A Unified Neural Coherence Model

Han Cheol Moon[*][¶]**, Tasnim Mohiuddin**[*][¶]**, Shafiq Joty**[*][¶][§]**, and Xu Chi**[†]

[¶]Nanyang Technological University, Singapore
[§]Salesforce Research Asia, Singapore
[†]A*STAR, Singapore
[¶]{hancheol001@e., mohi0004@e.}ntu.edu.sg
[§]sjoty@salesforce.com
[†]cxu@simtech.a-star.edu.sg

Abstract

Recently, neural approaches to coherence modeling have achieved state-of-the-art results in several evaluation tasks. However, we show that most of these models often fail on harder tasks with more realistic application scenarios. In particular, the existing models underperform on tasks that require the model to be sensitive to local contexts such as candidate ranking in conversational dialogue and in machine translation. In this paper, we propose a unified coherence model that incorporates sentence grammar, inter-sentence coherence relations, and global coherence patterns into a common neural framework. With extensive experiments on local and global discrimination tasks, we demonstrate that our proposed model outperforms existing models by a good margin, and establish a new state-of-the-art.

1 Introduction

Coherence modeling involves building text analysis models that can distinguish a coherent text from incoherent ones. It has been a key problem in discourse analysis with applications in text generation, summarization, and coherence scoring.

Various linguistic theories have been proposed to formulate coherence, some of which have inspired development of many of the existing coherence models. These include the **entity-based** local models (Barzilay and Lapata, 2008; Elsner and Charniak, 2011b) that consider syntactic realization of entities in adjacent sentences, inspired by the Centering Theory (Grosz et al., 1995). Another line of research uses **discourse relations** between sentences to predict local coherence (Pitler and Nenkova, 2008; Lin et al., 2011). These methods are inspired by the discourse structure theories like Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) that formalizes coherence in

terms of discourse relations. Other notable methods include **word co-occurrence** based local models (Soricut and Marcu, 2006), **content** (or topic distribution) based global models (Barzilay and Lee, 2004), and **syntax** based local and global models (Louis and Nenkova, 2012).

With the neural invasion, some of the above traditional models have got their neural versions with much improved performance. For example, Li and Hovy (2014) implicitly model syntax and inter-sentence relations using a neural framework that uses a recurrent (or recursive) layer to encode each sentence and a fully-connected layer with sigmoid activations to estimate coherence probability for every window of three sentences. Li and Jurafsky (2017) incorporate global topic information with an encoder-decoder architecture, which is also capable of generating discourse. Mesgar and Strube (2018) model change patterns of salient semantic information between sentences. Nguyen and Joty (2017); Mohiuddin et al. (2018) propose neural entity grid models using convolutions over distributed representations of entity transitions, and report state-of-the-art results in standard evaluation tasks on the Wall Street Journal (WSJ) corpus.

Traditionally coherence models have been evaluated on two kinds of tasks. The first kind includes synthetic tasks such as *discrimination* and *insertion* that evaluate the models directly based on their ability to identify the right order of the sentences in a text with different levels of difficulty (Barzilay and Lapata, 2008; Elsner and Charniak, 2011b). The latter kind of tasks evaluate the impact of coherence score as another feature in downstream applications like *readability assessment* and *essay scoring* (Barzilay and Lapata, 2008; Mesgar and Strube, 2018).

Although coherence modeling has come a long way in terms of novel models and innovative evaluation tasks (Elsner and Charniak, 2011a; Mohi-

---

*Equal contribution

uddin et al., 2018), it is far from being solved. As we show later, state-of-the-art models often fail on harder tasks like *local discrimination* and *insertion* that ask the model to evaluate a local context (*e.g.,* a 3-sentence window). This task has direct applications in utterance ranking (Lowe et al., 2015) or bot detection[1] in dialogue, and for sentence ordering in summarization.

According to Grosz and Sidner (1986), three factors *collectively* contribute to discourse coherence: (*a*) the organization of discourse segments, (*b*) intention or purpose of the discourse, and (*c*) attention or focused items. The entity-based approaches capture attentional structure, the syntax-based approaches consider intention, and the organizational structure is largely captured by models that consider discourse relations and content (topic) distribution. Although methods like (Elsner et al., 2007; Li and Jurafsky, 2017) attempt to combine these aspects of coherence, to our knowledge, no methods consider all the three aspects together in a single framework.

In this paper, we propose a unified neural model that incorporates sentence grammar (intentional structure), discourse relations, attention and topic structures in a single framework. We use an LSTM sentence encoder with explicit language model loss to capture the syntax. Inter-sentence discourse relations are modeled with a bilinear layer, and a lightweight convolution-pooling is used to capture the attention and topic structures. We evaluate our models on both local and global discrimination tasks on the benchmark dataset. Our results show that our approach outperforms existing methods by a wide margin in both tasks. We have released our code at https://ntunlpsg.github.io/project/coherence/n-coh-emnlp19/ for research purposes.

2 Related Works

Inspired by various linguistic theories of discourse, many coherence models have been proposed. In this section, we give a brief overview of the existing coherence models.

