Towards coherent and cohesive long-form text generation

Woon Sang Cho[⋆] **Pengchuan Zhang**[†] **Yizhe Zhang**[†] **Xiujun Li**[†]
Michel Galley[†] **Chris Brockett**[†] **Mengdi Wang**[⋆] **Jianfeng Gao**[†]
[⋆]Princeton University
[†]Microsoft Research AI
[⋆]{woonsang,mengdiw}@princeton.edu
[†]{penzhan,yizzhang,xiul,mgalley,chrisbkt,jfgao}@microsoft.com

Abstract

Generating coherent and cohesive long-form texts is a challenging task. Previous works relied on large amounts of human-generated texts to train neural language models. However, few attempted to explicitly improve neural language models from the perspectives of coherence and cohesion. In this work, we propose a new neural language model that is equipped with two neural discriminators which provide feedback signals at the levels of sentence (cohesion) and paragraph (coherence). Our model is trained using a simple yet efficient variant of policy gradient, called *negative-critical sequence training*, which is proposed to eliminate the need of training a separate critic for estimating *baseline*. Results demonstrate the effectiveness of our approach, showing improvements over the strong baseline – recurrent attention-based bidirectional MLE-trained neural language model.

1 Introduction

The terms *coherence* and *cohesion* in linguistics are commonly defined as follows (Williams and Colomb, 1995).

- *Cohesion*: sentence pairs fitting together the way two pieces of a jigsaw puzzle do.
- *Coherence*: what all the sentences in a piece of writing add up to, the way all the pieces in a puzzle add up to the picture on the box.

In layman's terms, *cohesion* indicates that two consecutive sentences are *locally* well-connected, and *coherence* indicates that multiple sentences *globally* hold together.

Generating cohesive and coherent natural language texts that span multiple sentences is a challenging task for two principal reasons. First, there is no formal specification of cross-sentence linguistic properties, such as coherence and cohesion of a text. Secondly, there is no widely accepted model to measure the two properties.

Most state-of-the-art neural approaches to natural language generation rely on a large amount of human-generated text to train language models (Cho et al., 2014; Graves, 2013; Sutskever et al., 2014). Although these models can generate sentences that, if judged individually, are similar to human-generated ones, they often fail to capture the local and global dependencies among sentences, resulting in a text that is neither coherent nor cohesive. For example, neural language models based on Recurrent Neural Networks (RNNs) are widely applied to response generation for dialogue (Vinyals and Le, 2015; Shang et al., 2015; Sordoni et al., 2015; Li et al., 2015). Although the responses by themselves look reasonable, they are detached from the whole dialogue session. See Gao et al. (2018) for a comprehensive survey.

In this paper, we address the challenge in a principled manner, employing a pair of discriminators to score whether and to what extent a text is coherent or cohesive. The coherence discriminator measures the compatibility among all sentences in a paragraph. The cohesion discriminator measures the compatibility of each pair of consecutive sentences. These models, given a conditional input text and multiple candidate output texts, are learned to score the candidates with respect to the criterion. The scores are used as reward signals to train an RNN-based language model to generate (more) coherent and cohesive texts.

Contributions. Our main contributions are: (**1**) we propose two neural discriminators for modeling coherence and cohesion of a text for long-form text generation; (**2**) we present a simple yet effective training mechanism to encode these linguistic properties; (**3**) we propose *negative-critical sequence training*, a policy gradient method that uses negative samples to estimate its reward *baseline* and therefore eliminates the need for a sepa-

Proceedings of the First Workshop on Narrative Understanding, pages 1–11
Minneapolis, Minnesota, June 7, 2019.

rate critic function; and (**4**) we develop a new neural language model that generates more coherent and cohesive long-form texts, and empirically validate its effectiveness using the TripAdvisor and Yelp English reviews datasets.

2 Related work

Coherence and cohesion. Coherence and cohesion have been extensively studied in the computational linguistics community, particularly in the 'pre-deep-learning' era. Lack of formal specifications for coherence and cohesion (Mani et al., 1998), resulted in many different formalisms, such as Rhetorical Structure Theory (Mann and Thompson, 1988), and other forms of coherence and cohesion relations and their quantification (Mani et al., 1998; Hobbs, 1985; Hovy, 1988; McKeown, 1985; Cohen and Levesque, 1985; Hovy, 1991; Cristea et al., 1998; Halliday and Hasan, 1996; Liddy, 1991; Van Dijk, 2013; Edmundson, 1969; Barzilay and Lapata, 2008). This list is not exhaustive. However, prior work jointly exploring coherence and cohesion using neural models in the context of long-form text generation has not come to our attention.

Reinforcement learning for text generation. The text generation task can be framed as a reinforcement learning (RL) problem (Daumé et al., 2009), in which the generator G is acting as a *policy* π, with parameters θ_π, and each generated word at time t, w_t, can be viewed as an action to be chosen by the policy from a large discrete space, or vocabulary, conditioned on state $s_{t-1} = w_{\leq t-1}$.

