Centering-based Neural Coherence Modeling with Hierarchical Discourse Segments

Sungho Jeon and **Michael Strube**
Heidelberg Institute for Theoretical Studies gGmbH
{sungho.jeon, michael.strube}@h-its.org

Abstract

Previous neural coherence models have focused on identifying semantic relations between adjacent sentences. However, they do not have the means to exploit structural information. In this work, we propose a coherence model which takes discourse structural information into account without relying on human annotations. We approximate a linguistic theory of coherence, Centering theory, which we use to track the changes of focus between discourse segments. Our model first identifies the focus of each sentence, recognized with regards to the context, and constructs the structural relationship for discourse segments by tracking the changes of the focus. The model then incorporates this structural information into a structure-aware transformer. We evaluate our model on two tasks, automated essay scoring and assessing writing quality. Our results demonstrate that our model, built on top of a pretrained language model, achieves state-of-the-art performance on both tasks. We next statistically examine the identified trees of texts assigned to different quality scores. Finally, we investigate what our model learns in terms of theoretical claims[1].

1 Introduction

Coherence describes the semantic relation between elements of a text. It identifies a text passage as either a unified whole or a collection of unrelated sentences. The most well-known formal theory, Centering theory, determines the most salient item in each sentence, the center or focus, and tracks the changes of the focus (Grosz et al., 1995). Prior studies of coherence have mainly focused on modeling local coherence in Centering theory (Barzilay and Lapata, 2008). They aim to identify the semantic relations between adjacent sentences. However, coherence arises not only at the local level, but also at the document level giving insight into the structure of the discourse.

Discourse structure represents the semantic organization of a text. Incorporating structural information into the model has been beneficial for diverse downstream tasks including text summarization (Marcu, 2000), translation (Guzmán et al., 2014), sentiment analysis (Bhatia et al., 2015), and text classification (Ji and Smith, 2017).

To identify discourse structure, earlier work adopts a supervised approach, relying on human annotations (Hernault et al., 2010; Wang et al., 2017). However, annotating discourse structure is time consuming and costly. It requires annotators to understand not only the local context surrounding the target sentence but also higher level relations. Learning latent structure has been proposed to alleviate this limitation. This approach induces the discourse structure from a text without annotations using an attention layer (Liu and Lapata, 2018). Recent work argues that, however, the learned trees have mostly little to no structure at the document level, and the model relies on specific linguistic cues (Ferracane et al., 2019).

In this paper, we propose a coherence model inspired by Centering theory which takes structural information into consideration. Our model does not rely on human annotations to identify this information. Our model consists of two components: (1) a discourse segments parser which constructs structural relationship for discourse segments by tracking the changes of the focus between discourse segments, and (2) a structure-aware transformer which exploits structural information to update sentence representations.

The discourse segments parser first identifies the hierarchical discourse segments of a text, building upon an approximation of Centering theory (Grosz et al., 1995). This theory first defines three data

[1]Our code is available at: https://github.com/sdeva14/emnlp20-centering-neural-hds

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 7458–7472,
November 16–20, 2020.

structures to describe the focus of a sentence, a list of forward-looking centers (*Cf*), the preferred center (*Cp*), and a single backward-looking center (*Cb*). *Cf* indicates the salient items of the sentence, that are candidates of the focus in the next sentence, and *Cp* indicates the most preferred item of *Cf*. *Cb* describes the focus of a sentence with regards to the previous context. The theory also defines centering transitions to describe the changes of focus by comparing two centers, *Cp* and *Cb*. We propose an algorithm to approximate this theory using a pretrained language model. Our algorithm first identifies the focus of sentences using multi-head attention scores provided by the pretrained language model and semantic similarity between vector representations. Our algorithm then constructs hierarchical discourse segments using a focus stack – inspired by the concept of Grosz and Sidner (1986) – to track the changes of the focus between discourse segments.

Secondly, we propose a structure-aware transformer to account for structural information. Vaswani et al. (2017) introduce the transformer, a model solely based on a self-attention mechanism. This mechanism relates all items to capture semantic relations in a sequence. In contrast, the self-attention of our transformer is restricted to considering sentences with regards to the identified hierarchical discourse segments. We first calculate document structure priors to allow self-attention to relate sentences connected in the identified structure. Then, the document structure attention is calculated by element-wise multiplication of the document structure priors and the self-attention of a naive transformer.

We evaluate our model on two tasks: automated essay scoring (AES) and assessing writing quality (AWQ). AES is the task of assigning a score for a given essay, aiming to replicate human scoring results (Dong and Zhang, 2016). This task has been used to evaluate coherence models (Burstein et al., 2010). Secondly, AWQ is the task of assigning labels of text quality recognized by human annotators. Coherence is one of the most essential aspects of text quality (Feng et al., 2014). We first show that a simple fine-tuned model, relying on a pretrained language model, outperforms the state of the art on both tasks. We then demonstrate that our model achieves state-of-the-art performance on both tasks. Our results indicate that the identified trees let the model assess text quality better by

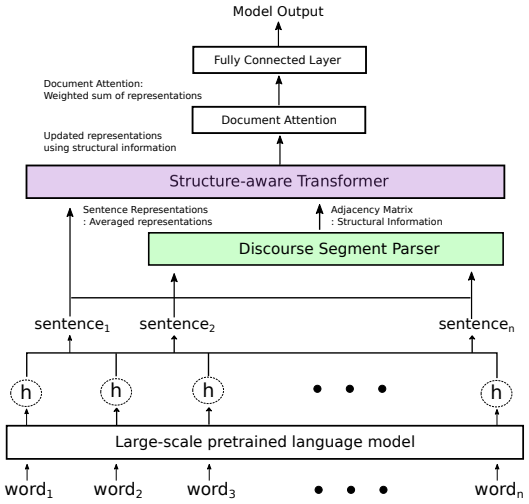


Figure 1: Our model architecture.

structure-aware coherence modeling. We then examine the identified trees to investigate differences of texts in writing quality. We finally inspect identified centers to investigate what our model learns in terms of theoretical claims.

2 Related Work

While unsupervised approaches for discourse parser have been developed (Marcu and Echihabi, 2002; Ji et al., 2015), earlier work mostly adopted a supervised approach to identify discourse structure relying on human annotations. Subba and Di Eugenio (2009) incorporate various linguistic features, including compositional semantics and part-of-speech information, to propose a discourse parser based on Inductive Logic Programming. Hernault et al. (2010) introduce a discourse parser which constructs discourse structure from a full input text. They train classifiers to identify discourse relations, and use them to build a tree structure of an input text. Feng and Hirst (2012) improve the tree building algorithm of this system by incorporating more linguistic features. Wang et al. (2017) introduce an SVM-based model that consists of two stages, one identifying discourse structure, and the other classifying types of relations between units.