Motivated by the Centering Theory (Grosz et al., 1995), Barzilay and Lapata (2005, 2008) proposed the **entity-based** local model for representing and assessing text coherence, which showed significant improvements in two out of three evaluation tasks. Their model represents a

text by a two-dimensional array called *entity grid* that captures local transitions of discourse entities across sentences as the deciding patterns for assessing coherence. They consider the *salience* of the entities to distinguish between transitions of important entities from unimportant ones, by measuring the occurrence frequency of the entities.

Subsequent studies extended the basic entity grid model. By including non-head nouns as entities in the grid, Elsner and Charniak (2011b) gained significant improvements. They incorporate entity-specific features like named entity, noun class, and modifiers to distinguish between entities of different types, which led to further improvements. Instead of using the transitions of grammatical roles, Lin et al. (2011) model the transitions of **discourse roles** for entities. Feng and Hirst (2012) used the basic entity grid, but improved its learning to rank scheme. Their model learns not only from the original document and its permutations but also from ranking preferences among the permutations themselves.

Guinaudeau and Strube (2013) proposed a **graph-based** unsupervised method for modeling text coherence. Assuming the sentences in a coherent discourse should share the same structural syntactic patterns, Louis and Nenkova (2012) introduced a coherence model based on **syntactic patterns** in text. Their proposed method comprises of local and global coherence model, where the former one captures the co-occurrence of structural features in adjacent sentences and the latter one captures the global structure based on clusters of sentences with similar syntax.

Our model also considers syntactic patterns through a biLSTM sentence encoder that is trained on an explicit language modeling loss. Compared to the entity grid and the syntax-based models, our model does not require any syntactic parser.

With the *neural tsunami*, some of the above traditional models have got their **neural** versions with better performance. Li and Hovy (2014) proposed a neural framework to compute the coherence score of a document by estimating coherence probability for every window of three sentences. Li and Jurafsky (2017) proposed two *encoder-decoder* models, where the first model incorporates global discourse information (e.g., topics) by feeding the output of a sentence-level HMM-LDA model (Gruber et al., 2007) and the second model is trained end-to-end with varia-

---

[1]http://workshop.colips.org/wochat/data/index.html

tional inference. Our proposed model also models inter-sentence relations and global coherence patterns. We use a bi-linear layer to model relations between two consecutive sentences exclusively. Also, our global model implements a light-weight convolution that requires much less parameters, which gives better generalization. Moreover, we train the whole network end-to-end with a window-based adaptive pairwise ranking loss.

Nguyen and Joty (2017) proposed a neural version of the entity grid model where they first transform the grammatical roles in a grid into their distributed representations. Then they employ a convolution operation over it to model entity transitions in the distributed space. Finally, they compute the coherence score from the convoluted features by a spatial max-pooling operation. The model is trained with a **document-level** (global) pairwise ranking loss. Mohiuddin et al. (2018) improve the neural entity grid model by *lexicalizing* its entity transitions They use *off-the-shelf* word embeddings to achieve better generalization with the lexicalized model. As we will demonstrate, because of the spatial-pooling operation, entity-based neural models are not sensitive to mismatch of **local patterns** in a document limiting their applicability to tasks that require local discrimination. Another crucial limitation of employing a **document-level** pairwise ranking loss is that the loss from the document-level permutation for a negative document may **penalize the convolution kernel weights** even if the local permutation matches that of the positive document. In contrast, we apply a window-level (local) **adaptive pairwise ranking loss** that gets activated only if the corresponding windows of the positive and negative documents differ. This way our model is sensitive to local patterns without penalizing the weights unfairly. We capture global patterns using a separate light-weight convolution module.

3 Proposed Model

Let $D = (\mathbf{s}_1, \cdots, \mathbf{s}_n)$ be a document consisting of n sentences. Our goal is to assess its coherence score. Figure 1 provides an overview of our proposed unified coherence model. It has four components in a Siamese architecture (Bromley et al., 1993): (*i*) a sentence encoder (§3.1), (*ii*) a local coherence model (§3.2), (*iii*) a global coherence model (§3.3), and (*iv*) a coherence scoring layer (§3.4). For encoding a sentence, we first map each

word of the sentence to its corresponding vector representation. We then use a bidirectional LSTM sentence encoder with explicit **language model loss** to capture the sentence grammar. Given the sentence representations, the local and global coherence model extract the respective coherence features. The local coherence model implements a bilinear layer to model inter-sentence discourse relations. This layer captures the **local contexts** of the document. To capture the attention (entity distribution) and topic structures, i.e., the **global coherence** of the document, our global coherence model uses a *light-weight convolution* (Wu et al., 2019) with average pooling. The coherence scoring is a linear layer that evaluates the coherence from the extracted features. The whole architecture is trained *end-to-end* with a pairwise ranking loss. In the following, we elaborate on different components of our proposed model.

3.1 Modeling Intention

A discourse has a purpose such as describing an event, explaining some results, evaluating a product, etc. As such, sentences in the discourse should support the purpose as a whole. The syntactic structure of the sentence can be used to model the intent structure (Louis and Nenkova, 2012). We use a bidirectional long short-term memory or bi-LSTM (Hochreiter and Schmidhuber, 1997) to encode each sentence into a vector representation while modeling its compositional structure.