Let r_t be the reward for a partially generated text sequence $w_{\leq t}$. We define the long-term expected reward $\mathcal{J}(\pi) = \mathbb{E}_{s_0 \sim q, \pi}[\sum_{t=1}^{\infty} \gamma^{t-1} r_t]$, where q is the initial distribution of conditional input texts. Following Sutton et al. (1999), the gradient of $\mathcal{J}$ with respect to θ_π is

$$\nabla_{\theta_\pi} \mathcal{J} = \mathbb{E}_{s \sim \rho^\pi, a \sim \pi(\cdot|s)}[Q^\pi(s,a) \nabla_{\theta_\pi} \log \pi_{\theta_\pi}(a|s)]$$

where ρ^π is the stationary distribution and $Q^\pi(s,a)$ is the expected return from state s and taking action a, both following policy π. For brevity, we omit the derivation. In this work, we formulate text generation as an episodic RL problem with episode length L, rewards r_L being available only at the end of episode and $\gamma = 1$.

There are many works on training neural language models using rewards, such as Ranzato et al. (2015) and Paulus et al. (2017). These works directly optimize for specific metrics, such as BLEU (Papineni et al., 2002) or ROUGE (Lin and Hovy, 2003), using REINFORCE (Williams, 1992). However, these metrics do not give a complete picture of the text generation quality. Only recently have there been efforts to provide more relevant objectives, such as consistency and repetition in a text (Li et al., 2015, 2016a; Holtzman et al., 2018). But these works use the objectives to re-rank candidate outputs, not to reward or penalize them. Li et al. (2016b) constructed a set of reward models for the dialogue task, such as information flow and semantic coherence, to tune the generator, yet they do not provide an ablation study on the relative contribution of these reward models individually. It is not clear that these reward models can be generalized to other tasks, in particular, long-form text generation tasks.

The most relevant to our work is Bosselut et al. (2018), which promotes text generation in the correct order, and discourages in its reverse order using rewards. However, this may not be sufficient in capturing coherence since there are many negative orderings given a paragraph. From this pool, we assess the relative quality of generations. Furthermore, we model cohesion between consecutive sentence pairs using word-level features.

GANs for text generation. Another line of research involves the use of Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) to incorporate feedback signals for text generation (Yu et al., 2017; Lin et al., 2017; Zhang et al., 2017; Guo et al., 2017; Fedus et al., 2018; Zhang et al., 2018). The discriminators in these works are trained to distinguish real texts from generated ones, operating as a black-box than providing feedback on linguistic aspects. Yang et al. (2018) partially addressed this issue by using a trained language model as the discriminator. Although the discriminator provides a fine-grained feedback at the word level, it does not model linguistic properties, such as cohesion and coherence.

Many text generator models are inadequate for generating a cohesive and coherent long-form text that span multiple sentences. As a result, human readers can easily distinguish the generated texts from real ones. In this paper, we argue that the primary reason is the lack of an effective mechanism to measure and control for the local and global consistency in model-generated texts.

3 Coherence and Cohesion Models

We assume that global coherence of a text depends to a large degree upon how its individual sentences with different meanings are organized. Therefore, we focus our evaluation of coherence solely based on the sentence-level features. If the sentences are not organized properly, the intention of the paragraph as a whole is obscure, regardless of seamless local connectivity between consecutive sentences.

This is not to say that local connections between any two neighboring sentences can be overlooked. One can easily distinguish a generated sentence from a real one by judging whether it is *semantically cohesive* with its neighboring sentences.

We strive to embody these two different yet important concepts by developing coherence and cohesion discriminators, operating on the sentence level and word level, respectively. Our design of these two discriminators is inspired by the Deep Structured Semantic Model (DSSM) which was originally developed to measure the semantic similarity between two texts (Huang et al., 2013; Gao et al., 2014; Palangi et al., 2016; Xu et al., 2017). In this study, we extend 'semantic similarity' to coherence and cohesion in a long-form text.

3.1 Coherence discriminator: $D_{\text{coherence}}$

The coherence discriminator models the coherence score, which measures how likely two text chunks add up to a single coherent paragraph. Let $S := [s_1, s_2, ..., s_n]$ be the source text chunk that consists of n sentences, $T := [t_1, t_2, ..., t_m]$ be the *real* target text chunk that consists of m sentences, and $\widetilde{T} := [\widetilde{t}_1, \widetilde{t}_2, ..., \widetilde{t}_{\widetilde{m}}]$ be the *artificially constructed incoherent* target text chunk that consists of $\widetilde{m}$ sentences. $D_{\text{coherence}}$ is designed to distinguish a positive (coherent) pair (S, T) from a negative (incoherent) pair $(S, \widetilde{T})$ by assigning different scores, i.e., $D_{\text{coherence}}(S, T) > D_{\text{coherence}}(S, \widetilde{T})$.

Model architecture. The model takes a form of dual encoder. Given source text chunk S and target text chunk T, the coherence discriminator $D_{\text{coherence}}$ computes the coherence score in three steps, as illustrated in Figure 1 (upper). First, each sentence is encoded by the bag-of-words (BOW) embedding, i.e., the average of its word vectors from a pre-trained word embedding (Pennington et al., 2014). Secondly, an encoder which can be implemented using a convolutional neural network (CNN)[1] or RNN[2], denoted as f, takes as input the BOW vectors of the source text chunk S and encodes it into a single vector $f(S)$. Similarly, g encodes the target text chunk T into $g(T)$. The two encoders $f(\cdot)$ and $g(\cdot)$ share the same architecture but do *not* share parameters, i.e., $\theta_f \neq \theta_g$, and thus $D_{\text{coherence}}(S, T)$ is *not* symmetric. Thirdly, $D_{\text{coherence}}(S, T)$ is computed as the cosine similarity of the two vectors $f(S)$ and $g(T)$. The score is a real value between -1 and 1, where 1 indicates maximal coherence, and -1 minimal coherence.

