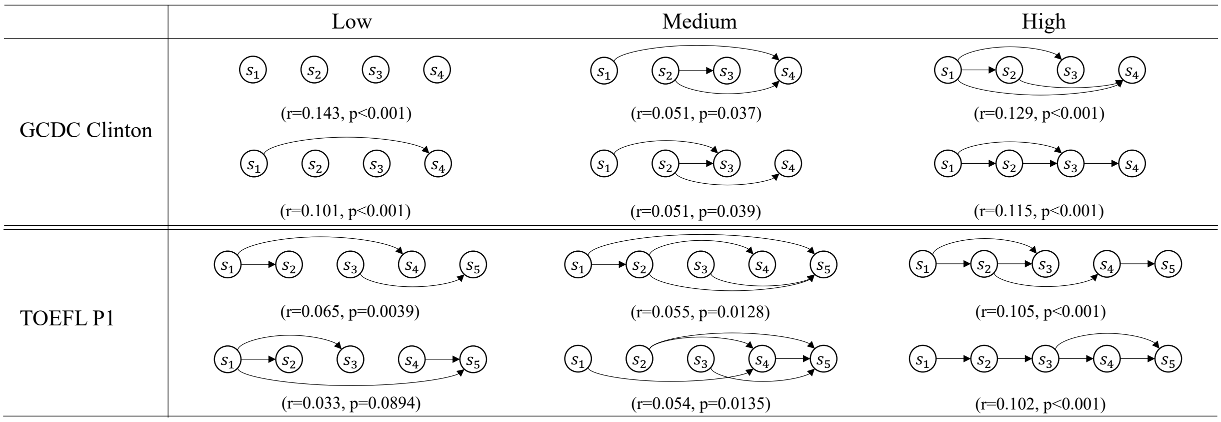


Figure 5: The top two most positively correlated subgraphs for each coherence level on the GCDC Clinton and TOEFL P1. r denotes the correlation coefficient value, and p is the p_value ($p < 0.05$ means statistically significant).

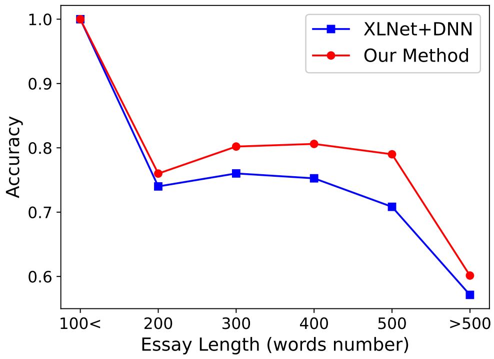


Figure 6: Accuracy against essay length.

model makes more even and accurate predictions.

Document Length. Figure 6 shows the accuracy trends of the baseline and our method on the TOEFL P1 dataset as essays become longer. The curve of XLNet+DNN generally shows a downward trend, accuracy decreasing as essays' length increases. The result is not surprising since long documents contain more complicated semantics and thus are more challenging. Our method performs similarly to the XLNet+DNN baseline over short documents (length <= 200). But when essays become longer, our model gives relatively high accuracy and even presents a slight increase (200 < length <= 400). This suggests that structural similarity information helps to improve the model's robustness when document length increases.

4.5 Ablation Study

We analyze the effectiveness of each type of edge in our method. To this end, we test the performance of our approach by first removing edges between subgraph nodes (ESS) and then removing edges between document node and subgraph node (EDS). Note that if all edges are removed (i.e. each document is an isolated node), our method degrades into the XLNet+DNN baseline.

Model	GCDC Clinton	TOEFL P1
	Acc	Acc
Our Method	**66.20**	**75.97**
- ESS	66.00	75.42
- ESS, EDS	64.00	74.70

Table 4: Ablation study for different edges on the GCDC Clinton and TOEFL P1 dataset.

Table 4 shows the results on the GCDC Clinton and TOEFL P1 datasets. We can observe from the table that eliminating any type of edges would hurt the performance. The decline in performance is more significant when removing the EDS than eliminating the ESS. The results are reasonable because edges between documents and subgraphs are the key to connecting documents with similar structures, while edges between subgraphs are considered to further assist it (Kondor et al., 2009).