For an input sentence $\mathbf{s}_i = (w_1, \cdots, w_m)$ of length m, we first map each word w_t to its corresponding vector representation $\mathbf{e}_t \in \mathbb{R}^d$, where d is the dimension of the word embedding. The LSTM recurrent layer then computes a compositional representation $\mathbf{h}_t \in \mathbb{R}^p$ at every time step t by performing nonlinear transformations of the current time step word vector representation $\mathbf{e}_t$ and the output of the previous time step $\mathbf{h}_{t-1}$, where p is the number of features in the LSTM hidden state. The output of the last time step $\mathbf{h}_m$ is considered as the representation of the sentence. A bi-LSTM processes a given sentence $\mathbf{s}_i$ in two directions: from *left-to-right* and *right-to-left*, yielding a representation $\mathbf{h}_i = [\overrightarrow{\mathbf{h}_m}; \overleftarrow{\mathbf{h}_m}] \in \mathbb{R}^{2p}$, where ';' denotes concatenation.

We train our sentence encoder with an explicit **language model loss**. A bidirectional language model combines a forward and a backward language model (LM). Similar to Peters et al. (2018),
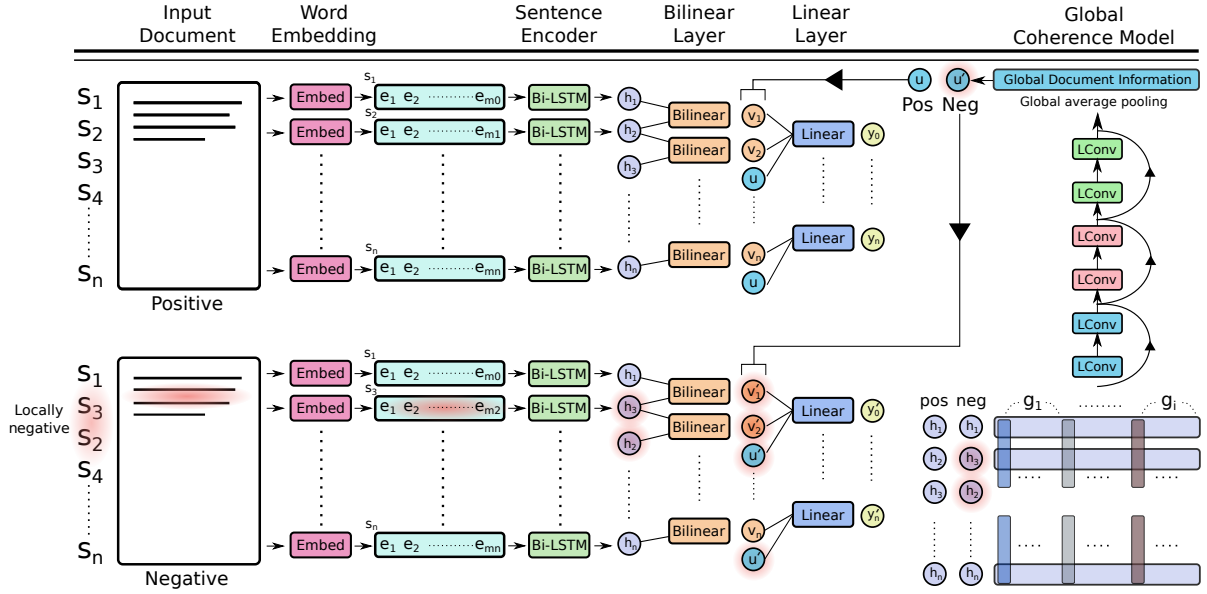
Figure 1: An overview of the proposed coherence model (best viewed in color). The superscript '′' above output of each component denotes negative outputs and the red shade represents incoherent portions in the document. Note that all network parameters and components are shared regardless of the input documents.

we jointly minimize the *negative log-likelihood* of the forward and backward directions:

$$\mathcal{L}_{lm} = -\sum_{j=1}^{m} \Bigg(\log p(w_j|w_1, \cdots, w_{j-1}; \theta, \overrightarrow{\theta}_{\text{lstm}}) +$$
$$\log p(w_j|w_{j+1}, \cdots, w_m; \theta, \overleftarrow{\theta}_{\text{lstm}}) \Bigg) (1)$$

where $\overrightarrow{\theta}_{\text{lstm}}$ and $\overleftarrow{\theta}_{\text{lstm}}$ are the parameters of the forward and backward LSTMs, and θ denote the rest of the parameters which are shared.

3.2 Modeling Inter-Sentence Relation

Discourse relations between sentences reflect the organizational structure of a discourse that can be used to evaluate the coherence of a text (Lin et al., 2011; Li and Hovy, 2014). To model inter-sentence discourse relations, we use a bilinear model. Our bi-LSTM sentence encoder gives a representation $\mathbf{h}_i \in \mathbb{R}^{2p}$ for each sentence $\mathbf{s}_i$ in the document. We feed the representations of every two consecutive sentences $(\mathbf{h}_i, \mathbf{h}_{i+1})$ to this layer, which applies a bilinear transformation as:

$$\mathbf{v}_i = \mathbf{h}_i^T \mathbf{W}_b \mathbf{h}_{i+1} + \mathbf{b} \qquad (2)$$

where $\mathbf{W}_b \in \mathbb{R}^{q \times 2p \times 2p}$ is a learnable tensor, and $\mathbf{b} \in \mathbb{R}^q$ is a learnable bias vector. Here q is the number of output features (i.e., $\mathbf{v}_i \in \mathbb{R}^q$).

