

# RETHINKING SELF-SUPERVISION OBJECTIVES FOR GENERALIZABLE COHERENCE MODELING

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Although large-scale pre-trained neural models have shown impressive performances in a variety of tasks, their ability to generate coherent text that appropriately models discourse phenomena is harder to evaluate and less understood. Given the claims of improved text generation quality across various systems, we consider the coherence evaluation of machine generated text to be one of the principal applications of coherence models that needs to be investigated. We explore training data and self-supervision objectives that result in a model that generalizes well across tasks and can be used off-the-shelf to perform such evaluations.

Prior work in neural coherence modeling has primarily focused on devising new architectures, and trained the model to distinguish coherent and incoherent text through pairwise self-supervision on the permuted documents task. We instead use a basic model architecture and show significant improvements over state of the art within the same training regime. We then design a harder self-supervision objective by increasing the ratio of negative samples within a contrastive learning setup, and enhance the model further through automatic hard negative mining coupled with a large global negative queue encoded by a momentum encoder. We show empirically that increasing the density of negative samples improves the basic model, and using a global negative queue further improves and stabilizes the model while training with hard negative samples. We evaluate the coherence model on task-independent test sets that resemble real-world use cases and show significant improvements in coherence evaluations of downstream applications.

## 1 INTRODUCTION

Coherence is a property of a well-written text that makes it different from a random set of sentences: sentences in a coherent text are connected in systematic ways such that each sentence follows naturally from previous ones and leads into the following ones (Halliday & Hasan, 1976; Grosz & Sidner, 1986). **Coherence models** (Barzilay & Lapata, 2005) that can distinguish a coherent text from incoherent ones have a wide range of applications in language generation, summarization, and coherence assessment tasks such as essay scoring and sentence ordering.

With the advancements of neural methods in recent years, claims of fluency in summarization (Liu et al., 2017; Celikyilmaz et al., 2018), language modeling (Radford et al., 2019; Brown et al., 2020), response generation (Zhang et al., 2020; Hosseini-Asl et al., 2020) and human parity in machine translation (Hassan et al., 2018) have led to calls for finer-grained discourse-level evaluations (Läubli et al., 2018; Sharma et al., 2019; Popel et al., 2020), since traditional metrics such as BLEU and ROUGE are unable to measure text quality and readability (Paulus et al., 2018; Reiter, 2018). Coherence models that can evaluate machine-generated text have become the need of the hour.

A majority of coherence models proposed optimize their learning objectives on the permuted document task that uses the Penn Treebank (WSJ) corpus. The current paradigm of coherence modeling that uses permuted documents to train pairwise ranking models was originally proposed by Barzilay & Lapata (2005; 2008) to emulate *entity-based* incoherence, which has its origins in Centering Theory (Grosz et al., 1995). An original article is considered a ‘positive’ sample of a coherent document, while a permutation of its sentences is considered a ‘negative’ or incoherent sample (see Appendix A.1 for an example). Models are usually trained in a *pairwise ranking* fashion to distinguish the two.

The basic entity-grid model proposed by Barzilay & Lapata (2005; 2008) was extended to incorporate entity-specific features (Elsner & Charniak, 2011), multiple ranks (Feng & Hirst, 2012), and coherence relations (Lin et al., 2011; Feng et al., 2014). Their neural extensions have also been proposed (Nguyen & Joty, 2017; Mohiuddin et al., 2018). More recent state-of-the-art models like the Transferable Neural model (Xu et al., 2019) consider coherence at a local level by training a forward and backward model only on adjacent sentences, in addition to generative pre-training of the sentence encoders. The Unified Coherence model (Moon et al., 2019) uses bi-linear layer and lightweight convolution-pooling in a Siamese framework to capture discourse relations and topic structures, along with an explicit language model loss to capture syntactic patterns.