4.6 Subgraph Analysis

In this section, we statistically investigate which subgraphs (sentence connection styles) mostly correlate to each level of coherence[3]. Specifically, we calculate the Pearson correlation coefficient between each subgraph and per label, and test the significance of the correlation. Figure 5 shows the top two results on GCDC Clinton and TOEFL P1.

In general, subgraphs positively correlated with higher coherence tend to contain more edges. This is somewhat aligned with the previous finding (Guinaudeau and Strube, 2013) that coherence correlates with the average out-degree of sentence graphs. Weakly connected subgraphs are more

[3] We perform this analysis on the whole corpus and show readable text examples in Appendix E.2

likely to reflect higher coherence than disconnected ones. Taking results on GCDC Clinton as an example, the top two most correlated subgraphs for low coherence contain isolated nodes or components while nodes in subgraphs for high coherence are (weakly) connected. Furthermore, subgraphs with more connections between adjacent sentences seem more correlated with high coherence. For example, there is an almost linear subgraph (or contains linear structure) in the high category of both datasets. We also find that the subgraph results per coherence level on the GCDC Clinton dataset differ from that on the TOEFL P1 dataset. This could be due to two reasons. First, the two datasets contain texts from various domains with domain-specific writing styles and structures. Second, they are built by different annotators, who may have different preferences for text organization styles.

5 Conclusion

In this paper, we investigated the effectiveness of structural similarity information between documents for coherence modeling. We proposed a graph-based method to connect structurally similar documents based on shared subgraphs, and model the connectivity relations with a GCN. Experiments on two benchmark tasks show that our method consistently outperforms the baseline model, achieving state-of-the-art results on both tasks.

6 Limitations

Despite showing impressive performance, our graph-based approach still has several limitations. The first one is related to the construction of the sentence graph. At present, we consider two sentences to be semantically related if they share similar nouns. But coherence can be achieved not only by describing similar entities but also by discourse (rhetorical) relations (Jurafsky and Martin, 2021). So it will be an exciting direction to incorporate discourse relations into the construction of a graph. The second one is that we implemented our method using only a plain GCN. Recent work has pointed out that the original GCN can be further improved with more advanced aggregation functions (Xu et al., 2019a) or attention mechanisms (Velickovic et al., 2018). So another interesting direction is to explore the benefits of more powerful graph neural networks for our method, which we leave for future study.

Acknowledgements

The authors would like to thank the four anonymous reviewers for their comments. This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany. The first and second authors have been supported by a Heidelberg Institute for Theoretical Studies Ph.D. scholarship.

References

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 141–148, Ann Arbor, Michigan. Association for Computational Linguistics.

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

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2014. ETS corpus of non-native written english ldc2014t06. *Philadelphia, Penn.: Linguistic Data Consortium*.

Jill Burstein, Joel Tetreault, and Slava Andreyev. 2010. Using entity-based features to model coherence in student essays. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 681–684, Los Angeles, California. Association for Computational Linguistics.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Jie Chen, Tengfei Ma, and Cao Xiao. 2018. FastGCN: Fast learning with graph convolutional networks via importance sampling. In *International Conference on Learning Representations*.

Frantisek Danes. 1974. Functional sentence perspective and the organization of the text. *Papers on functional sentence perspective*, 23:106–128.

Shizhe Diao, Xinwei Shen, Kashun Shum, Yan Song, and Tong Zhang. 2021. TILGAN: Transformer-based implicit latent GAN for diverse and coherent text generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4844–4858, Online. Association for Computational Linguistics.

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162, Vancouver, Canada. Association for Computational Linguistics.

Micha Elsner and Eugene Charniak. 2011. Extending the entity grid with entity-specific features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 125–129, Portland, Oregon, USA. Association for Computational Linguistics.

Youmna Farag and Helen Yannakoudakis. 2019. Multi-task learning for coherence modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 629–639, Florence, Italy. Association for Computational Linguistics.

Katja Filippova and Michael Strube. 2007. Extending the entity-grid coherence model to semantically related entities. In *Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG 07)*, pages 139–142, Saarbrücken, Germany. DFKI GmbH.

Peter W. Foltz, Walter Kintsch, and Thomas K. Landauer. 1998. The measurement of textual coherence with latent semantic analysis. *Discourse processes*, 25(2-3):285–307.

Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. 2016. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1191, Austin, Texas. Association for Computational Linguistics.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Camille Guinaudeau and Michael Strube. 2013. Graph-based local coherence modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 93–103, Sofia, Bulgaria. Association for Computational Linguistics.

Zhiyu Guo and Minh Le Nguyen. 2020. Document-level neural machine translation using BERT as context encoder. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 101–107, Suzhou, China. Association for Computational Linguistics.

Sungho Jeon and Michael Strube. 2020a. Centering-based neural coherence modeling with hierarchical discourse segments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7458–7472, Online. Association for Computational Linguistics.

Sungho Jeon and Michael Strube. 2020b. Incremental neural lexical coherence modeling. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6752–6758, Barcelona, Spain (Online). International Committee on Computational Linguistics.

H. Ji, C. Yang, C. Shi, and P. Li. 2021. Heterogeneous graph neural network with distance encoding. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1138–1143, Los Alamitos, CA, USA. IEEE Computer Society.

Shafiq Joty, Muhammad Tasnim Mohiuddin, and Dat Tien Nguyen. 2018. Coherence modeling of asynchronous conversations: A neural entity grid approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 558–568, Melbourne, Australia. Association for Computational Linguistics.

Daniel Jurafsky and James H Martin. 2021. Speech and language processing (3rd ed. draft).

Douwe Kiela and Stephen Clark. 2014. A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 21–30, Gothenburg, Sweden. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Risi Kondor, Nino Shervashidze, and Karsten M. Borgwardt. 2009. The graphlet spectrum. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 529–536.

Alice Lai and Joel Tetreault. 2018. Discourse coherence in the wild: A dataset, evaluation and methods. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 214–223, Melbourne, Australia. Association for Computational Linguistics.

Jiwei Li and Eduard Hovy. 2014. A model of coherence based on distributed sentence representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2039–2048, Doha, Qatar. Association for Computational Linguistics.

Jiwei Li and Dan Jurafsky. 2017. Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 198–209, Copenhagen, Denmark. Association for Computational Linguistics.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 3538–3545.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2011. Automatically evaluating text coherence using discourse relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 997–1006, Portland, Oregon, USA. Association for Computational Linguistics.

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. 2020. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8449–8456.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Danielle S McNamara, Scott A Crossley, and Philip M McCarthy. 2010. Linguistic features of writing quality. *Written Communication*, 27(1):57–86.

Mohsen Mesgar, Sebastian Bücker, and Iryna Gurevych. 2020. Dialogue coherence assessment without explicit dialogue act labels. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1439–1450, Online. Association for Computational Linguistics.

Mohsen Mesgar, Leonardo F. R. Ribeiro, and Iryna Gurevych. 2021. A neural graph-based local coherence model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2316–2321, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mohsen Mesgar and Michael Strube. 2015. Graph-based coherence modeling for assessing readability. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 309–318, Denver, Colorado. Association for Computational Linguistics.

Mohsen Mesgar and Michael Strube. 2016. Lexical coherence graph modeling using word embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1414–1423, San Diego, California. Association for Computational Linguistics.

Mohsen Mesgar and Michael Strube. 2018. A neural local coherence model for text quality assessment. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4328–4339, Brussels, Belgium. Association for Computational Linguistics.

Dmitrijs Milajevs, Mehrnoosh Sadrzadeh, and Matthew Purver. 2016. Robust co-occurrence quantification for lexical distributional semantics. In *Proceedings of the ACL 2016 Student Research Workshop*, pages 58–64, Berlin, Germany. Association for Computational Linguistics.

Han Cheol Moon, Tasnim Mohiuddin, Shafiq Joty, and Chi Xu. 2019. A unified neural coherence model. 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)*, pages 2262–2272, Hong Kong, China. Association for Computational Linguistics.

Daraksha Parveen, Hans-Martin Ramsl, and Michael Strube. 2015. Topical coherence for graph-based extractive summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1949–1954, Lisbon, Portugal. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher 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, Doha, Qatar. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Tanya Reinhart. 1980. Conditions for text coherence. *Poetics Today*, 1(4):161–180.

Alexandre Salle and Aline Villavicencio. 2019. Why so down? The role of negative (and positive) pointwise mutual information in distributional semantics. *arXiv preprint arXiv:1908.06941*.