3.3 Modeling Global Coherence Patterns

The model proposed so far captures only local information. However, global discourse phenomena like entity or topic distributions are also important for coherence evaluation (Barzilay and Lee, 2004; Elsner et al., 2007; Louis and Nenkova, 2012). Global coherence is modeled in our architecture by a *convolution-pooling* mechanism.

As shown in the Figure 1, our **global coherence** sub-module takes the representations $\mathbf{H} = (\mathbf{h}_1, \cdots, \mathbf{h}_n)$ of all the sentences in a document generated by the bi-LSTM encoder. The module uses six convolution layers with residual connections, followed by an average pooling layer. Instead of using regular convolutions, we use *lightweight convolution* (Wu et al., 2019), which is built upon *depth-wise* convolution (Chollet, 2016). **Depth-wise convolutions** perform a convolution independently over every input channel which significantly reduces the number of parameters as shown in Figure 2. For a given input $\mathbf{H} \in \mathbb{R}^{n \times d}$, the output $\mathbf{O} \in \mathbb{R}^{n \times d}$ of the depth-wise convolution with convolution weight $\mathbf{W} \in \mathbb{R}^{d \times k}$ with kernel size k for element i and output dimension c can be written as:

$$\mathbf{O}_{i,c} = \text{Dcon}(\mathbf{H}, \mathbf{W}_{c,:}, i, c)$$
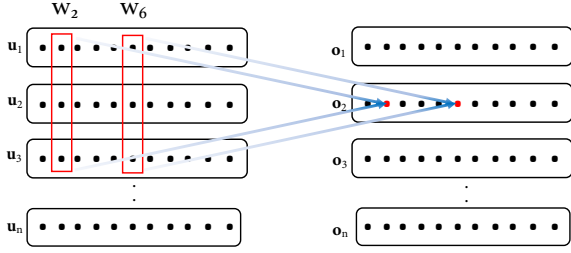$$= \sum_{j=1}^{k} \mathbf{W}_{c,j} \mathbf{H}_{(i+j-\lceil \frac{k+1}{2} \rceil),c} \qquad (3)$$

2265

Figure 2: Depth-wise convolution for kernel size $k = 3$. The convolutions are done over the input dimensions.

Compared to the regular convolutions, depth-wise convolutions reduces the number of parameters from $d^2 k$ to dk (note that $d = 2p$ in our case).

Light-weight Convolutions make the depth-wise convolution even simpler by sharing groups of output channels and normalizing weights across the temporal dimension using a softmax. It has a fixed context window which determines the importance of context elements with a set of weights that do not change over time steps. For the i-th element in the sequence and output channel c, *light-weight convolution* computes:

$$\text{Lcon}(\mathbf{H}, \mathbf{W}_{g,:}, i, c) = \text{Dcon}(\mathbf{H}, \text{softmax}(\mathbf{W}_{g,:}), i, c) \quad (4)$$

where $g = \lceil \frac{cG}{d} \rceil$ with G being the number of groups. The number of parameters with light-weight convolutions reduces to $H.k$. Wu et al. (2019) show that models equipped with light-weight convolution exhibit better generalization compared to regular convolutions. It is indeed crucial in our case since we use convolutions to model a document, with large kernel size it would be difficult to learn from small datasets compared to (sentence-level) machine translation datasets.

The light-weight convolution layers generate d feature maps $\mathbf{f}^i \in \mathbb{R}^n, i = 1, ..., d$ for each input document by the convolutional operation over an input dimension across all the sentences in a document. Subsequently, **global average pooling** is performed over the extracted feature maps to achieve a global view of the input document. The achieved global feature $\mathbf{u} \in \mathbb{R}^d$ can be expressed as follows:

$$\mathbf{u} = \frac{1}{n}[\mathbf{1} \cdot \mathbf{f}^1, ..., \mathbf{1} \cdot \mathbf{f}^d] \quad (5)$$

where $\mathbf{1} \in \mathbb{R}^n$ is the vector of ones and n is the number of sentences in an input document. The

global document level features $\mathbf{u}$ are then concatenated with the local features of each consecutive sentence pair $(\mathbf{v}_i; \mathbf{v}_{i+1})$ in the document, i.e., the output of the bilinear layer (see Figure 1).

$$\mathbf{z}_i = [\mathbf{v}_i; \mathbf{v}_{i+1}; \mathbf{u}] \quad (6)$$

where $\mathbf{z}_i \in \mathbb{R}^{2q+2p}$ and ';' denotes concatenation.

3.4 Coherence Scoring

We then feed the concatenated global and local features $\mathbf{z}_i$ to the final linear layer of our model to compute the coherence score $y_i \in \mathbb{R}^n$ for each local window.

$$y_i = \mathbf{z}_i^T \mathbf{w}_l + b_{\text{l}} \quad (7)$$

where $\mathbf{w}_l$ is weight vector and b_l is a bias. The final decision on documents is made by summing up all local scores of documents and compares the summed scores.