More recently, neural models have been developed to recognize discourse structure. Li et al. (2014) present a simple model based on a recursive neural network. Li et al. (2016) claim that this model suffers from a vanishing gradient problem caused by long sequences, and propose an attention-based hierarchical neural network model. To alleviate the shortage of human annotations, Braud

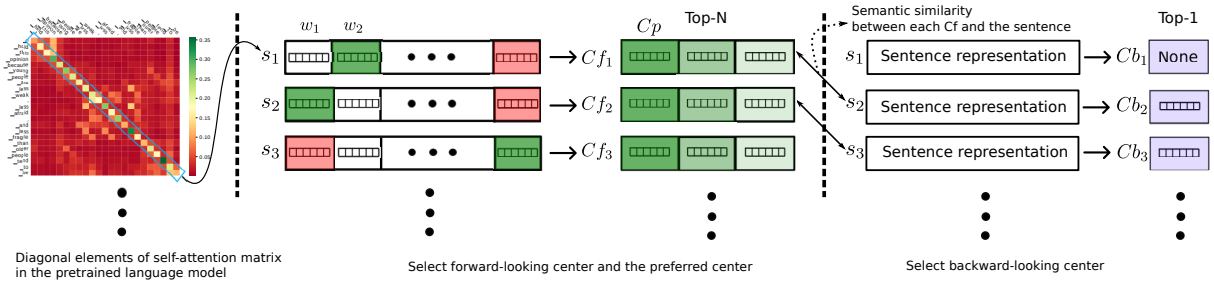


Figure 2: An overview of selecting forward-looking looking centers (*Cf*), preferred centers (*Cp*), and backward-looking centers (*Cb*).

et al. (2016) introduce a simple LSTM-based model which has a multi-view learning architecture. This model uses different views of the same data. Yu et al. (2018) extract syntactical representations by a neural syntax parser, and incorporate them into an RNN-based model.

Previous models of discourse parsing are mostly based on Rhetorical Structure Theory (Mann and Thompson, 1988). This theory represents a document as a tree structure built by connecting discourse units recursively through predefined discourse relations. Another line of work is based on the Penn Discourse Treebank (Webber et al., 2019), which annotates discourse structure in a lexically-grounded approach. These studies represent discourse structure with discourse relations. Unlike these studies, our model does not consider discourse relations but we investigate Centering theory to take structural relationships for discourse segments into account.

A supervised approach requires annotations for each task. To overcome the lack of a labeled dataset, recent work has investigated to learn latent structures, which induce the tree structure directly from a text. While Yogatama et al. (2017) and Choi et al. (2018) induce structure at the sentence level to learn syntax, Liu and Lapata (2018) propose a neural model which induces structural information without a labeled resource. They induce the non-projective dependency structure from a text by structured attention. More recently, however, Ferracane et al. (2019) claim that induced document-level structures do neither match human intuitions nor align with linguistic theories. Unlike latent structure learning, we identify hierarchical discourse segments using a pretrained language model. It lets our model identify the focus of a sentence by comparing semantic similarities between representations of sentences without relying on a resource of manually labeled discourse structure.

3 Our Model

Figure 1 presents the architecture of our coherence model. We first introduce input representations at the sentence level using a pretrained language model. We then describe the algorithm of the discourse segments parser. Finally, we present a structure-aware transformer and the document representation created.

3.1 Sentence Representations

We use a pretrained language model to obtain representations of sentences. In this work, we employ XLNet for the pretrained language model (Yang et al., 2019). XLNet not only outperforms BERT (Devlin et al., 2019), XLNet also has the advantage to model coherence because of its training objective. XLNet maximizes the expected likelihood over all permutations in the training.

We first encode an input document using XLNet to produce word representations. We obtain sentence representations by averaging all word representations in a sentence. We then feed the sentence representations to the discourse parser and the structure-aware transformer.

3.2 Discourse Segment Parser

Our discourse segment parser is inspired by Centering theory Grosz et al. (1995). We modify Centering theory to approximate it in a neural model. The theory considers entities as candidates of centers. To determine centers at the phrase level or the entity level, we would need to incorporate an external parser into the model to identify phrases or entities. The performance of the model then crucially would rely on how accurately the external parser would identify them. Hence, we determine centers at the word level so that our model is not affected by the performance of an external parser.

Figure 2 gives an overview of our approach to identify the focus of sentences. To represent

	$Cb(S_{i-1}) \approx Cb(S_i)$	$Cb(S_{i-1}) \neq Cb(S_i)$
$Cb(S_i) \approx Cp(S_i)$	Continue	Shifting
$Cb(S_i) \neq Cp(S_i)$	Retain	

Table 1: Three types of centering transitions.

the focus of a sentence, we model the backward-looking center and forward-looking centers using scores computed by multi-head self-attention in XLNet. Recent work shows that multi-head attention of a pretrained language model represents important linguistic notions of the input sequence (Clark et al., 2019; Vig and Belinkov, 2019; Sen et al., 2020). It also claims that self-attention might be biased to specialized tokens used in training, <SEP>, <CLS> and the token of a punctuation mark, hence we only consider actual items by filtering these tokens. Following previous work, we use the averaged scores of the multi-head self-attention extracted from the last layer of the model. To identify the salient items of sentences, we encode each sentence separately to identify centers of the sentence.

To determine the forward-looking centers of the sentence at the word level, we extract diagonal elements of the matrix representing multi-head self-attention of the encoded sentence. We then select the top-k vectors obtained by XLNet as the forward-looking centers in the extracted elements. The preferred center of a sentence is the top-1 item in the forward-looking centers. The backward-looking center of a sentence is the item most related to one of the forward-looking centers of the immediately preceding sentence (Brennan et al., 1987). We compare semantic similarity between the averaged word representations of the current sentence and each forward-looking center of the immediately preceding sentence. We use cosine similarity to measure semantic similarity.

Previous work introduces concepts to describe the changes of focus. Grosz et al. (1995) describe three types of centering transitions: *Continue*, *Retain*, and *Shifting*, as shown in Table 1. *Continue* maintains the current focus, and *Retain* intends to change the focus to an item recognized in the current sentence. *Shifting* indicates that the focus is different from the previous sentence. Grosz and Sidner (1986) introduce a focus stack which stores discourse segments related to the current focus.