Monika Schwarz. 2001. Establishing coherence in text. conceptual continuity and text-world models. *Logos and Language*, 2(1):15–24.

Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. 2009. Efficient graphlet kernels for large graph comparison. In *Artificial intelligence and statistics*, pages 488–495. PMLR.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas. Association for Computational Linguistics.

Xin Tan, Longyin Zhang, Deyi Xiong, and Guodong Zhou. 2019. Hierarchical modeling of global context for document-level neural machine translation. 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), pages 1576–1585, Hong Kong, China. Association for Computational Linguistics.

Lei Tang and Huan Liu. 2009. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 817–826.

Dat Tien Nguyen and Shafiq Joty. 2017. A neural local coherence model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1320–1330, Vancouver, Canada. Association for Computational Linguistics.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Ziao Wang, Xiaofeng Zhang, and Hongwei Du. 2021. Building the directed semantic graph for coherent long text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2563–2572, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuxiang Wu and Baotian Hu. 2018. Learning to extract coherent summary via deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, volume 32, pages 5602–5609.

Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Modeling coherence for discourse neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7338–7345.

Jun Xu, Zeyang Lei, Haifeng Wang, Zheng-Yu Niu, Hua Wu, and Wanxiang Che. 2021. Discovering dialog structure graph for coherent dialog generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1726–1739, Online. Association for Computational Linguistics.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019a. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Peng Xu, Hamidreza Saghir, Jin Sung Kang, Teng Long, Avishek Joey Bose, Yanshuai Cao, and Jackie Chi Kit Cheung. 2019b. A cross-domain transferable neural coherence model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 678–687, Florence, Italy. Association for Computational Linguistics.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canada. Association for Computational Linguistics.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. DocNLI: A large-scale dataset for document-level natural language inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online. Association for Computational Linguistics.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14612–14620.

A Graph Construction

Algorithm 1 Constructing sentence graph

```
Input: Document $d$, threshold $\delta$
Output: Sentence graph G
$S, NS \leftarrow$ stanza($d$)                ▷ Sentences and nouns
$L \leftarrow$ len($S$)
$G \leftarrow$ zeros($L, L$)                  ▷ Init adjacency matrix
for $u \leftarrow 1$ to $L - 1$ do
    for $v \leftarrow u + 1$ to $L$ do
        $un, vn \leftarrow$ len($NS_u$), len($NS_v$)
        $sim\_scores \leftarrow$ [ ]
        for $a \leftarrow 1$ to $un$ do
            for $b \leftarrow 1$ to $vn$ do
                $e_a \leftarrow$ embed($NS_{u,a}$)
                $e_b \leftarrow$ embed($NS_{v,b}$)
                $score \leftarrow$ cos_sim($e_a, e_b$)
                Append($score$, $sim\_scores$)
            end for
        end for
        $max\_score \leftarrow$ max($sim\_scores$)
        if $max\_score > \delta$ then
            $G_{u,v} \leftarrow 1$
        end if
    end for
end for
```

Algorithm 2 Counting Subgraph Frequency

```
Input: Sentence graph $G$, subgraph size $k$, max
    sentence distance $w$
Output: subgraph set $freq$
$freq \leftarrow \{\}$          ▷ frequency of each subgraph
$nodes \leftarrow G.nodes()$
$i,\ n \leftarrow 0,$ len($nodes$)
while $i < (n - k + 1)$ do
    $w\_n \leftarrow nodes[i : i + w]$    ▷ distance $< w$
    $k\_node\_combs \leftarrow$ combinations($w\_n$, $k$)
    for $k\_nodes$ in $k\_node\_combs$ do
        $subgraph \leftarrow$ extract($G$, $k\_nodes$)
        $signature \leftarrow$ pynauty($subgraph$)
        Add($freq[signature]$, 1)
    end for
    $i \leftarrow i + (w - k + 1)$
end while
```