3.5 Overall Objective and Training Details

Our model assigns a coherence score y_i to every possible local window D^ℓ in the document D, where ℓ is the *local window* index. During implementation, the input document is *padded*, so that the number of possible local window is the same as the number of sentences (n) in the document D.

Let $\mathbf{y} = \Omega(D|\Theta)$ define our model that produces the coherence scores $\mathbf{y} = (y_1, ..., y_n)$ for an input document D, with Θ being the parameters. We use a *window-level pairwise ranking approach* (Collobert et al., 2011) to learn Θ.

Our training set contains *ordered* pairs of documents $(D_{\text{pos}}, D_{\text{neg}})$, where document D_{pos} exhibits a higher degree of coherence than document D_{neg}. See Section 4 for details about the dataset. We seek to learn Θ that assigns higher coherence scores to D_{pos} than to D_{neg}. We observed that the naive pairwise ranking loss that uses a fixed margin unfairly penalizes the locally positive sentences during training. In other words, the loss should be active only for local windows that differ in D_{pos} and D_{neg}. To address this, we propose to use an **adaptive pairwise ranking loss** $\mathcal{L}_\Theta$ defined as follows.

$$\mathcal{L}_\Theta = \frac{1}{n} \sum_{\ell=1}^{n} \max \Big\{ 0, \phi(D_{\text{pos}}^\ell, D_{\text{neg}}^\ell) -$$
$$\Omega(D_{\text{pos}}^\ell|\Theta) + \Omega(D_{\text{neg}}^\ell|\Theta) \Big\} \quad (8)$$

where $\phi(D_{\text{pos}}^{\ell}, D_{\text{neg}}^{\ell})$ is an adaptive margin given by

$$\phi(D_{\text{pos}}^{\ell}, D_{\text{neg}}^{\ell}) = \begin{cases} 0 & \text{if } D_{\text{pos}}^{\ell} = D_{\text{neg}}^{\ell} \\ \tau & \text{otherwise} \end{cases}$$

where τ is a margin constant.

Our total loss, $\mathcal{L}_{\Theta} = \mathcal{L}_{\Theta} + \mathcal{L}_{lm}$.

Note that our model shares all the layers and components, i.e., Θ to obtain $\Omega(D_{\text{pos}}|\Theta)$ and $\Omega(D_{\text{neg}}|\Theta)$ from a pair of document $(D_{\text{pos}}, D_{\text{neg}})$. Therefore, once trained, it can be used to score any input document independently.

4 Evaluation Tasks and Datasets

For comparison purposes with previous work, we evaluate our models on the standard **"global" discrimination** task (Barzilay and Lapata, 2008), where a document is compared to a *random permutation* of its sentences, which is considered to be incoherent. We also evaluate on an *inverse discrimination* task (Mohiuddin et al., 2018), where the sentences of the original document are placed in the reverse order to create the incoherent document. Similar to them, we do not train our models explicitly on this task, rather we use the trained model from the standard discrimination task. In addition and more importantly, we evaluate the models on a more challenging **"local" discrimination** task, where two documents differ only in a local context (*e.g.,* a 3-sentence window), as shown with an example in Figure 3.

Dataset for Global Discrimination. We follow the same experimental setting of the WSJ news dataset as used in previous works (Mohiuddin et al., 2018; Nguyen and Joty, 2017; Elsner and Charniak, 2011b; Feng et al., 2014). Similar to them, we use 20 random permutations of each document for both training and testing, and exclude permutations that match the original one. Table 1 summarizes the data sets used in global discrimination task. We randomly selected 10% of the training set for development purposes.

Dataset for Local Discrimination. We use the same WSJ articles used in the global discrimination task (Table 1) to create our local discrimination datasets. Sentences inside a local **window of size 3** are re-ordered to form a locally incoherent text. Only articles with **more than 10 sentences** are included in our dataset. This gives 748 documents for training and 618 for testing.

	Sections	# Doc.	# Pairs
Train	00-13	1,378	26,422
Test	14-24	1,053	20,411

Table 1: Statistics of the WSJ news dataset used for **"global" discrimination** task.

	Sections	# Doc.	# Pairs			
			$\mathcal{D}_{w=1}$	$\mathcal{D}_{w=2}$	$\mathcal{D}_{w=3}$	$\mathcal{D}_{w=1,2,3}$
Train	00-13	748	7,890	12,280	12,440	32,610
Test	14-24	618	6,568	9,936	9,906	26,410

Table 2: Statistics on the WSJ news dataset used for **"local" discrimination** task. The w denotes the number of permuted local windows in a document.

We first set w as the number of local windows that we want to permute in a document. Based on this, we create four datasets for our local discrimination task: $\mathcal{D}_{w=1}$, $\mathcal{D}_{w=2}$, $\mathcal{D}_{w=3}$ and $\mathcal{D}_{w=1,2,3}$. $\mathcal{D}_{w=1}$ contains the documents, where only one randomly selected window is permuted. Similarly, $\mathcal{D}_{w=2}$ contains the documents, where two randomly selected windows are permuted. $\mathcal{D}_{w=3}$ is similarly created for 3 windows. $\mathcal{D}_{w=1,2,3}$ denotes the concatenated datasets. The number of negative documents for each article was restricted **not to exceed 20 samples**. Additionally, we exclude the cases of the overlap between windows. In other words, the sentences are allowed to be permuted only inside their respective window.