In this work, we propose an algorithm to construct the hierarchical discourse segments of a text using these concepts (Algorithm 1). For each sentence, we iterate the process until the focus stack is empty or we find a change of the focus. For *Continue*, we add the current sentence to the current segment without changing the stack (line 9-10). For *Retain*, we push the current segment to the stack, which results in connecting the discourse segment of the top item in the stack to the current segment (line 11-13). For *Shifting*, we pop the discourse segment from the stack, and iterate the process for the next sentence (line 16-17). If the process is completed because of an empty stack, then we push s_i as a new segment to process the next sentence (line 20-23). During the process, we build an adjacency matrix to represent the changes of the focus stack. Finally, we connect the adjacent sentences in the discourse segment.

Algorithm 1 The discourse segment parser.

```
1:  procedure PARSER(S, Cb, Cp, t_sim)
2:      Seg ← {} ▷ A list for the current segment
3:      for s_i ← s_1 to s_n do
4:          Seg ← Seg + s_i
5:          while f_stack ≠ ∅ do
6:              sim_{Cb_{i-1},Cb_i} = Sim(Cb_{i-1}, Cb_i)
7:              sim_{Cb_i,Cp_i} = Sim( Cb_i, Cp_i)
8:              if sim_{Cb_i,Cb_{i-1}} > t_sim then
9:                  if sim_{Cp_i,Cb_i} > t_sim then
10:                     isCont ← True
11:                 else
12:                     Push(f_stack, Seg)
13:                     Seg ← {}
14:                 end if
15:                 break              ▷ Exit the loop
16:             else
17:                 Pop(f_stack)
18:                 isCont ← False
19:             end if
20:         end while
21:         if ∼ isCont and f_stack = ∅ then
22:             Push(f_stack, Seg)
23:             Seg ← {}
24:         end if
25:     end for
26:     Adj_Mat = Gen_Ad_Mat(Adj_List)
27:     return Adj_Mat
28: end procedure
```

3.3 Structure-aware Transformer

To take structural information into account, we propose a structure-aware transformer. Our structure-aware transformer is inspired by the Tree-Transformer (Wang et al., 2019), which updates its hidden representations by inducing a tree-structure from a document. The Tree-Transformer generates constituent priors by calculating neighboring attention which represents the probability whether adjacent items are in the same constituent. The constituent priors constrain the self-attention of the transformer to follow the induced structure. Instead of inducing a tree structure, our model uses input structural information to generate document structure priors, which guide the self-attention of the transformer. The sentences which are not connected in the structure are constrained to not attend each other. Document structure priors are then used to calculate structure-aware attention.

We calculate structure-aware attention scores using the identified hierarchical discourse segments. We compute the score $s_{i,j}$ to relate s_i and s_j by the scaled dot-product attention: $s_{i,j} = (q_i^{ds} \cdot k_j^{ds})/d$. We use $(q_i^{ds} \cdot k_j^{ds})$ to represent the semantic relation between s_i and s_j, where q^{ds} is a query matrix and k^{ds} is a key matrix of document structure attention. We represent hierarchical discourse segments by an adjacency matrix. To let the model learn attention with the structural information, we mask scores by the adjacency matrix: $\hat{S} = mask(S, adj)$ where adj is the adjacency matrix representing document structure. We apply a softmax function to each row of the score matrix to represent the probability that s_i attends to other connected sentences: $p_i = softmax(\hat{s}_i)$. To make a symmetric matrix, we calculate the structure-aware attention score: $\hat{a} = \sqrt{p_{i,j} \times p_{j,i}}$. We follow Wang et al. (2019) to cover more relations at the higher level by applying a hierarchical constraint. This restricts a_k^l to be larger than $a_k^l - 1$ for layer l and sentence index k: $a_k^l = a_k^{l-1} + (1 - a_k^{l-1})\hat{a}_k^l$.

We then calculate document structure priors $(D_{i,j})$ using a log-sum instead of multiplication to calculate it efficiently:

$$D_{i,j} = e^{\sum_{k=i}^{j-1} \log(a_k)} \quad (1)$$

Finally, the attention score (E) of the structure-aware transformer is calculated by element-wise multiplication of the document structure priors and the self-attention of a naive transformer:

$$E = D \odot softmax(\frac{QK^T}{d}) \quad (2)$$

where Q is query vectors, K is key vectors with dimension d_k in the naive transformer.

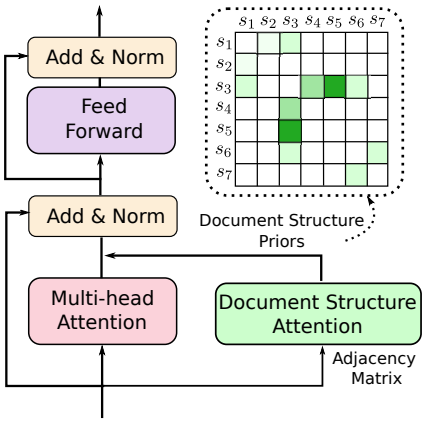


Figure 3: Structure-Aware Transformer.

3.4 Document Representation

In the last layer of our model, we apply document attention to produce the weighted sum of all the updated sentence representations. The document attention identifies relative weights of updated sentence representations which enables our model to handle any document length. Finally, a feed-forward network is applied to the representation to produce the output value.

4 Experiments

4.1 Implementation Details

We implement our model using the PyTorch library and use the Stanford Stanza library[2] for sentence tokenization. We employ XLNet for the pretrained-language model. For the baselines that do not use the pretrained language model, we use Glove for word embeddings, the pretrained word embeddings trained on Google News (Pennington et al., 2014). We set the top-n for selecting *Cf* to 3 and the semantic threshold to compare vector representations to 0.945 (see Appendix B for more training details and parameters).

Due to memory constraints, we encode each sentence separately using XLNet instead of the whole document at once. Our dataset consists of long documents i.e., journal articles with more than 3,000 tokens. For employing the pretrained model, it is

[2]https://stanfordnlp.github.io/stanza/

Model	Prompt								Avg Acc
	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 (2018)	56.25	55.94	55.20	57.20	56.57	55.10	56.97	58.39	56.45
Liu and Lapata (2018)	55.60	55.80	65.60	61.30	57.80	57.50	52.40	52.80	57.80
Averaged-XLNet	70.73	69.48	68.98	67.52	72.35	70.94	70.14	69.01	69.89
XLNet + Wang et al. (2019)	71.65	71.50	71.71	71.64	74.23	69.58	70.76	68.98	71.26
Our Model	**75.10**	**73.35**	**74.75**	**74.18**	**76.38**	**74.30**	**73.61**	**73.44**	**74.39**

Table 2: TOEFL Accuracy performance comparison on the test sets (see Appendix D for more details).

practically infeasible to encode all words in a document at once due to memory limitations. We use 46GB GPU memory of two NVidia P40s for each run.