Algorithm 3 Evaluation

```
Input: Test corpus TC, Doc-subgraph graph $G$,
    Trained GCN
Output: Predictions $preds$
$preds \leftarrow$ []
$N \leftarrow$ len(TC)
for $i \leftarrow 1$ to $N$ do
    $d_i \leftarrow$ TC[$i$]
    $G^* \leftarrow$ Add($d_i$, $G$)          ▷ Add document
    $G^* \leftarrow$ Norm($G^*$)             ▷ Norm graph
    $l_i \leftarrow$ GCN($G^*$)              ▷ Predict label
    Append($l_i$, $preds$)
end for
```

Dataset		Split	#Doc	Avg #W	Max #W	Avg #S
GCDC	Yahoo	Train	1000	157.2	339	7.8
		Test	200	162.7	314	7.8
	Clinton	Train	1000	182.9	346	8.9
		Test	200	186.0	352	8.8
	Enron	Train	1000	185.1	353	9.2
		Test	200	191.1	348	9.3
	Yelp	Train	1000	178.2	347	10.4
		Test	200	179.1	340	10.1
TOEFL	Prompt 1	Total	1656	339.1	806	13.7
	Prompt 2	Total	1562	357.8	770	15.7
	Prompt 3	Total	1396	343.5	731	14.7
	Prompt 4	Total	1509	338.0	699	15.1
	Prompt 5	Total	1648	358.4	876	15.2
	Prompt 6	Total	960	358.3	784	15.3
	Prompt 7	Total	1686	336.6	638	14.0
	Prompt 8	Total	1683	340.9	659	14.7

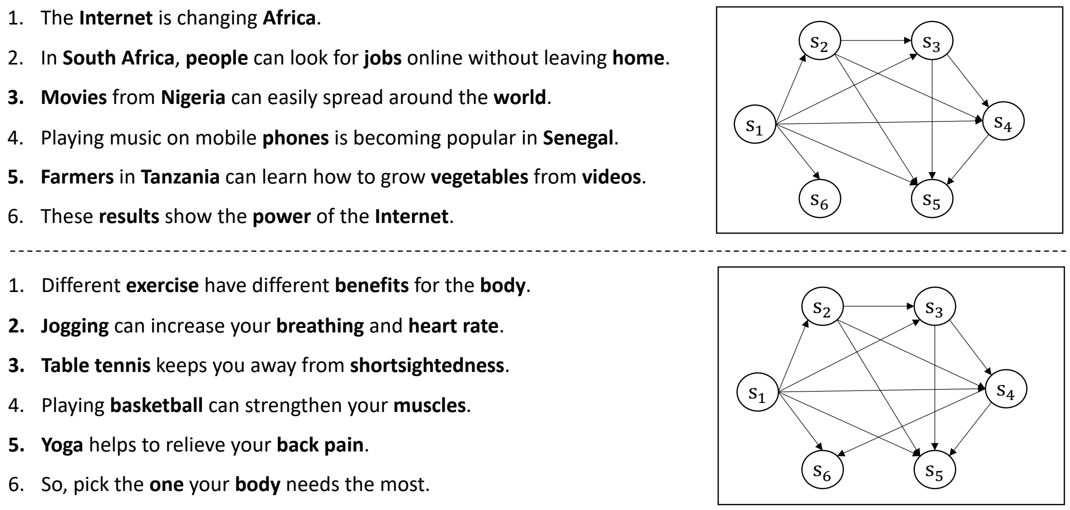
Table 5: The statistics of datasets. #Doc, #W, #S denotes the number of documents, words, sentences.

B Train and Evaluation

Vanilla GCN is a transductive method in which both training and test data are presented to the model during training. This, however, is not applicable in practice since we do not know the evaluation documents in advance. To overcome this drawback, we implement an inductive GCN inspired by the work in fast GCN (Chen et al., 2018). Specifically, we first construct the doc-subgraph graph based on the training corpus (Section 3.3) and train GCN on this graph (Section 3.4). While evaluating, for each document in the test corpus, we add it to the doc-subgraph graph, normalize the adjacency matrix of the new graph, and predict its label (refer to Algorithm 3). Consequently, our method is in a pure inductive setting. That is, our model does not see the test corpus during training, and its evaluation is performed on individual documents without using the information of other samples in the test corpus. Note that when calculating weights for edges between the newly added document node and subgraph nodes, the inverse document frequency we used in equation (1) is the one we computed using only the training corpus.

C Dataset Description

The statistics of the GCDC and TOEFL datasets is shown in Table 5.