Mohiuddin et al. (2021) recently tested these state-of-the-art models by conducting coherence evaluations on the WSJ permuted document task, machine translation, summarization and next utterance ranking tasks. They found that while models performed well on the permuted document task, when tested off-the-shelf, models generalized poorly to downstream evaluation tasks. They call for more comprehensive evaluations of coherence models. Pishdad et al. (2020) also reached a similar conclusion. They retrained several neural coherence models for tasks analogous to coherence modeling such as detecting connective substitution and topic switching. They found that performance on the permuted document task is only partially indicative of a model’s coherence modeling capabilities.

In light of these recent findings, our aim in this work is to propose a coherence model that generalizes well to other tasks, and can be used off-the-shelf for coherence evaluations of downstream applications such as machine generated text. We train our model purely through *self-supervision*, without tailoring the model architecture to be specific to the permuted document task or any other form of supervision. Our main hypothesis is that large-scale pre-trained models like XLNet (Yang et al., 2019) are expressive enough to capture coherence information given the right self-supervision.

Li & Jurafsky (2017) point out that coherence models are exposed to a limited number of incoherent samples in the *pairwise* setup, since only a small sample of all possible incoherent permutations of a document are used to train models. Learning with more negatives can better maximize the mutual information between their representations (van den Oord et al., 2018). By using a *contrastive learning* (Gutmann & Hyvärinen, 2010) setup, where each ‘positive’ document is compared with multiple ‘negative’ documents, we increase the proportion of negative samples that the model is exposed to, and show that the coherence model shows significant improvements in performance.

Wu et al. (2020) recently show that the difficulty of the negative samples used for contrastive training can strongly influence model success for visual representation learning. Guided by this principle, we train the model with hard negative samples that are automatically mined, coupled with a large global negative queue encoded by a momentum encoder (He et al., 2019).

We evaluate our model on various independent test sets that demonstrate its applicability in downstream applications: machine generated summaries, language model outputs and commonsense reasoning, in addition to testing on coherence-specific test sets. In summary, our contributions are:

- A neural coherence model trained purely through well-designed self-supervision tasks that generalizes well to downstream applications and can be used off-the-shelf for coherence evaluation.
- Evaluation on multiple independent test sets that are more indicative of real-world performance of the coherence model.
- Empirical results demonstrating that an increase in the density and quality of negative samples leads to better generalization for coherence models.

## 2 DATASETS

In order to ensure that our coherence model is useful for evaluation in downstream applications, we use a selection of task-independent test sets that cover a variety of domains and genres, including machine generated text from summarization systems and language models. Following Pishdad et al. (2020), we also evaluate the models on a commonsense reasoning narrative dataset. Since our objective is to find the best training paradigm that can be used off-the-shelf for coherence evaluation, we train (and validate) the coherence models on standard WSJ data, while using the rest as “independent” test sets to indicate the generalizability of the trained models. All evaluations on the independent test sets are conducted in a pairwise setting to enable a fair comparison.

## 2.1 TRAINING DATA

**WSJ** The Wall Street Journal (WSJ) corpus consists of news articles which are divided into 1,240 documents for training, 138 documents for development and 1,053 documents for testing in the standard setup. We exclude documents with fewer than 4 sentences and truncate them to a maximum length of 600 tokens. In order to maximally utilize documents which are otherwise truncated due to GPU memory constraints, we partition documents with 20+ sentences into blocks of 10 sentences and consider each block as a separate positive document. This increases the number of coherent ‘documents’ that we can use to generate a much larger training set. Moon et al. (2019) use upto 20 permutations of a document to train their model; since their training setup is pairwise, it means that the original positive document is repeated 20 times. We regenerate the permuted documents similarly, sampling a larger set of permutations for our contrastive learning setup.<sup>1</sup> This gives us 46,522 instances of positive and their corresponding negative documents for training and 4,522 instances for development. We use the original pairwise test set used by Moon et al. (2019) with 20,411 instances for testing.