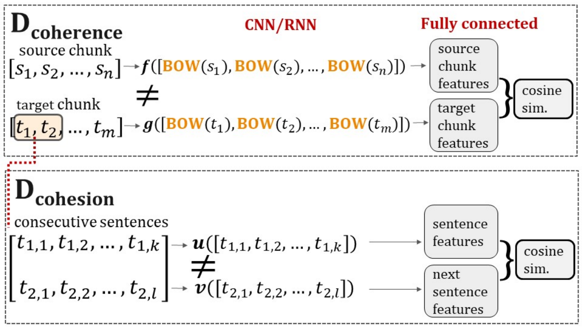

Figure 1: Illustration of coherence and cohesion discriminators. $D_{\text{coherence}}$ takes in bag-of-words sentence embeddings as inputs, and D_{cohesion} takes in the raw word embeddings of consecutive sentences as inputs. The source encoder f (or u) is different from the target encoder g (or v).

Note that we use the simple BOW vectors to encode sentences in the coherence discriminator, which is different from the CNN sentence embedding scheme in the cohesion discriminator that we introduce in Section 3.2. Although the BOW vector ignores the word-order information in the sentence, it is empirically shown to be effective in preserving the high-level semantic information in the sentences and achieves success in sentence similarity and entailment tasks (Wieting et al., 2016; Arora et al., 2017). Because high-level semantic information of sentences is sufficient to determine whether a paragraph is coherent, we choose to use BOW vectors to encode sentences in $D_{\text{coherence}}$.

The parameters of $D_{\text{coherence}}$, θ_f and θ_g are optimized using a pairwise ranking loss. To this end, we need both positive and negative pairs. While the positive (coherent) pairs come from the train-

[1] We explored with deeper networks. However, the performance difference was marginal. For simplicity, we decided to use a 1-layer convolutional network architecture (Kim, 2014; Collobert et al., 2011).

[2] For clarity in our model description, we omit RNN hereafter. We present results using both CNN and RNN encoders in Table 2.

ing data, negative (incoherent) pairs need to be artificially constructed. The next section describes the way these negative pairs are generated.

Constructing negative (incoherent) pairs. Given a training minibatch $\{(S_i, T_i)\}_{i=1}^{B}$, we construct $2*B-1$ negative pairs $\{(S_i, \widetilde{T_{i,j}})\}_{j=1}^{2B-1}$ for *every* positive pair (S_i, T_i) using three different methods, inspired by Wieting et al. (2016). For notation simplicity, we omit the minibatch index i in the rest of this section. For each positive pair (S, T) in the minibatch:

- We rotate T with S fixed, and thus obtain all $B-1$ mismatched pairs $\{(S, \widetilde{T}_j)\}_{j=1}^{B-1}$ as negative pairs.
- We shuffle the sentence order in T once, known as a derangement, to break its coherence. This yields one negative pair $(S, \widetilde{T})$.
- We combine the previous two methods, that is, we rotate T in the minibatch and shuffle sentences within the target chunk, yielding another $B-1$ negative pairs $\{(S, \widetilde{T}_j)\}_{j=1}^{B-1}$.

These $2B-1$ negative pairs and a single positive pair, in total, pose a challenge for the discriminator in learning to retrieve the correct pair.

Training using a pairwise ranking loss. The parameters of $f(\cdot)$ and $g(\cdot)$ are optimized in such a way that a positive pair scores higher than its negative pairs, i.e., $D_{\text{coherence}}(S, T) > D_{\text{coherence}}(S, \widetilde{T}_j)$ for any j. To achieve this, we propose to minimize the following pairwise ranking loss (Gong et al., 2013) with margin δ:

$$L_{\text{coherence}}(\theta_f, \theta_g) := \max\Big(0, \delta - D_{\text{coherence}}(S, T) + \text{AVG}^{\lambda}\Big(\{D_{\text{coherence}}(S, \widetilde{T}_j)\}_{j=1}^{2B-1}\Big)\Big). \quad (1)$$

where $\text{AVG}^{\lambda}(\{x_j\}_{j=1}^{N}) = \sum_{j=1}^{N} w_j x_j$ and $w_j = e^{\lambda x_j} / \sum_k e^{\lambda x_k}$.

Notice that AVG^{λ} is the *mean* operator when $\lambda = 0$ and approaches the *max* operator when $\lambda \to \infty$. These two extreme cases correspond to ranking against the average of all negative pairs and ranking against the single most challenging negative pair, respectively. Empirically, training the models using the *weighted* average ($0 < \lambda \ll \infty$), which assigns larger weights to more challenging negative pairs, stabilizes the training and expedites the convergence.