We randomly select 10% of the training set for development purposes. Table 2 summarizes the datasets. Consequently, the training and the test dataset for $\mathcal{D}_{w=1,2,3}$ consists of 32,610 and 26,410 pairs, respectively.

5 Experiments

This section presents details of our experiment procedures and results.

5.1 Models Compared

We compare our proposed unified coherence model with several existing models. Some of the baselines that are not publicly available were re-implemented during experiments, otherwise we conducted experiments with publicly available codes, and the rest of the reported results are from their original papers. In the following sections, we present brief descriptions of the existing models.

s_0 | "The House voted to boost the federal minimum wage for the first time since early 1981 , casting a solid 382-37 vote for a compromise measure backed by President Bush."

s_1 | "The vote came after a debate replete with complaints from both proponents and critics of a substantial increase in the wage floor."

s_2 | "Advocates said the 90-cent-an-hour rise , to $ 4.25 an hour by April 1991 , is too small for the working poor , while opponents argued that the increase will still hurt small business and cost many thousands of jobs."

s_3 | "But the legislation reflected a compromise agreed to on Tuesday by President Bush and Democratic leaders in Congress , after congressional Republicans urged the White House to bend a bit from its previous resistance to compromise."

s_4 | "So both sides accepted the compromise , which would lead to the first lifting of the minimum wage since a four-year law was enacted in 1977 , raising the wage to $ 3.35 an hour from $ 2.65."

(a) Positive document

s_2 | "Advocates said the 90-cent-an-hour rise , to $ 4.25 an hour by April 1991 , is too small for the working poor , while opponents argued that the increase will still hurt small business and cost many thousands of jobs."

s_1 | "The vote came after a debate replete with complaints from both proponents and critics of a substantial increase in the wage floor."

s_0 | "The House voted to boost the federal minimum wage for the first time since early 1981 , casting a solid 382-37 vote for a compromise measure backed by President Bush."

s_3 | "But the legislation reflected a compromise agreed to on Tuesday by President Bush and Democratic leaders in Congress , after congressional Republicans urged the White House to bend a bit from its previous resistance to compromise."

s_4 | "So both sides accepted the compromise , which would lead to the first lifting of the minimum wage since a four-year law was enacted in 1977 , raising the wage to $ 3.35 an hour from $ 2.65."

(b) Negative document

Figure 3: Sample data in the local permutation data set. (a) is the positive document, WSJ0098, and (b) is the negative version of the positive document, which is locally sentence-order permuted.

Distributed Sentence Model (L&H). This is the neural model proposed by Li and Hovy (2014). Similar to our local model, it extracts local coherence features for small windows of sentences to compute the coherence score of a document. First, they use a recurrent or a recursive neural network to compute the representation for each sentence in the local window from its words and their pre-trained embeddings. Then the concatenated vector is passed through a non-linear hidden layer, and finally the output layer decides if the window of sentences is a coherent text or not. The main differences between our implementation and the implementation referred in their paper are that we used a bi-LSTM (as opposed to simple RNN) for sentence encoding and trained the network with the Adam optimizer (as opposed to AdaGrad).

Grid-all nouns & Extended grid (E&C)[2]. Elsner and Charniak (2011b) report significant gains by including all nouns as entities in the original entity grid model as opposed to considering only head nouns. In their extended grid model, they used 9 additional entity-specific features, 4 of which are computed from external corpora.

Neural Grid & Ext. Neural Grid (N&J)[3]. These are the neural versions of the entity grid models as proposed by (Tien Nguyen and Joty, 2017). They use convolutions over grammatical roles to model entity transitions in the distributed space. In the extended model, they incorporate three entity-specific features.

Lex. Neural Grid (M&J)[4]. Mohiuddin et al. (2018) improved the neural grid model by lexicalizing the entity transitions. Experiment results for this model were obtained with the optimal setting described in the original paper.

Global Coherence Model. This is the global coherence model component in our proposed unified model as described in Section 3.3. The model extracts document-level features through lightweight convolutions. The extracted features are subsequently averaged along the temporal dimension, which is in turn used in a linear layer for coherence scoring. This model used a kernel size of 5 and each document was padded by the size of 3.

5.2 Settings of Our Model

We held out 10% of the training documents to form a development set (DEV) on which we tune the hyper-parameters of our models. In our experiments, we use both word2vec (Mikolov et al., 2013) and ELMo (Peters et al., 2018) for the distributed representations of the words. Unlike word2vec, ELMo is capable of capturing both subword information and contextual clues. We implemented our models in PyTorch framework on a Linux machine with a single GTX 1080 Ti GPU.