## 2.2 MACHINE GENERATED TEXTS

**SUMMEVAL** Fabbri et al. (2020) conduct a manual coherence evaluation of the summaries generated by 16 different summarization systems for 100 source articles based on the CNN/DailyMail (Hermann et al., 2015) dataset. Likert-style coherence ratings from 3 expert annotators are available for each summarized text. We adapt this to the pairwise setting by creating pairs of summaries from every system for each unique source article. The summary with the higher average coherence rating is designated as the positive document, while the summary with the lower rating is the negative document for that pair. This results in  $\binom{16}{2} \times 100 = 12,000$  pairs for evaluation.

**LMVLM** To cover a wider variety of machine generated text, we generated texts from various language models using prompts taken from the validation and test sets of the WritingPrompts dataset (Fan et al., 2018). Four language models were chosen for this purpose: GPT2-Small, GPT2-XL, CTRL and GPT3. The continuations produced by these models for each prompt were truncated at approximately 150 tokens and paired together. Using these texts, we conducted a user study on Amazon Mechanical Turk. Workers were instructed about the concept of coherence and shown examples of coherent and incoherent texts. Given the prompt, they were asked to choose the more coherent text out of two given language model outputs; they were also given an option to choose neither in case the texts were equally coherent/incoherent (see Appendix A.3 for more details such as the study interface). After removing the samples with low agreements and ties, a total of 1046 pairs with judgments from 3 annotators each were collected. The Krippendorff’s alpha coefficient (Krippendorff, 2011) between the annotators was **0.84**. We calculate the agreements of the coherence model ranking with these judgments, designated LMVLM.

## 2.3 CURATED TEST SETS

**INSTED** Shen et al. (2021) propose a sentence intrusion detection task in order to test the coherence modeling capabilities of pre-trained language models. Incoherent documents are created by substituting a sentence from a document with another sentence from a different document, ensuring that the replacement sentence is similar to the original document to make the task sufficiently hard. We adapt their task to the pairwise setting by pairing the original coherent and the corrupted incoherent document, giving us 7,168 instances from their CNN test set (INSTED-CNN) and 3,666 instances from their Wikipedia test set (INSTED-WIKI) for evaluation. Shen et al. (2021) also create a hand-crafted linguistic probe test set, where incoherence is manually inserted based on a range of linguistic phenomena; we use this test set for analysis (§4).

**STORYCLOZE** The STORYCLOZE dataset (created from ROCSTORIES (Sharma et al., 2018)) consists of a short narrative-style text with two possible endings, one of which is implausible. The test set labels are not public so we use the validation set. We designate the text with the correct ending as the positive document and the text with the incorrect ending as the negative document, resulting in a total of 1,571 pairs for evaluation.

<sup>1</sup>We ensure that the generated permuted documents are not repeated. For example, our contrastive learning setup requires 5 negative samples per instance; because each positive document appears 20 times in the original dataset, 100 unique permutations would be generated and divided accordingly.

### 3 METHODOLOGY

#### 3.1 MODEL ARCHITECTURE

Previous work on coherence modeling proposed elaborate architectures to capture various aspects of coherence (see §1). However, our key hypothesis is that large-scale pre-trained models are expressive enough to model coherence given the right self-supervision; [Abhishek et al. \(2021\) show some results to this effect](#). Effective bi-directional encoding through large Transformer networks (Vaswani et al., 2017) can consider longer language context, while language modeling objectives enforce syntactic and local coherence patterns in the model.

In our work, we adopt XLNet (Yang et al., 2019) as the backbone model. It is trained using a permuted language modeling objective, in which the expected log-likelihood of a sequence with respect to all permutations of the factorization order is maximized. This allows the modeling of bi-directional context, while maintaining the auto-regressive property and avoiding the pretrain-finetune discrepancy. In addition, XLNet also incorporates segment recurrence (or memory) and the relative encoding scheme of Transformer-XL (Dai et al., 2019), which makes it effective in modeling longer text sequences. This makes it suitable for our purpose of coherence modeling.