3.2 Cohesion discriminator: D_{cohesion}

The cohesion discriminator models the cohesion score, which measures how likely two sentences form a cohesive pair of consecutive sentences. Let $s_k := \left[s_k^1, s_k^2, ..., s_k^n\right]$ be the k^{th} sentence that consists of n words, $s_{k+1} := \left[s_{k+1}^1, s_{k+1}^2, ..., s_{k+1}^m\right]$ be the *real* next sentence that consists of m words, and $\widetilde{s}_{k+1} := \left[\widetilde{s}_{k+1}^1, \widetilde{s}_{k+1}^2, ..., \widetilde{s}_{k+1}^{\widetilde{m}}\right]$ be the *artificially constructed incohesive* next sentence that consists of $\widetilde{m}$ words. D_{cohesion} is designed to distinguish a positive (cohesive) pair (s_k, s_{k+1}) from a negative (incohesive) pair $(s_k, \widetilde{s}_{k+1})$ by assigning them with different scores, i.e., $D_{\text{cohesion}}(s_k, s_{k+1}) > D_{\text{cohesion}}(s_k, \widetilde{s}_{k+1})$.

Model architecture. Like the coherence discriminator, this model also takes a form of dual encoder. Given (s_k, s_{k+1}), D_{cohesion} computes the cohesion score in three steps, as illustrated in Figure 1 (lower). The first step is to obtain two sequences of word embedding to represent the two sentences. Then, a pair of source network $u(\cdot)$ and target network $v(\cdot)$ are utilized to encode both s_k and s_{k+1} into two low-dimensional continuous vectors. The two encoders $u(\cdot)$ and $v(\cdot)$ share the same architecture but do *not* share parameters, i.e., $\theta_u \neq \theta_v$, and thus the $D_{\text{cohesion}}(s_k, s_{k+1})$ is *not* symmetric. Finally, $D_{\text{cohesion}}(s_k, s_{k+1})$ is computed as the cosine similarity of the two vectors.

Note that we use CNNs or RNNs to embed sentences in D_{cohesion}, which takes the word order in a sentence into consideration. This is different from the BOW embedding in the $D_{\text{coherence}}$ where the word order does not matter, because the word order indeed matters when determining the cohesion of two consecutive sentences. As an example from Table 1, for the source sentence *"Once you get there you are greeted by the staff."*, *"They explain everything to you."* is a cohesive follow-up while *"You explain everything to them."* is not.

The parameters of D_{cohesion}, θ_u and θ_v are optimized using the same pairwise ranking loss. The positive pairs (a training minibatch) for D_{cohesion} is obtained from (1) decomposing each paragraph (S, T) in $\{(S_i, T_i)\}_{i=1}^{B}$ into pairs of consecutive sentences and (2) randomly selecting B pairs as the positive (cohesive) pairs $\{(s_k, s_{k+1})_i\}_{i=1}^{B}$. We construct negative (incohesive) pairs using the same methods as in the coherence discriminator.

Constructing negative (incohesive) pairs. We construct $2*B-1$ negative pairs $\{(s_k, \widetilde{s}_{k+1,j})_i\}_{j=1}^{2B-1}$ for *every* positive pair $(s_k, s_{k+1})_i$ using three different methods and omit the minibatch index i hereafter. For each positive pair (s_k, s_{k+1}) in the minibatch:

- We mismatch sentence pairs to obtain $\{(s_k, \widetilde{s}_{k+1,j})\}_{j=1}^{B-1}$.
- We shuffle words in s_{k+1} to obtain $\widetilde{s}_{k+1}$.
- We combine the previous two methods and obtain additional pairs $\{(s_k, \widetilde{s}_{k+1,j})\}_{j=1}^{B-1}$.

In total, we obtain $2B - 1$ negative pairs for each positive pair in the minibatch.

Training using a pairwise ranking loss. The parameters of $u(\cdot)$ and $v(\cdot)$ are optimized such that $D_{\text{cohesion}}(s_k, s_{k+1}) > D_{\text{cohesion}}(s_k, \widetilde{s}_{k+1,j})$ for any j. To achieve this, we propose to minimize the following pairwise ranking loss with margin δ:

$$L_{\text{cohesion}}(\theta_u, \theta_v) := \max\Big(0, \delta - D_{\text{cohesion}}(s_k, s_{k+1}) + \text{AVG}^{\lambda}\Big(\{D_{\text{cohesion}}(s_k, \widetilde{s}_{k+1,j})\}_{j=1}^{2B-1}\Big)\Big). \quad (2)$$

We leave the training details and hyper-parameter configurations to Section 5.2.

4 Negative-Critical Sequence Training for Long-form Text Generation

4.1 Long-form text generator: G

The generator G is an attention-based bidirectional sequence-to-sequence model (Bahdanau et al., 2014) and is pre-trained by maximizing the log likelihood on training data, which we denote as G_{MLE}. However, long-form texts generated using G_{MLE} often do not meet our high coherence and cohesion standards.

We propose to use the two pre-trained discriminators, $D_{\text{coherence}}$ and D_{cohesion}, to modify the text generation behavior of G_{MLE}. The scores from the discriminators are used as reward (or penalty) signals to adjust the parameters of G_{MLE} using a variant of policy gradient, called *negative-critical sequence training*, which we propose for our task and describe in details in the next subsection.