1. The **Internet** is changing **Africa**.
2. In **South Africa**, **people** can look for **jobs** online without leaving **home**.
3. **Movies** from **Nigeria** can easily spread around the **world**.
4. Playing music on mobile **phones** is becoming popular in **Senegal**.
5. **Farmers** in **Tanzania** can learn how to grow **vegetables** from **videos**.
6. These **results** show the **power** of the **Internet**.

1. Different **exercise** have different **benefits** for the **body**.
2. **Jogging** can increase your **breathing** and **heart rate**.
3. **Table tennis** keeps you away from **shortsightedness**.
4. Playing **basketball** can strengthen your **muscles**.
5. **Yoga** helps to relieve your **back pain**.
6. So, pick the **one** your **body** needs the most.

Figure 7: An example of two highly coherent texts with similar connection structures. We bold the recognized nouns in the example.

D Detailed Experimental Setting

For the GCDC dataset, we perform 10-fold cross-validation over the training dataset following previous work (Lai and Tetreault, 2018). The dimensionality of the two-layer GCN is set to be 240 for Clinton and Enron domains, and 360 for Yahoo and Yelp domains. We use the Adam optimizer with an initial learning rate of 0.01 on Clinton and Enron, 0.008 on Yahoo and Yelp. As for the TOEFL corpus, we conduct 5-fold cross-validation on the dataset of each prompt, which is the common evaluation setting for the AES task (Taghipour and Ng, 2016). A two-layer GCN with dimension size 240 and the Adam optimizer with an initial learning rate of 0.05 is employed for every prompt dataset. Dropout with a rate of 0.5 is applied to both tasks. And we train the model for 160 epochs on the GCDC dataset and 400 on the TOEFL dataset. For the XLNet+DNN baseline, we configure it with the same trainable parameters as our method. As for Mesgar and Strube (2016), we concatenate the document presentation from XLNet and subgraphs mining from the document's sentence graph as input, and also use a two-layer DNN and a softmax layer. Note that this model has more trainable parameters since its input dimension becomes larger (the concatenation of subgraphs and XLNet representation). For other baseline models, we apply the same experimental setting and XLNet to them as our method and tune their hyperparameters according to the performance on the Dev set. We conduct all experiments on a single Tesla P40 GPU with 24GB memory. It takes about 0.5 days to train our model on the GCDC dataset and 1.5 days on the TOEFL dataset.

We follow previous works (Lai and Tetreault, 2018; Taghipour and Ng, 2016) to use the mean of multi-run accuracy (std) as the evaluation metric.