During training, for optimization we use Adam optimizer (Kingma and Ba, 2015) with L_2 regularization (0.00001 regularization parameter). We trained the model up to 25 epochs to make the models' performance converge. To search for optimal parameters, we conducted various experiments while varying the hyper-parameters. Precisely, minibatch size in {5, 10, 20, 25}, sentence embedding size in {128, 256}, lightweight convolution kernel size in {3, 5, 7, 9}, bilinear output dimension size in {32, 64} are investigated. We present the optimal hyper-parameter values in the *supplementary document*. The results are reported by averaging over five different runs of the model with different seeds for statistical stability.

Model	Emb.	$\mathcal{D}_{w=1,2,3}$	$\mathcal{D}_{w=1}$	$\mathcal{D}_{w=2}$	$\mathcal{D}_{w=3}$
Lex. Neural Grid (M&J)*	word2vec	60.27	56.11	60.23	62.23
Lex. Neural Grid (M&J)	word2vec	55.01	53.81	55.37	56.16
Dist. sentence (L&H)	word2vec	6.76	4.28	6.82	9.25
Our Global Model	word2vec	57.24	53.35	56.58	59.67
Our Local Model	word2vec	73.23	66.21	73.16	77.93
Our Local Model	ELMo	74.12	65.82	73.54	78.16
Our Full Model	word2vec	75.37	**67.29**	75.58	80.21
Our Full Model	ELMo	**77.07**	64.38	**76.12**	**81.23**

Table 3: Results in accuracy on the **Local Discrimination task**. * is a pre-trained model on the global discrimination task.

	Model	Emb.	Standard	Inverse
I	Dist. sentence (L&H)	word2vec	17.39	18.11
II	Grid-all nouns (E&C)	-	81.60	75.78
	Ext. Grid (E&C)	-	84.95	80.34
III	Neural Grid (N&J)	Random	84.36	83.94
	Ext. Neural Grid (N&J)	Random	85.93	83.00
IV	Lex. Neural Grid (M&J)	Random	87.03	86.88
	Lex. Neural Grid (M&J)(paper)	word2vec	88.56	88.23
	Lex. Neural Grid (M&J)(code)	word2vec	88.51	88.13
V	Our Best Model	word2vec	90.42	95.27
	Our Best Model	ELMo	**93.19**	**96.78**

Table 4: Results in accuracy on the **Global Discrimination** task.

5.3 Results on Local Discrimination

Table 3 shows the results in accuracy on the **"local" discrimination** task. From the table, we see that existing models including our global model perform poorly compared to our proposed local models. They are likely to fail in distinguishing the text segments that are locally coherent and penalize them unfairly. One of the possible explanations of this phenomenon can be found in the nature of the global model. These models (except L&H) are designed to make a decision at a global level, thus they are likely to penalize locally coherent segments of a text. This observation is further bolstered by the performance of our local coherence models, which show higher sensitivity in discriminating locally coherent texts and achieve significantly higher accuracy compared to the baseline models and our global model.

Another aspect to notice here is that the performance of all the models become gradually better with the increase in the number of permutation windows in the dataset. This is not surprising because in the datasets with a lower number of permutation windows, the difference between a positive and a negative document is very subtle. For example, in $\mathcal{D}_{w=1}$ dataset, positive and negative documents differ only in a small window position. Another interesting observation regarding the entity-grid based neural models is that the model pretrained on the global discrimination task performs better than the ones trained on the specific tasks. From the table, we observe that our full model with ELMo word embeddings achieves the highest accuracies on the datasets $\mathcal{D}_{w=1,2,3}$, $\mathcal{D}_{w=2}$, and $\mathcal{D}_{w=3}$, while on the $\mathcal{D}_{w=1}$ dataset, our full model with the pretrained word2vec embeddings performs the best. The reason could be that with more generalized contextual embeddings, our model losses the discrimination capability for small changes in the document.

5.4 Results on Global Discrimination

Table 4 presents the results in accuracy on the two **"global" discrimination** tasks – the *Standard* and the *Inverse* order discrimination. The reported results of the entity-grid models are from the original papers. 'Lex. Neural Grid (M&J)(code)' refers to the results achieved by running the code released by Mohiuddin et al. (2018) on our machine.

From the table, we see that our unified neural coherence model outperforms the existing models by a good margin. In this dataset, our best model with the word2vec embeddings achieves 90.42% and 95.27%, on *Standard* and *Inverse* order discrimination tasks, respectively. We achieve the best results with our proposed model by using the ELMo word embeddings, where we get 93.19% and 96.78% accuracies on *Standard* and *Inverse* order discrimination tasks, respectively.

5.5 Ablation Study

To investigate the impact of different components in our proposed model, we conducted two sets of ablation study on the local and global discrimination tasks. Specifically, we want to see: (*i*) the impact of our global model component, and (*ii*) the impact of the language model (LM) loss.