4.2 Negative-critical sequence training

For an arbitrary pair of S and T_{gen}, where T_{gen} is the generator's output conditioned on S, we compute the coherence and cohesion scores by calling $D_{\text{coherence}}$ and D_{cohesion}. Since each generated text consists of multiple sentences, the overall cohesion score is computed as the mean of all the consecutive sentence pairs, $(s_k, s_{k+1}) \subset [S_{-1}, T_{gen}]$, where S_{-1} is the last sentence from the source.

These scalar scores, however, are not interpretable since the discriminators are trained by optimizing a pairwise ranking loss. Instead, the differences between positive pair scores and the maximal or average negative pair scores provide insights of how well the models distinguish between the positive and the negative pairs.

This difference relates to reward with baseline in actor-critic methods (Barto et al., 1983; Witten, 1977; Williams, 1992; Sutton et al., 1999) that typically require a separate critic function as a baseline. In NLP, we have observed similar practices by Ranzato et al. (2015), Bahdanau et al. (2016), and Nguyen et al. (2017). Rennie et al. (2017) proposed a method that avoids learning a separate critic. Similarly, our method does not require learning a separate critic since this margin is a form of reward minus baseline. Specifically, we define the reward functions with baselines as:

$$R_{\text{coherence}}(S, T_{gen}) := D_{\text{coherence}}(S, T_{gen}) - \mathbb{E}_{\widetilde{T}}\Big[D_{\text{coherence}}(S, \widetilde{T})\Big] \quad (3)$$

$$R_{\text{cohesion}}([S_{-1}, T_{gen}]) := \frac{1}{|T_{gen}|} \sum_{\substack{(s_k, s_{k+1}) \\ \subset [S_{-1}, T_{gen}]}} D_{\text{cohesion}}(s_k, s_{k+1}) - \mathbb{E}_{\widetilde{s}_{k+1} \Big| \substack{(s_k, s_{k+1}) \\ \subset [S,T]}}\Big[D_{\text{cohesion}}(s_k, \widetilde{s}_{k+1})\Big] \quad (4)$$

where $|T_{gen}|$ denotes the number of sentences in T_{gen}, and $\mathbb{E}_{\widetilde{T}}$ (and $\mathbb{E}_{\widetilde{s}_{k+1}}$) are computed by averaging over an ensemble of negative pairs.

Notice that this reward resembles the ranking loss we use to train our discriminators, except that our baseline is the mean score (instead of the weighted mean) over negative pairs. The rationale for this difference is that: because the best artificially constructed negative sample may be a *formidably* good sample, the maximal or the weighted mean can in fact be noisy as a baseline and thus introduce noise in rewards. To alleviate such noise, we use the *mean discriminator score* of negative pairs as the baseline, and this turns out to be an empirically better alternative. Then we use policy gradient to maximize a weighted sum of the coherence and cohesion rewards.

5 Experiments

In this section, we detail the training and evaluation of $D_{\text{coherence}}$, D_{cohesion}, the baseline generator G_{MLE}, and the RL-tuned generators $G_{\text{MLE+RL(cohesion)}}$, $G_{\text{MLE+RL(coherence)}}$, and

source	**cohesion**	**coherence**
this hotel was unbelievably overpriced .	0.0002	
we were looking for something cheaper but thought we would at least be staying in a decent hotel having paid that much when booking .	0.0411	
it wasn t clear when booking that we would have to share a bathroom .	0.0084	
there was one shower for the whole floor which was tiny and unclean .	0.0054	
the room was old and lacking in facilities .		
target		
the beds were very uncomfortable and the linen was very old .	0.0768	
breakfast was ok , but the staff were incompetent .	0.0591	
on our last day they were too lazy to clean our table and never bothered taking our order .	-0.0097	**+0.3735**
we had to leave having had no breakfast , as we ran out of time .	0.0457	
they saw us get up and leave and didn t even apologise for the appalling lack of service .		
negative target		
the staff recommended great restaurants with very reasonable prices within walking distance .	0.0514	
the paris hop on bus stops nearby .	0.0798	
the gare l est is within 3 blocks .	-0.0156	
we paid 75 euro per nite excluding breakfast but paid for breakfast one day and found it very good and reasonably priced .	0.0082	**-0.2001**
the rooms are clean and bathrooms ensuite .		
more examples of cohesion		
once you get there you are greeted by the staff . they explain everything to you , and in english , not the best , but good enough .	0.1004	
the coffee was even good for a coffee snob like myself . the hotel is much smaller than i thought and only has six floors .	-0.1103	
the only negative was the curtain in the bathroom . it was very shear and we felt that people in the building across the street could look right in at night .	0.0787	
the beer at the lobby bar was stale . there are many friendly cats on the grounds .	-0.0830	

Table 1: Coherence and cohesion rewards on test data. The cohesion reward at the end of each line is computed with its next sentence. This is an example of contradiction and inconsistent sentiment, suggestive of incoherence. We append more examples with extreme cohesion rewards.