We re-implemented all baselines to compare on the same deep-learning framework, PyTorch. We then used our re-implementation to report the performance of models with 10 runs with different random seeds. We verified statistical significance (p-value$<$0.01) in both a one-sample t-test, which verifies the reproducibility of the performance of each model, and a two-sample t-test, which verifies that the performance of our model is statistically significant compared to other models. To fulfill the request for fairer comparisons between neural models (Dodge et al., 2019), we also report validation performance and standard deviation of the performance (see Appendix D for more details).

4.2 Baselines

We first compare against the latent learning model for discourse parsing by Liu and Lapata (2018). While their model induces structure at both the sentence level and the document level, we only induce structure at the document level due to memory constraints for large documents. We then compare against a neural coherence model. Mesgar and Strube (2018) propose a local coherence model inspired by Centering Theory. This model finds the two most similar RNN outputs to determine the most salient part of sentences to connect adjacent sentences. This model is evaluated on the AES task as well as the task of assessing readability.

To investigate the influence of a pretrained language model on this task, we implement two models for baselines. We first develop a simple fine-tuned model relying on the pretrained language model (Averaged-XLNet). This simple model encodes an input document at the sentence level and averages the encoded representations. We also implement a second model which combines a state-of-the-art latent tree learning model and the pretrained language model (XLNet+Wang et al. (2019)). This model encodes an input document at the sentence level and updates representations using the Tree-Transformer (Wang et al., 2019). Instead of averaging, document attention is applied to produce a weighted-sum vector representation.

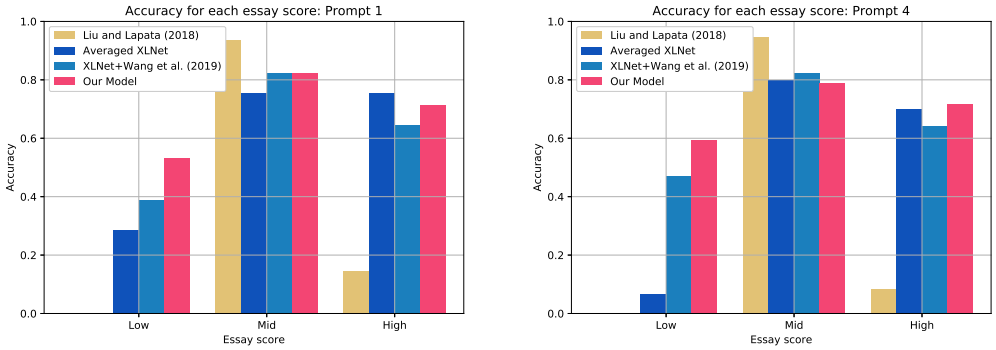
For AES, we also compare against the state of the art for this task. Dong et al. (2017) introduce a model which consists of a convolutional layer followed by a recurrent layer and an attention layer (Bahdanau et al., 2015).

4.3 Automated Essay Scoring

Datasets. To examine the effectiveness of our model on AES, we evaluate our model on the Test of English as a Foreign Language (TOEFL) dataset. TOEFL has overall higher quality of essays compared to essays in the frequently used dataset for AES, the Automated Student Assessment Prize (ASAP) dataset[3]. The prompts in ASAP are written by students in grade levels 7 to 10 of US middle schools. Many essays in ASAP consist of only a few sentences. In contrast, the prompts in TOEFL are submitted for the standard English test for the entrance to universities by non-native students. The prompts in TOEFL do not vary so much, the student population is more controlled, and the essays have a similar length (see Appendix A for more details).

Evaluation Setup. We follow the evaluation setup of previous work on AES (Taghipour and Ng, 2016). For TOEFL, we evaluate performance with accuracy for the three-class classification problem with 5-fold cross-validation. We deploy the cross-entropy loss for training. We use the ADAM optimizer with a learning rate of 0.003. We evaluate performance for 20 epochs on the validation set. The model which reaches the best accuracy on the validation set is then applied to the test set. We use a mini-batch size of 32 with random shuffle.

[3]https://kaggle.com/c/asap-aes/



(a) TOEFL: P1

(b) TOEFL: P4

Figure 4: Accuracy per score in TOEFL (see Appendix D for more details).

	NYT
Liu and Lapata (2018)-reported	82.69 (1.36)
Liu and Lapata (2018)-ours	54.35 (1.00)
Averaged-XLNet	67.53 (3.48)
XLNet+Wang et al. (2019)	71.79 (0.77)
Our Model	**75.12 (1.10)**

Table 3: Mean (standard deviation) accuracy performance of assessing writing quality on the test sets in NYT. We compare the performance of Liu and Lapata (2018), reported in Ferracane et al. (2019) which uses an embedding layer trained on NYT and our implementation which uses a pretrained Glove embedding layer.

Results. Table 2 shows the performance on TOEFL. Dong et al. (2017), the state of the art on AES, show significantly better performance than the model of discourse structure parsing and the neural model of coherence. Interestingly, the simple model relying on the pretrained language model outperforms these three models. XLNet+Wang et al. (2019) then shows better performance. Since we encode a text at the sentence level and not the whole document at once, encoded representations do not include any structural information at the document level. Hence, this indicates that structural information improves the performance of this model compared to Averaged-XLNet. Finally, our model achieves state-of-the-art performance.

To better understand how the model works, we conduct an error analysis. This analysis shows that uneven label distributions cause biased predictions in the model of Liu and Lapata (2018). The TOEFL dataset has an uneven label distribution, 11.0%/54.3%/34.7% for low, mid, and high scores, respectively. In contrast, all models built upon pretrained language models generally predict different scores in an unbiased fashion. XLNet+Wang et al. (2019) shows, however, more bias toward the middle score than Averaged-XLNet. This indicates that, as the model of Liu and Lapata (2018), the baseline model predicts the uneven distribution which leads to better performance. Our model mostly predicts the low and the high score better. This suggests that our model does not take advantage of the uneven distribution but assesses essay quality by modeling coherence.

4.4 Assessing Writing Quality

Datasets. Louis and Nenkova (2013) use a dataset of scientific articles from the New York Times (NYT) for assessing writing quality. They assign each article to one of two classes by a semi-supervised approach: typical or good. Though articles included in both classes are of good quality generally, Louis and Nenkova (2013) show that linguistic features can distinguish different classes of writing quality. Ferracane et al. (2019) use this dataset to evaluate the model of Liu and Lapata (2018).