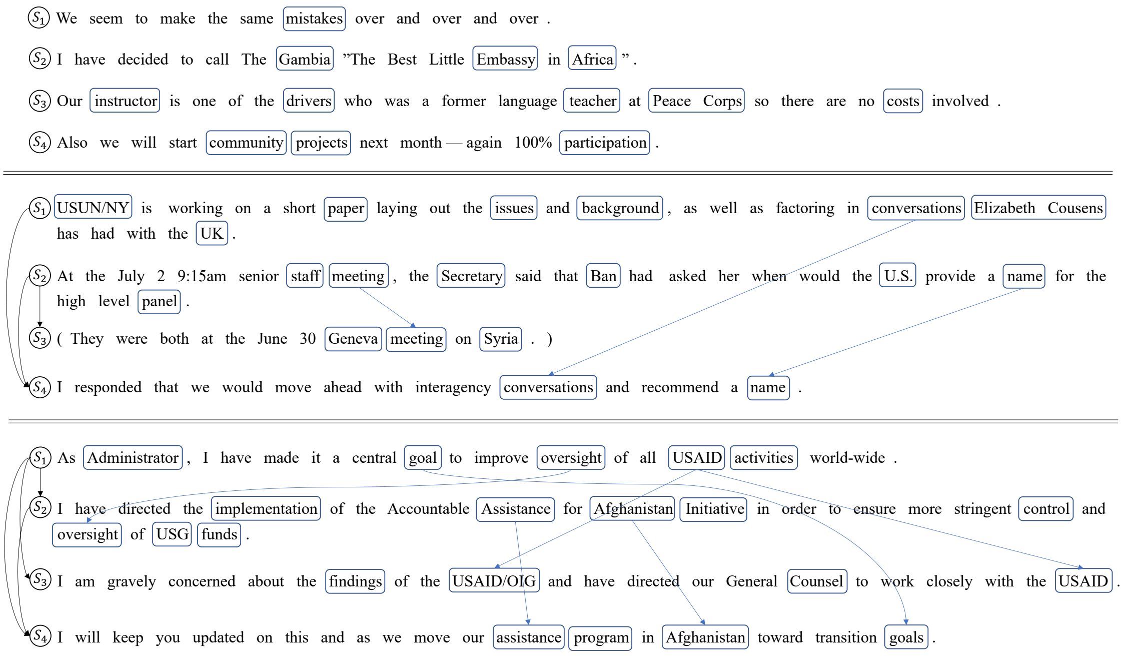
E Examples

E.1 Text Example

Coherence describes how sentences of a text connect to each other (Reinhart, 1980; Foltz et al., 1998; Schwarz, 2001). Theoretically, documents with similar connection structures should tend to have similar degrees of coherence. To help readers understand it, we show two texts in Figure 7. Although the two texts have different content, they share very similar connection structures. For example, the first text first talks about Africa, then discusses specific African countries, and finally makes a conclusion. The second text starts with exercise, then goes to certain daily sports, and finally makes a summary. Based on the linguistic definition of text coherence, the two texts should have a similar degree of coherence due to their similar organizational structures. This could be a very useful prior knowledge when we measure a text's coherence. For example, in Figure 7, we can easily assess the coherence of one text by referring to the label of the other one since they have very similar organizational structures.

E.2 Subgraph Examples

We show several text pieces with constructed subgraphs in Figure 8 (from the GCDC Clinton dataset) and Figure 9 (from the TOEFL P1 dataset). In each example, the corresponding subgraph is



S_1 We seem to make the same mistakes over and over and over .

S_2 I have decided to call The Gambia ”The Best Little Embassy in Africa ” .

S_3 Our instructor is one of the drivers who was a former language teacher at Peace Corps so there are no costs involved .

S_4 Also we will start community projects next month — again 100% participation .

S_1 USUN/NY is working on a short paper laying out the issues and background , as well as factoring in conversations Elizabeth Cousens has had with the UK .

S_2 At the July 2 9:15am senior staff meeting , the Secretary said that Ban had asked her when would the U.S. provide a name for the high level panel .

S_3 (They were both at the June 30 Geneva meeting on Syria .)

S_4 I responded that we would move ahead with interagency conversations and recommend a name .

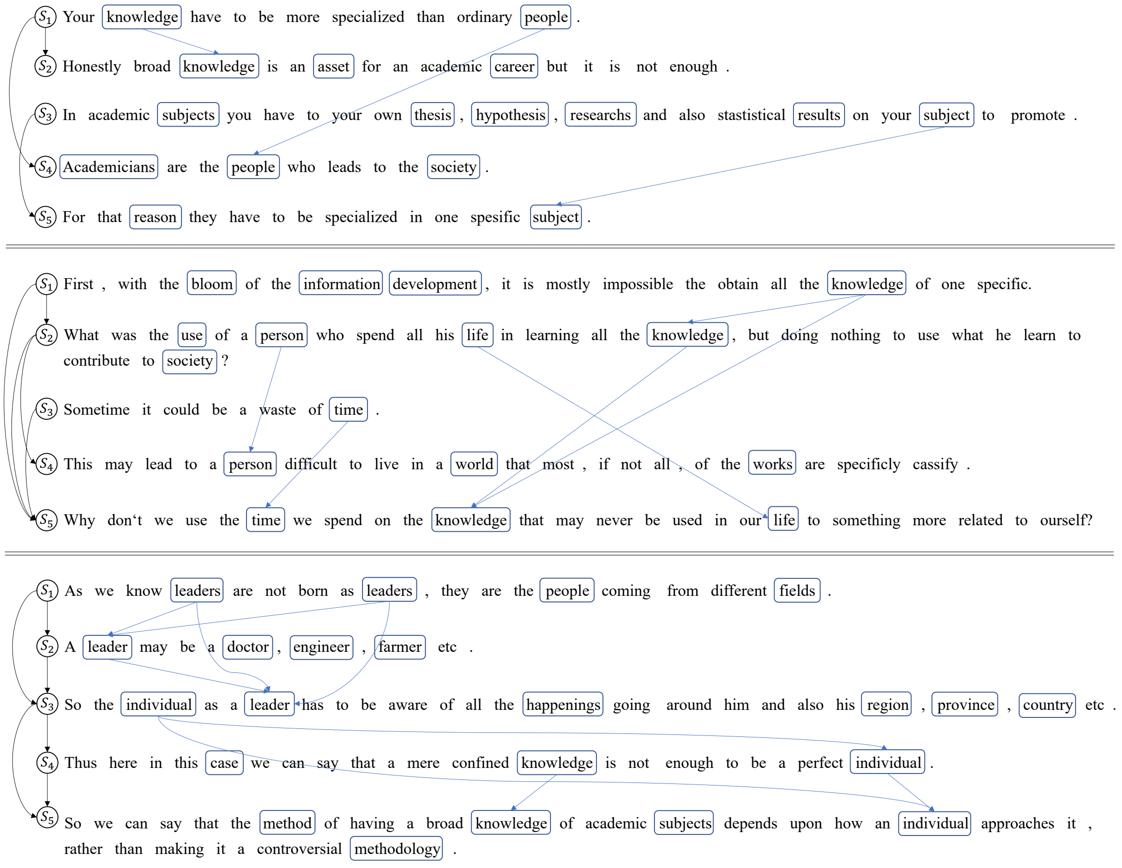
S_1 As Administrator , I have made it a central goal to improve oversight of all USAID activities world-wide .