TripAdvisor		**Target Sentences Retrieval**		
Discriminators	**Encoding**	**R@1**	**R@5**	**R@10**
$D_{\text{coherence}}$	$\text{Conv}^{512}_{2,3,4,5}$	0.18	0.43	0.60
	$\text{GRU}^{1024}_{\text{1-layer, bi-dir.}}$	**0.26**	**0.50**	**0.65**
D_{cohesion}	$\text{Conv}^{512}_{3,4,5,6}$	**0.12**	**0.28**	**0.43**
	$\text{GRU}^{1024}_{\text{1-layer, bi-dir.}}$	0.11	0.21	0.33

Yelp		**Target Sentences Retrieval**		
Discriminators	**Encoding**	**R@1**	**R@5**	**R@10**
$D_{\text{coherence}}$	$\text{Conv}^{512}_{2,3,4,5}$	0.33	0.61	0.74
	$\text{GRU}^{1024}_{\text{1-layer, bi-dir.}}$	**0.39**	**0.68**	**0.81**
D_{cohesion}	$\text{Conv}^{512}_{3,4,5,6}$	**0.14**	**0.33**	**0.47**
	$\text{GRU}^{1024}_{\text{1-layer, bi-dir.}}$	0.11	0.26	0.39

Table 2: Retrieval ratios for coherence and cohesion discriminators from a collection of 100 negative candidates from the test data. The reported numbers are the averages over 20 evaluations. Notations: $\text{Conv}^{512}_{2,3,4,5}$ is a convolutional input encoder with filter sizes 2, 3, 4, and 5, and there are 512 filters for each filter size. $\text{GRU}^{1024}_{\text{1-layer, bi-dir.}}$ is a 1-layered bi-directional GRU input encoder with hidden size 1024. We experimented different configurations for both encoder types, and selected the best performing models for the negative-critical sequence training step.

$G_{\text{MLE+RL(coherence, cohesion)}}$. We show that, by using feedback from the discriminators, the quality of the generated texts is significantly improved. See Table 3 for a sample comparison.

5.1 Dataset

We use the TripAdvisor hotel English reviews dataset collected by Wang et al. (2010) and the Yelp English reviews dataset[3]. We use only the

[3]https://www.yelp.com/dataset

source sentences	the hotel inglaterra delivered as promised . the staff was welcoming and spoke good english . the cleaning staff did a very good job every day . the rooms were spotless and very modern . the bathroom was large and had a very nice shower , and there were two generously sized bath towels that were twice the size of normal towels .
$G_{\textbf{MLE}}$	the breakfast in the morning was delicious and very good . it was the only hotel where i slept very well . the staff was very helpful in late afternoon or late times . the breakfast was adequate , with a decent range of cereals , fruit , and fruits . there is also free use of the coffee in the reception area .
$G_{\textbf{MLE+RL(coherence, cohesion)}}$	the breakfast was plentiful including fresh breads and cooked to order . the location was fantastic . it is in the north of the marina and in a very short distance . the marina has a small swimming pool with sitting area and a small gym . they are very popular and guests have an evening reception which is very nice .

Table 3: Sample generations from our MLE-trained baseline model, G_{MLE}, and our discriminator-guided model $G_{\text{MLE+RL(coherence, cohesion)}}$. The red texts highlight a common problem in G_{MLE} - it exhibits a repetition, and an inconsistent opinion as a review. In contrast, our discriminator-guided model is able to generate a more interesting, and sentiment-consistent continuation.

	Model	NLL	PPL	BLEU-3	BLEU-4	BLEU-5	intra-unique-1	intra-unique-2	inter-unique-2	inter-unique-3	length ratio
TripAdvisor	G_{MLE} (baseline)	0.86	2.36	0.38	0.19	0.08	0.66	0.93	0.40	0.72	1.08
	$G_{\text{MLE +RL(cohesion)}}$	**0.77**	**2.18**	**0.46**	**0.27**	**0.14**	0.64	0.94	0.38	0.71	0.97
	$G_{\text{MLE+RL(coherence)}}$	0.80	2.24	0.44	0.25	0.12	0.64	0.94	0.39	0.72	1.06
	$G_{\text{MLE+RL(coherence, cohesion)}}$	0.80	2.25	0.44	0.24	0.12	0.65	0.94	0.40	0.72	1.02

	Model	NLL	PPL	BLEU-3	BLEU-4	BLEU-5	intra-unique-1	intra-unique-2	inter-unique-2	inter-unique-3	length ratio
Yelp	G_{MLE} (baseline)	1.32	3.84	0.37	0.17	0.07	0.68	0.95	0.54	0.86	1.07
	$G_{\text{MLE+RL(cohesion)}}$	1.26	3.65	**0.45**	**0.23**	**0.11**	0.68	0.95	0.53	0.85	1.05
	$G_{\text{MLE+RL(coherence)}}$	**1.24**	**3.56**	0.45	0.23	0.11	0.69	0.95	0.55	0.87	1.00
	$G_{\text{MLE+RL(coherence, cohesion)}}$	1.25	3.59	0.43	0.22	0.11	0.69	0.95	0.56	0.88	1.05

Table 4: An ablation study with automated evaluation metric scores: NLL, PPL, BLEU-n, intra/inter-unique-n, along with the length ratio with the length of corresponding true target sentences as 1. Significant numbers are highlighted in **bold** before rounding.

subsets of the two datasets that satisfy the following two conditions: (1) a review must have at least 10 sentences, and (2) each sentence has from 5 to 30 words. This yields roughly 60,000 TripAdvisor reviews and 220,000 Yelp reviews, split into $[0.8, 0.1, 0.1]$ ratio for train/dev/test sets.