Evaluation Setup. For NYT, we follow the setup used in previous work. Louis and Nenkova (2013) and Ferracane et al. (2019) undersample the dataset to alleviate the bias of the uneven label distribution. We partition the dataset following Ferracane et al. (2019), into 80% training, 10% validation, and 10% test set, respectively. We use the ADAM optimizer with a learning rate of 0.001. For training, we evaluate performance for 20 epochs and use a mini-batch size of 128 with random shuffle.

Results. We first compare against the state-of-the-art model in latent learning on NYT. Ferracane et al. (2019) show the performance of the latent learning model in Liu and Lapata (2018) on NYT[4].

[4]https://github.com/elisaF/structured

	TOEFL			NYT	
	Low	Mid	High	Typical	Good
Normalized tree height	0.362 (0.190)	0.277 (0.142)	0.242 (0.119)	0.102 (0.051)	0.100 (0.049)
Proportion of leaf nodes	0.149 (0.095)	0.110 (0.073)	0.096 (0.061)	0.037 (0.032)	0.036 (0.031)
Normalized arc length	0.740 (0.279)	0.806 (0.238)	0.846 (0.197)	0.954 (0.058)	0.953 (0.056)
Ratio of small trees	0.0%	0.0%	0.0%	0.0%	0.0%
Proportion of nodes at the top level	0.470 (0.193)	0.505 (0.194)	0.536 (0.187)	0.664 (0.120)	0.660 (0.117)

Table 4: Statistics for learned trees as labels by our model described as mean (standard deviation).

They report the performance of this model with an embedding layer trained on the NYT corpus itself[5]. To ensure fair comparison of the model across different datasets, we use a pretrained Glove embedding layer.

Table 3 reports performance of models on the NYT test set. The model of Liu and Lapata (2018) with the pretrained Glove embedding layer shows significantly lower performance than the same model with the embedding layer trained on NYT. Averaged-XLNet performs better, which shows that employing a pretrained language model is beneficial, and XLNet+Wang et al. (2019) outperforms this model. Our model achieves state-of-the-art performance on NYT among the models using the pretrained embedding layer, but it still shows lower performance than the model using the embedding layer trained on the target corpus. This suggests that linguistic cues have the potential to improve this model further.

4.5 Learned Discourse Structure

We next statistically examine the discourse structure identified by our parser. Ferracane et al. (2019) evaluate the induced structure learned by the model of Liu and Lapata (2018) using four measures: the average height of trees, the proportion of leaf nodes, the normalized arc length, and the ratio of vacuous trees. They define a vacuous tree as a shallow tree whose nodes are connected to the root directly.

We report statistics on the trees identified by our parser as shown in Table 4. We modify two measures, the normalized tree height and the ratio of small trees. We normalize the tree height by the number of nodes to take the length of documents into account. Since there are no vacuous trees in our trees, we report the ratio of small trees, defined as a tree whose normalized tree height is smaller than 0.2 and whose height is smaller than 3. In addition, we report the proportion of the nodes at the top level.

[5]We confirmed this by examining their implementation and emailing the first author.

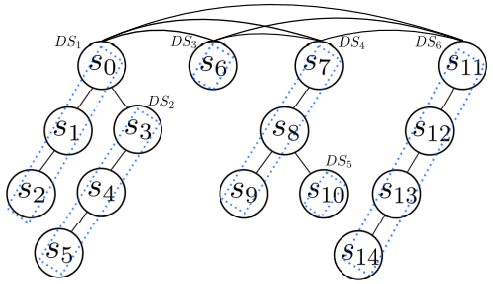


Figure 5: Example of the identified hierarchical discourse segments where DS is a discourse segment and s is a sentence: An essay of high score whose essay-id is 913590 in TOEFL (see Appendix E for more details).

Ferracane et al. (2019) show that trees learned by the model of Liu and Lapata (2018) mostly are vacuous or shallow trees, whose proportion of leaf nodes is greater than 0.9. In contrast, the measures confirm that our model finds differences in the structure of texts of different score levels. The trees are not shallow trees, there is even no small tress, and the proportion of leaf nodes is less than 15%. The normalized arc length is high in our trees, which indicates that there is content connected to the root in the late part of a document. We suspect that this is the result of modeling the changes of focus instead of being biased to the focus captured in the beginning a document.

Figure 5 visualizes an example essay in TOEFL. If texts are scored lower, trees are higher with more leaf nodes, and the proportion of nodes at the top level is lower. In NYT, we observe that the trees are more similar according to the four measures. However, we still observe texts of lower quality in NYT have higher trees and more leaf nodes. These trees are more skewed. This suggests that the focus is more biased to specific content in the texts of lower quality. In our manual examination, we also observe a few cases that texts of lower quality show very shallow trees. This suggests that the focus changes less frequently than in texts of higher quality.

TOEFL-P1 (%)	TOEFL-P5 (%)	NYT-1458761 (%)	NYT-1516415 (%)
_broad (3.63)	_use (2.14)	wyoming (4.44)	_theory (4.03)
_many (1.79)	_twenty (1.79)	colorado (4.44)	_universe (3.22)
_special (1.50)	_cars (1.29)	montana (4.44)	_said (3.23)
i (1.47)	_years (1.20)	ut (2.96)	stan (2.42)
_specialize (1.46)	i (0.99)	_high (2.96)	ein (2.42)
_know (1.05)	_fewer (0.78)	_good (2.22)	dr (2.42)
_specialized (0.99)	_think (0.75)	pi (1.48)	_do (2.42)
_knowledge (0.90)	_car (0.69)	_so (1.48)	_can (1.61)
_academic (0.90)	_today (0.67)	_could (1.48)	_extra (1.61)
_major (0.65)	_number (0.55)	ver (1.48)	_co (1.61)

Table 5: Top-10 most preferred centers (proportions) of essays submitted to the same prompt in TOEFL, a NYT article whose id is 1458761, and a NYT article whose id is 1516415 (see Appendix F for more details).

	T-P1	T-P5	N-14*	N-15*
Prop of "_the" (%)	0.12	0.40	0.00	0.00
Prop of "_a" (%)	0.19	0.18	0.00	0.08
Prop of "_an" (%)	0.04	0.02	0.07	0.00
Prop of "," (%)	0.37	0.40	0.00	0.81
Prop of "_at" (%)	0.03	0.01	0.00	0.00
Prop of "_on" (%)	0.08	0.07	0.00	0.00
Avg prop (%)	0.03	0.03	0.95	1.00
Std prop (%)	0.10	0.07	0.71	0.57

Table 6: Proportion of function words determined as centers in essays submitted to the prompt 1 and 5 in TOEFL (T), a NYT article whose id is 1458761 (N-14*), and a NYT article whose id is 1516415 (N-15*).