S_2 I have directed the implementation of the Accountable Assistance for Afghanistan Initiative in order to ensure more stringent control and oversight of USG funds .

S_3 I am gravely concerned about the findings of the USAID/OIG and have directed our General Counsel to work closely with the USAID .

S_4 I will keep you updated on this and as we move our assistance program in Afghanistan toward transition goals .

Figure 8: Three text examples with constructed subgraphs from the GCDC Clinton dataset. We show subgraphs of each text example to the left of that example.



S_1 Your knowledge have to be more specialized than ordinary people .

S_2 Honestly broad knowledge is an asset for an academic career but it is not enough .

S_3 In academic subjects you have to your own thesis , hypothesis , researchs and also stastistical results on your subject to promote .

S_4 Academicians are the people who leads to the society .

S_5 For that reason they have to be specialized in one spesific subject .

S_1 First , with the bloom of the information development , it is mostly impossible the obtain all the knowledge of one specific.

S_2 What was the use of a person who spend all his life in learning all the knowledge , but doing nothing to use what he learn to contribute to society ?

S_3 Sometime it could be a waste of time .

S_4 This may lead to a person difficult to live in a world that most , if not all , of the works are specificly cassify .

S_5 Why don‘t we use the time we spend on the knowledge that may never be used in our life to something more related to ourself?

S_1 As we know leaders are not born as leaders , they are the people coming from different fields .

S_2 A leader may be a doctor , engineer , farmer etc .

S_3 So the individual as a leader has to be aware of all the happenings going around him and also his region , province , country etc .

S_4 Thus here in this case we can say that a mere confined knowledge is not enough to be a perfect individual .

S_5 So we can say that the method of having a broad knowledge of academic subjects depends upon how an individual approaches it , rather than making it a controversial methodology .

Figure 9: Three text examples with constructed subgraphs from the TOEFL Clinton dataset.

shown on the left. We use blue boxes to mark the recognized nouns in each sentence and link semantically related nouns between different sentences by a directed edge between two boxes. Two sentences will be connected if there are semantically related nouns between them.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
In Section 6.

☒ A2. Did you discuss any potential risks of your work?
This is an entirely technical paper. We don't think it has any risk of bias or otherwise.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
In Abstract section and Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

In Section 4.

☑ B1. Did you cite the creators of artifacts you used?
In Section 4.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We are the member of LDC, so we can use those corpora.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Those corpora are created for research purpose.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Those corpora have been widely used in the field for a long time. We don't think it contains any offensive content.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
In Section 4.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
In Appendix C.

C ☑ Did you run computational experiments?

In Section 4.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
In Appendix D.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
In Appendix D.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
In Section 4.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
In Section 3.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.