We merge the source and target vocabularies, and limit it to the top 50,000 frequent words, excluding special tokens. For each review, we use the first five sentences as the input S to G, and the next five sentences as the target output T from G.

5.2 Implementation details

Baseline G_{MLE}. G_{MLE} takes individual words as inputs and embeds into a pre-trained GloVe 300-dimensional word vectors. This embedding layer is fixed throughout training. G_{MLE} uses a two-layered GRU and hidden size of 1024 for both encoder and decoder. During optimization using Adam (Kingma and Ba, 2014), we set the learning rate to 2e-4 and clip the gradient's L2-norm to 1.0. We initially train G_{MLE} for 60 epochs on the TripAdvisor data and 30 epochs on the Yelp data.

Discriminators. For the CNN-based encoder, the convolutional layer consists of filters of sizes 2, 3, 4, and 5 for $D_{\text{coherence}}$ (3, 4, 5, and 6 for D_{cohesion}), each with 512 filters. Each convolution filter is followed by a $\tanh$ activation. Then, we max-pool in time and append a fully connected layer to generate a feature vector of dimension 512, followed by a batch normalization layer and a $\tanh$ activation. For the RNN-based encoder, we use a 1-layered bi-directional GRU, concatenate the final hidden states at both ends, and append the same remaining layers.

Both discriminators use the pre-trained GloVe word embedding vectors[4], which are fixed during the training. We use an Adam optimizer with a learning rate of 1e-5. We fix $\lambda = 2$ and $\delta = 0.2$ in equations (1) and (2).[5] We train both discriminators for 50 epochs and choose the models with the best R@1 scores on the validation dataset.

Model $G_{\text{MLE+RL}}$. In the fine-tuning stage, we use the negative-critical sequence training method,

[4]The vector dimension can be different from that of G. The differences were marginal for sizes 50, 100, and 300. For results shown in this paper, we used the same dimension of size 300.

[5]We performed a coarse grid search over the values of λ and δ and these values for the hyper-parameters pair resulted in fast convergence to high recall scores on the dev dataset.

Cohesion					Coherence				
Human judges preferred:					*Human judges preferred:*				
Our Method		Neutral	Comparison		Our Method		Neutral	Comparison	
$G_{\text{MLE+RL}}$	**36.41%**	33.57%	30.50%	G_{MLE}	$G_{\text{MLE+RL}}$	**37.23%**	31.44%	31.80%	G_{MLE}
$G_{\text{MLE+RL}}$	29.91%	30.85%	**39.24%**	Human	$G_{\text{MLE+RL}}$	28.96%	31.32%	**39.72%**	Human

Table 5: Results of **Human Evaluation** showing preferences (%) for our model $G_{\text{MLE+RL(coherence, cohesion)}}$ vis-a-vis the baseline G_{MLE} after adjustment for spamming. $G_{\text{MLE+RL(coherence, cohesion)}}$ is preferred over G_{MLE}. For simplicity, the 5-point Likert scale has been collapsed to a 3-point scale. See the Appendix for further details of distributions.

as described in Section 4, up to 5 epochs, with a learning rate of 1e-5. We equally weight the coherence and cohesion rewards, $\frac{1}{2}R_{\text{coherence}}(S, T_{gen}) + \frac{1}{2}R_{\text{cohesion}}([S_{-1}, T_{gen}])$. We also continue the supervised learning of G to constrain the policy search within a space that represents the sentences that are likely to be grammatically plausible, similar to Paulus et al. (2017); Wu et al. (2016); Lewis et al. (2017). For all the generations from G_{MLE} and $G_{\text{MLE+RL}}$, we use the simple greedy decoding method because we do not observe any significant difference when switching to beam search.

5.3 Results

Evaluating $D_{\text{coherence}}$ and D_{cohesion}. Since the discriminators are implemented as pairwise rankers, we employ the metrics commonly used in information retrieval for evaluation, i.e., recall at K (R@K), which is defined as the fraction of correctly identifying an item in the TOP-K retrieved list (Baeza-Yates and Ribeiro-Neto, 1999). We present the retrieval results in Table 2. To help readers understand the roles of $D_{\text{coherence}}$ and D_{cohesion}, we present examples of positive and negative pairs and their rewards in Table 1.

Automatic evaluation of G. It is widely known that there is no perfect automated metric to evaluate text generators. Nevertheless, we report the scores of widely used metrics, including negative log-likelihood (NLL), perplexity (PPL), BLEU and the proportion of unique n-grams within a single generation (intra-unique-n), and across generations (inter-unique-n), as in Gu et al. (2018). Results in Table 4 show that our discriminators significantly improve BLEU scores, NLL and PPL, with marginal difference in diversity.

Human evaluation of G. Coherence and cohesion of a text cannot be easily measured using standard automated metrics. Thus, we perform crowd-sourced human evaluation. We randomly selected 200 samples from the TripAdvisor dataset, including corresponding generated output from the baseline G_{MLE} and our model $G_{\text{MLE+RL}}$. For comparison, we pair systems as (*Human* $\leftrightarrow G_{\text{MLE+RL}}$) and ($G_{\text{MLE+RL}} \leftrightarrow G_{\text{MLE}}$).