4.6 Centering Analysis

We finally inspect the identified centers to investigate what our model learns with regard to the most preferred centers in Centering theory. We explore two questions, (1) whether the identified centers are related to the given topic of a text and (2) whether the centers rely on function words.

While all essays submitted to a prompt in TOEFL have the same topic, articles in NYT have different topics. Hence, we inspect centers at the prompt level in TOEFL and for each document in NYT.

We first examine the proportion of most preferred centers. Table 5 shows that our discourse structure parser indeed identifies centers related to the topic of prompts in TOEFL and to the title of each document in NYT. For instance, the given topic of prompt 1 in TOEFL is "Is it better to have a broad knowledge of many academic subjects than to specialize in one specific subject?", and we observe that preferred centers are related to their topic. However, we also observe a few types of undesirable cases when interpreting centers. The most common case is that the identified centers are related to the topic but also are redundant to other centers. They indicate the same meaning, but they have a different form, such as different tense or grammatical number. Another undesirable case is when centers are subword-level tokens which are produced by subword tokenization deployed in the pretrained language model. It not only makes us difficult to interpret centers intuitively, but also the model might capture a focus different from the author's intention.

We then verify whether our model determines function words as centers. Table 6 shows the proportion of function words determined as centers and the average proportion among all centers. It shows that the proportion of function word is less or comparable to other centers. Hence, this analysis indicates that our model does not exploit function words to capture focus.

5 Conclusions

We propose a neural model of coherence inspired by Centering theory. The intuition is that it describes coherence by tracking the changes of the focus between discourse segments. Our model identifies the hierarchy of discourse segments without human annotations, and incorporates structural information into the model. We demonstrate that the identified hierarchical discourse segments improve performance of the model on two tasks, automated essay scoring and assessing writing quality. Interestingly, we find statistical differences of trees generated from texts of different quality.

Acknowledgments

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

References

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

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

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from RST discourse parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2212–2218, Lisbon, Portugal. Association for Computational Linguistics.

Chloé Braud, Barbara Plank, and Anders Søgaard. 2016. Multi-view and multi-task training of RST discourse parsers. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1903–1913, Osaka, Japan. The COLING 2016 Organizing Committee.

Susan E Brennan, Marilyn W Friedman, and Carl Pollard. 1987. A centering approach to pronouns. In *25th Annual Meeting of the Association for Computational Linguistics*, pages 155–162.

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.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structures. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? An analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. 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 2185–2194, Hong Kong, China. Association for Computational Linguistics.

Fei Dong and Yue Zhang. 2016. Automatic features for essay scoring – an empirical study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1072–1077, Austin, Texas. 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.

Vanessa Wei Feng and Graeme Hirst. 2012. Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 60–68, Jeju Island, Korea. Association for Computational Linguistics.

Vanessa Wei Feng, Ziheng Lin, and Graeme Hirst. 2014. The impact of deep hierarchical discourse structures in the evaluation of text coherence. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 940–949, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Elisa Ferracane, Greg Durrett, Junyi Jessy Li, and Katrin Erk. 2019. Evaluating discourse in structured text representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 646–653, Florence, Italy. 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.

Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Francisco Guzmán, Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2014. Using discourse structure improves machine translation evaluation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 687–698, Baltimore, Maryland. Association for Computational Linguistics.

Hugo Hernault, Helmut Prendinger, Mitsuru Ishizuka, et al. 2010. HILDA: A discourse parser using support vector machine classification. *Dialogue & Discourse*, 1(3).

Yangfeng Ji and Noah A. Smith. 2017. Neural discourse structure for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1005, Vancouver, Canada. Association for Computational Linguistics.

Yangfeng Ji, Gongbo Zhang, and Jacob Eisenstein. 2015. Closing the gap: Domain adaptation from explicit to implicit discourse relations. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2219–2224, Lisbon, Portugal. Association for Computational Linguistics.

Jiwei Li, Rumeng Li, and Eduard Hovy. 2014. Recursive deep models for discourse parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2061–2069, Doha, Qatar. Association for Computational Linguistics.

Qi Li, Tianshi Li, and Baobao Chang. 2016. Discourse parsing with attention-based hierarchical neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 362–371, Austin, Texas. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2018. Learning structured text representations. *Transactions of the Association for Computational Linguistics*, 6:63–75.

Annie Louis and Ani Nenkova. 2013. What makes writing great? First experiments on article quality prediction in the science journalism domain. *Transactions of the Association for Computational Linguistics*, 1:341–352.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory. Toward a functional theory of text organization. *Text*, 8(3):243–281.

Daniel Marcu. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA, USA.

Daniel Marcu and Abdessamad Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 368–375, Philadelphia, Pennsylvania, USA. 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.

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.

Cansu Sen, Thomas Hartvigsen, Biao Yin, Xiangnan Kong, and Elke Rundensteiner. 2020. Human attention maps for text classification: Do humans and neural networks focus on the same words? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4596–4608, Online. Association for Computational Linguistics.

Rajen Subba and Barbara Di Eugenio. 2009. An effective discourse parser that uses rich linguistic information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 566–574, Boulder, Colorado. Association for Computational Linguistics.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.

Yaushian Wang, Hung-Yi Lee, and Yun-Nung Chen. 2019. Tree transformer: Integrating tree structures into self-attention. 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 1061–1070, Hong Kong, China. Association for Computational Linguistics.

Yizhong Wang, Sujian Li, and Houfeng Wang. 2017. A two-stage parsing method for text-level discourse analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 184–188, Vancouver, Canada. Association for Computational Linguistics.

Bonnie Webber, Rashmi Prasad, Alan Lee, and Aravind Joshi. 2019. The Penn discourse treebank 3.0 annotation manual.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, pages 5754–5764.

Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. 2017. Learning to compose words into sentences with reinforcement learning. *In Proceedings of the ICLR Conference*.

Nan Yu, Meishan Zhang, and Guohong Fu. 2018. Transition-based neural RST parsing with implicit syntax features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 559–570, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

A Data Description Details

Table 7 describes statistics on two datasets, TOEFL[6] and NYT[7]. We split a text at the sentence level by Stanford Stanza library, and tokenize them by the XLNet tokenizer. Table 8 describes the topic of each prompt in TOEFL. They are all open-ended tasks, that do not have given context but require students to submit their opinion.