Local Discrimination. In the local discrimination task, we first compare the performance of the proposed model without the LM loss. As shown in the first block of Table 5, addition of the global model to the local model degrades the performance on the $\mathcal{D}_{w=1}$ dataset by 1.17% and 1.21% for word2vec and ELMo embeddings, respectively. While for the other datasets, we see improvements in performance for the addition of the global model. However, in the presence of the LM loss (second block in Table 5), the addition of the global model improves the performance across

Model	Emb.	$\mathcal{D}_{w=1,2,3}$	$\mathcal{D}_{w=1}$	$\mathcal{D}_{w=2}$	$\mathcal{D}_{w=3}$
Without LM Loss					
Local Model	word2vec	73.23	66.21	73.16	77.93
Local Model	ELMo	74.12	65.82	73.54	78.16
Local + Global Model	word2vec	74.69	65.04	75.27	79.69
Local + Global Model	ELMo	76.01	64.61	75.22	79.37
With LM Loss					
Local Model	word2vec	75.03	66.95	75.04	80.07
Local Model	ELMo	75.20	64.22	75.93	80.57
Local + Global Model	word2vec	75.37	**67.29**	75.58	80.21
Local + Global Model	ELMo	**77.07**	64.38	**76.12**	**81.23**

Table 5: Ablation study of different model components on the **Local Discrimination task**.

all the datasets on the local discrimination task.

On the other hand, the addition of the LM loss to our model (with/without global model) increases the accuracy in most of the datasets and embeddings. Exception is the ELMo embeddings on $\mathcal{D}_{w=1}$ dataset, where the overall performance drops by 1.60% and 0.23% for the local model with and without the global model, respectively.

Another interesting observation on $\mathcal{D}_{w=1}$ dataset is that in all the cases word2vec embeddings outperforms ELMo. This unusual behavior of $\mathcal{D}_{w=1}$ dataset to the rests is not surprising because it is the hardest dataset where the difference between the positive and the negative document is subtle. In this case, generally flexible and simple models outperform complex ones.

For the performance degradation of the global model in $\mathcal{D}_{w=1}$ case, we assume that in some texts, the global model fails to capture the significant feature from the locally negative region. Subsequently, the global feature is added into the score calculation at every local window, so the overall influence of the global model becomes bigger than that of the local model in the decision making.

Global Discrimination. We also studied the impact of our global model and the LM loss in the global discrimination task. As shown in Table 6, the addition of the global model and LM loss to the local model improves performance on the standard discrimination task by 1.34%.

However, the addition of global model impacts negatively on the inverse order task and degrades accuracies by 2.42% and 1.28% in the presence and absence of LM loss respectively. We suspect that the global model is adding noise because of the pooling operation, which throws away the spatial relation between sentences and provides the global information that is invariant to the sentence-order. But in this task, order information is crucial.

Model	Emb.	Standard	Inverse
Our Local Model	word2vec	88.93	94.72
+ LM Loss	word2vec	89.92	**96.24**
+ Global Model	word2vec	89.53	93.44
+ Global Model + LM Loss	word2vec	**90.27**	93.82
(**Our Full Model**)			

Table 6: Ablation study of different model components on the **Global Discrimination task**.

In the inverse order task, we get the best performance by adding the LM loss to our local model.

6 Conclusion

In this paper, we proposed a unified coherence model. The proposed model incorporates a local coherence model and a global coherence model to capture the sentence grammar (intentional structure), discourse relations, attention and topic structures in a single framework. The unified coherence model shows state of the art results on the standard coherence assessment tasks: the inverse-order and the global discrimination tasks. Also, our evaluation of the local discrimination task demonstrates the effectiveness of the unified coherence model in assessing global and local coherence of texts.

References

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting on 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.

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *HLT-NAACL 2004: Main Proceedings*, pages 113–120, Boston, Massachusetts, USA. Association for Computational Linguistics.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature ver-

ification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, pages 737–744, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

François Chollet. 2016. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Micha Elsner, Joseph Austerweil, and Eugene Charniak. 2007. A unified local and global model for discourse coherence. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 436–443, Rochester, New York. Association for Computational Linguistics.

Micha Elsner and Eugene Charniak. 2011a. Disentangling chat with local coherence models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1179–1189, Stroudsburg, PA, USA. Association for Computational Linguistics.

Micha Elsner and Eugene Charniak. 2011b. Extending the entity grid with entity-specific features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 125–129, Portland, Oregon. Association for Computational Linguistics.

Vanessa Wei Feng and Graeme Hirst. 2012. Extending the entity-based coherence model with multiple ranks. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 315–324, Avignon, France. 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 *COLING*.

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

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

Amit Gruber, Yair Weiss, and Michal Rosen-Zvi. 2007. Hidden topic markov models. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 163–170, San Juan, Puerto Rico. PMLR.

Camille Guinaudeau and Michael Strube. 2013. Graph-based local coherence modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 93–103.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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.

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 - Volume 1*, HLT '11, pages 997–1006, Portland, Oregon. Association for Computational Linguistics.

Annie Louis and Ani Nenkova. 2012. A coherence model based on syntactic patterns. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1157–1168, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294, Prague, Czech Republic. Association for Computational Linguistics.

W. Mann and S. Thompson. 1988. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text*, 8(3):243–281.

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.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Muhammad Tasnim Mohiuddin, Shafiq Joty, 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.

Dat 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. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 186–195, Honolulu, Hawaii. Association for Computational Linguistics.

Radu Soricut and Daniel Marcu. 2006. Discourse generation using utility-trained coherence models. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, COLING-ACL '06, pages 803–810, Sydney, Australia. Association for Computational Linguistics.

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.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.