The outputs of these system pairs are presented in random order and each is ranked in terms of coherence and cohesion using a five-point Likert scale by human judges. Initially, we hired 7 judges to judge each pair. We identified a group of poor judges (probable spammers) who choose $G_{\text{MLE+RL}}$ over the *Human* more than 40% of the time, and eliminated them from the judge pool. Table 5 reports the final scores in terms of percentages of the total remaining judgments.

6 Conclusion

This paper proposes a neural approach to explicitly modeling cross-sentence linguistic properties, coherence and cohesion, for long-form text generation. The coherence discriminator $D_{\text{coherence}}$ provides a macro-level view on structuring a paragraph. The cohesion discriminator D_{cohesion} provides a micro-level view on local connectivity between neighboring sentences. The pre-trained discriminators are used to score the generated texts and artificially constructed negative pair scores are used to form baselines for the policy gradient, which we call negative-critical sequence training, to train neural language models.

On two long-form text generation tasks, human evaluation results are consistent with automatic evaluation results, which together demonstrate that our proposed method generates more locally and globally consistent texts with the help of the discriminators.

Despite the encouraging initial results, we only scratched the surface of the problem. The proposed method is yet to be significantly improved to meet the ultimate goal of generating meaningful and logical long-form texts.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*.

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern information retrieval*, volume 463. ACM Press Books.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Andrew G Barto, Richard S Sutton, and Charles W Anderson. 1983. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, SMC-13(5):834–846.

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

Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, Jianfeng Gao, Po-Sen Huang, and Yejin Choi. 2018. Discourse-aware neural rewards for coherent text generation. In *Proc. of NAACL*, pages 173–184.

Kyunghyun Cho, Bart van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*.

Philip R Cohen and Hector J Levesque. 1985. Speech acts and rationality. In *Proceedings of the 23rd annual meeting on Association for Computational Linguistics*, pages 49–60. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Dan Cristea, Nancy Ide, and Laurent Romary. 1998. Veins theory: A model of global discourse cohesion and coherence. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 281–285. Association for Computational Linguistics.

Hal Daumé, John Langford, and Daniel Marcu. 2009. Search-based structured prediction. *CoRR*, abs/0907.0786.

Harold P Edmundson. 1969. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285.

William Fedus, Ian Goodfellow, and Andrew Dai. 2018. MaskGAN: Better text generation via filling in the ····. In *ICLR*.

Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational AI. *arXiv preprint arXiv:1809.08267*.

Jianfeng Gao, Patrick Pantel, Michael Gamon, Xiaodong He, and Li Deng. 2014. Modeling interestingness with deep neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2–13.

Yunchao Gong, Yangqing Jia, Thomas Leung, Alexander Toshev, and Sergey Ioffe. 2013. Deep convolutional ranking for multilabel image annotation. *arXiv preprint arXiv:1312.4894*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Xiaodong Gu, Kyunghyun Cho, JungWoo Ha, and Sunghun Kim. 2018. DialogWAE: Multimodal response generation with conditional wasserstein auto-encoder. *CoRR*, abs/1805.12352.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2017. Long text generation via adversarial training with leaked information. *arXiv preprint arXiv:1709.08624*.

M Halliday and Ruqaiya Hasan. 1996. Cohesion in text.

Jerry R Hobbs. 1985. On the coherence and structure of discourse.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *Proceedings of the Association for Computational Linguistics*.

Eduard H Hovy. 1988. Planning coherent multisentential text. In *Proceedings of the 26th annual meeting on Association for Computational Linguistics*, pages 163–169. Association for Computational Linguistics.

Eduard H Hovy. 1991. Approaches to the planning of coherent text. In *Natural language generation in artificial intelligence and computational linguistics*, pages 83–102. Springer.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.

Elizabeth DuRoss Liddy. 1991. The discourse-level structure of empirical abstracts: An exploratory study. *Information Processing & Management*, 27(1):55–81.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram cooccurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 71–78, Stroudsburg, PA, USA.

Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial ranking for language generation. In *Advances in Neural Information Processing Systems*, pages 3155–3165.

Inderjeet Mani, Eric Bloedorn, and Barbara Gates. 1998. Using cohesion and coherence models for text summarization. In *Intelligent Text Summarization Symposium*, pages 69–76.

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

Kathleen R McKeown. 1985. Discourse strategies for generating natural-language text. *Artificial Intelligence*, 27(1):1–41.

Khanh Nguyen, Hal Daumé, and Jordan L. Boyd-Graber. 2017. Reinforcement learning for bandit neural machine translation with simulated human feedback. In *EMNLP*.

H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):694–707.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *CoRR*, abs/1511.06732.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *NAACL-HLT*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, pages 1057–1063. MIT Press.

Teun A Van Dijk. 2013. *News as discourse*. Routledge.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *ICML Deep Learning Workshop*.

Hongning Wang, Yue Lu, and ChengXiang Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *KDD*.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. *ICLR*.

J.M. Williams and G.G. Colomb. 1995. *Style: Toward Clarity and Grace*. Chicago guides to writing, editing, and publishing. University of Chicago Press.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3-4):229–256.

Ian H Witten. 1977. An adaptive optimal controller for discrete-time markov environments. *Information and control*, 34(4):286–295.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. 2017. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. *arXiv preprint*.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. *arXiv preprint arXiv:1805.11749*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. In *AAAI*.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *NIPS*.

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. In *NIPS*.