Dataset	#Texts	Avg len (Std)	Max len	Scores
T-P1	1,656	401 (97)	902	1-3
T-P2	1,562	423 (97)	902	1-3
T-P3	1,396	407 (102)	837	1-3
T-P4	1,509	405 (99)	852	1-3
T-P5	1,648	424 (101)	993	1-3
T-P6	960	425 (101)	925	1-3
T-P7	1,686	396 (87)	755	1-3
T-P8	1,683	407 (92)	795	1-3
NYT	8,512	1,841 (1,221)	18,728	1-2

Table 7: Dataset statistics on tokenization: each TOEFL prompt (T-P) and NYT.

B Training and Parameters

For TOEFL, we use a mini-batch size of 32 with random-shuffle. For NYT, we use a mini-batch size of 128 with random-shuffle. For both datasets, we train models with a learning rate of 0.003 and epsilon of 1e-4. We use the ADAM optimizer with a learning rate of 0.003. We evaluate performance for 20 epochs. For the baseline models which do not use a pretrained language model, we use Glove pretrained embeddings with 100-dimensional for TOEFL and with 50-dimensional for NYT. We clip gradients by 1.0 excepts for the latent learning model of discourse parsing. To update sentence representations obtained by a pretrained language model, we use the same dimension of the pretrained language model on a structure-aware transformer. We manually tune hyperparameters.

We use 46GB GPU memory of two NVidia P40s for each run. For training our model, it takes approximately 0.3 days on TOEFL and 11 days on NYT. It takes less processing time to train other two baselines relying on the pretrained language model.

C Scores on Muti-head Attention

Figure 6 visualizes multi-head self-attention scores obtained by XLNet for the example which consists of four sentences as follows. The visualization shows that multi-head self-attention scores capture salient items such as a piano or a home, or linguistic notions such as he or it.

- s_1: Peter wants to play the piano.
- s_2: He went to the piano store to buy one.
- s_3: It was closed.
- s_4: So, he went home.

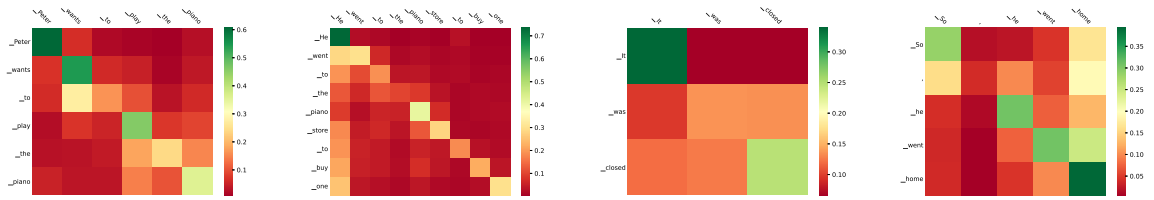
[6]https://catalog.ldc.upenn.edu/LDC2014T06
[7]https://catalog.ldc.upenn.edu/LDC2008T19

Prompt 1	Agree or Disagree: It is better to have broad knowledge of many academic subjects than to specialize in one specific subject.
Prompt 2	Agree or Disagree: Young people enjoy life more than older people do.
Prompt 3	Agree or Disagree: Young people nowadays do not give enough time to helping their communities.
Prompt 4	Agree or Disagree: Most advertisements make products seem much better than they really are.
Prompt 5	Agree or Disagree: In twenty years, there will be fewer cars in use than there are today.
Prompt 6	Agree or Disagree: The best way to travel is in a group led by a tour guide.
Prompt 7	Agree or Disagree: It is more important for students to understand ideas and concepts than it is for them to learn facts.
Prompt 8	Agree or Disagree: Successful people try new things and take risks rather than only doing what they already know how to do well.

Table 8: Topic description: TOEFL.

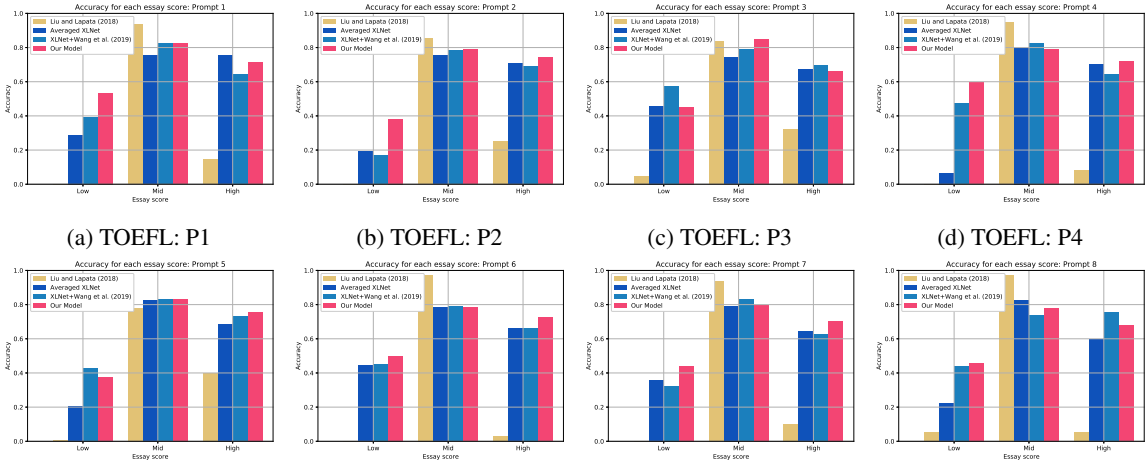
D Experiments Details

We report not only performance of models on test sets, also performance on validation sets, and standard deviation in 10 runs as shown in Table 9-10. These results indicate that our model achieves state-of-the-art performance on both validation sets and test sets. Figure 7 shows the error analysis on TOEFL.



(a) Sentence 1 (b) Sentence 2 (c) Sentence 3 (d) Sentence 4

Figure 6: Multi-head self-attention scores for four sentences, obtained by XLNet.



(a) TOEFL: P1 (b) TOEFL: P2 (c) TOEFL: P3 (d) TOEFL: P4

(e) TOEFL: P5 (f) TOEFL: P6 (g) TOEFL: P7 (h) TOEFL: P8

Figure 7: Accuracy per score in TOEFL.

E Example of a identified structure

Figure 8 visualizes the identified structure from the essay whose score is low. We only present the identified structure due to licensing restrictions of TOEFL.

F Centering Analysis Details

Table 11 shows top-10 most preferred centers in TOEFL and four articles in NYT.

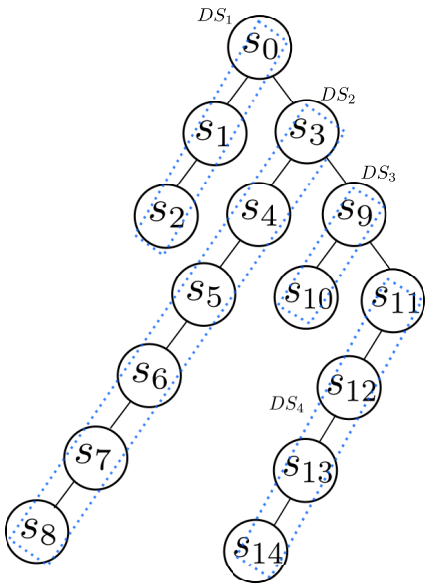


Figure 8: Example of the identified hierarchical discourse segments where DS is a discourse segment and s is a sentence: an essay of low score whose essay-id is 1563434 in TOEFL (see Appendix E for more details).

Model	Prompt								Avg Acc
	1	2	3	4	5	6	7	8	
Liu and Lapata (2018)	55.60	55.80	65.60	61.30	57.80	57.50	52.40	52.80	57.80
	(0.72)	(0.44)	(0.75)	(0.16)	(0.49)	(0.39)	(0.56)	(0.29)	
Averaged-XLNet	69.69	69.99	68.58	66.78	72.01	70.68	68.80	68.59	69.39
	(0.73)	(0.53)	(1.12)	(0.51)	(0.46)	(0.82)	(0.42)	(0.56)	
XLNet + Wang et al. (2019)	71.65	71.50	71.71	71.64	74.23	69.58	70.76	68.98	71.26
	(0.66)	(1.04)	(0.58)	(0.80)	(0.50)	(0.61)	(0.78)	(1.04)	
Our Model	**75.10**	**73.35**	**74.75**	**74.18**	**76.38**	**74.30**	**73.61**	**73.44**	**74.39**
	(0.74)	(0.92)	(0.61)	(1.07)	(0.91)	(1.13)	(0.72)	(1.15)	

Table 9: TOEFL accuracy performance comparison on the test sets, described as mean (std).

Model	Prompt								Avg Acc
	1	2	3	4	5	6	7	8	
Liu and Lapata (2018)	54.97	57.54	54.81	54.08	55.52	54.69	55.19	57.41	55.53
	(0.59)	(0.38)	(0.48)	(0.31)	(0.55)	(0.38)	(0.62)	(0.53)	
Averaged-XLNet	71.06	70.56	67.17	67.02	71.42	69.76	68.54	68.72	69.28
	(0.43)	(0.50)	(0.99)	(0.98)	(0.31)	(0.77)	(0.73)	(0.51)	
XLNet + Wang et al. (2019)	71.44	71.40	71.49	**73.85**	73.86	69.38	70.86	69.67	71.49
	(0.89)	(0.88)	(0.78)	(1.50)	(0.75)	(0.70)	(0.85)	(0.63)	
Our Model	**73.76**	**71.09**	**72.57**	71.86	**73.87**	**71.08**	**71.49**	**71.46**	**72.15**
	(0.74)	(0.92)	(0.61)	(1.07)	(0.91)	(1.13)	(0.72)	(1.15)	

Table 10: TOEFL accuracy performance comparison on the validation sets, described as mean (std).

TOEFL-P1 (%)	TOEFL-P2 (%)	TOEFL-P3 (%)	TOEFL-P4 (%)
_broad (3.63)	_young (5.27)	_young (4.77)	_most (1.44)
_many (1.79)	_enjoy (5.11)	i (1.54)	i (1.43)
_special (1.50)	_older (2.23)	_helping (1.33)	advert (1.22)
i (1.47)	i (1.14)	_help (0.99)	_good (0.87)
_specialize (1.46)	_enjoying (0.82)	_community (0.96)	_advertisement (0.87)
_know (1.05)	_they (0.76)	_communities (0.95)	_advertisements (0.82)
_specialized (0.99)	_younger (0.66)	_time (0.93)	tv (0.73)
_knowledge (0.90)	, (0.53)	_think (0.68)	_seem (0.72)
_academic (0.90)	_people (0.52)	_they (0.64)	_agree (0.70)
_major (0.65)	_more (0.47)	_enough (0.62)	_better (0.70)
TOEFL-P5 (%)	**TOEFL-P6 (%)**	**TOEFL-P7 (%)**	**TOEFL-P8 (%)**
_use (2.14)	_tour (4.43)	_ideas (6.47)	_successful (3.27)
_twenty (1.79)	_guide (3.27)	_learn (1.80)	_succ (1.63)
_cars (1.29)	_best (2.37)	_understand (1.48)	_risk (1.32)
_years (1.20)	_group (2.05)	_understanding (1.48)	i (1.27)
i (0.99)	i (0.99)	_facts (1.37)	_try (1.19)
_fewer (0.78)	_led (1.26)	i (1.30)	_new (1.11)
_think (0.75)	_travel (1.16)	_learning (1.26)	_success (0.98)
_car (0.69)	_good (0.66)	_and (1.08)	_taking (0.80)
_today (0.67)	_alone (0.64)	_concepts (0.91)	_agree (0.70)
_number (0.55)	_traveling (0.55)	_idea (0.86)	_already (0.69)
NYT-1458761 (%)	**NYT-1516415 (%)**	**NYT-1705265 (%)**	**NYT-1254567 (%)**
wyoming (4.44)	_theory (4.03)	_stamp (3.97)	_quantum (4.20)
colorado (4.44)	_universe (3.22)	_prostate (2.65)	ein (4.20)
montana (4.44)	_said (3.23)	_by (2.65)	_led (2.80)
ut (2.96)	stan (2.42)	_say (1.99)	quant (2.10)
_high (2.96)	ein (2.42)	_diet (1.99)	hr (2.10)
_good (2.22)	dr (2.42)	_said (1.99)	_cope (2.10)
pi (1.48)	_do (2.42)	_cancer (1.99)	_computation (2.10)
_so (1.48)	_can (1.61)	ele (1.99)	physicist (1.40)
_could (1.48)	_extra (1.61)	ich (1.32)	_plan (1.40)
ver (1.48)	_co (1.61)	ate (1.32)	ger (1.40)

Table 11: Top-10 most preferred centers (proportions) of essays submitted to the same prompt in TOEFL (see Appendix. A for given topics) and four articles in NYT whose id is 1458761, 1516415, 1705265, and 1254567, respectively. The title of NYT articles are as follows, 1458761: "Among 4 States, a Great Divide in Fortunes", 1516415: "One Cosmic Question, Too Many Answers", 1705265: "Which of These Foods Will Stop Cancer?", and 1254567: "Quantum Theory Tugged, And All of Physics Unraveled".