Given a document  $\mathcal{D}$  with  $n$  sentences  $(s_1, s_2, \dots, s_n)$  as input, our model uses the representations obtained through XLNet (parameterized by  $\phi$  in Figure 1) to assign a coherence score to the model. Specifically, for each sentence  $s_i$  with  $k$  tokens  $(w_1, w_2 \dots w_k)$ , XLNet maps each token  $w_t$  to its vector representation  $v_t \in \mathbb{R}^d$  where  $d$  is the dimension of the embedding. In addition, the complete input  $\mathcal{D}$  is also mapped to a document representation  $\mathbf{z} \in \mathbb{R}^d$  (i.e., the representation of the [CLS] token). We simply add a linear layer to convert document representation  $\mathbf{z}$  to obtain the final coherence score:  $f_\theta(\mathcal{D}) = \mathbf{w}^\top \mathbf{z} + b$ , where  $\mathbf{w}$  and  $b$  are the weight and bias of the linear layer with  $\theta = \{\phi, \mathbf{w}, b\}$  being the entire parameter set of the model (see the upper part of Figure 1).

#### 3.2 MARGIN-BASED PAIRWISE RANKING

**Setup.** Traditionally, coherence model training has been done in a pairwise ranking setup. In this setup, the model is trained to score the coherent or positive document higher than the incoherent or negative document, using a pairwise ranking loss (Collobert et al., 2011) defined as follows:

$$\mathcal{L}_\theta(\mathcal{D}^+, \mathcal{D}^-) = \max(0, \tau - f_\theta(\mathcal{D}^+) + f_\theta(\mathcal{D}^-)) \quad (1)$$

where  $f_\theta(\mathcal{D}^+)$  is the coherence score of the positive document,  $f_\theta(\mathcal{D}^-)$  is the coherence score of the negative document and  $\tau$  is the margin.

**Baseline.** Results from evaluation of existing coherence models by both Pishdad et al. (2020) and Mohiuddin et al. (2021) indicate that the Unified Coherence model or UNC (Moon et al., 2019) is overall the best-performing model (see [Appendix A.5 for a full comparison](#)). We retrain their model with our training data for comparison<sup>2</sup>. In addition, to ascertain the contribution of the pre-trained XLNet embeddings, we train our pairwise model without fine-tuning the representations, i.e., only the score-producing linear layer weights  $\mathbf{w}$  and  $b$  are trained on the pairwise ranking task.

**Results.** Results for the baseline models are given in Table 1 (see first two rows). We see that despite relatively high performance on the WSJ test set (94.11%), UNC’s performance on the independent test sets is quite poor, often failing to do better than a random baseline of 50% in 3 out of 5 cases. The performance on the INSTED-CNN dataset, which is the same domain (news) as the training data, is relatively better at 67.21%. [Our XLNet-Pairwise model trained without fine-tuning the representations \(No FT\) has some success on the SUMMEVAL and STORYCLOZE datasets compared to UNC, but overall the performance of this model is worse. This shows that the UNC model is in fact a strong baseline model despite using ELMo \(Peters et al., 2018\) pretrained representations.](#) Our fully-trained XLNet-Pairwise model not only outperforms the SOTA UNC model on the standard WSJ permuted document task, but also significantly outperforms this model on the independent test sets, showing an absolute improvement of 15-20% on the SUMMEVAL, INSTED-CNN, INSTED-WIKI and the STORYCLOZE datasets. On LMvLM, the UNC model has a better performance; we suspect that its explicit conditional language modeling loss might provide an additional advantage for this particular task. Overall, our results are consistent with observations from Mohiuddin et al. (2021) that show poor generalizability in the previous SOTA model.

<sup>2</sup>Code taken from <https://github.com/taasnim/unified-coherence-model>

Table 1: Results on the WSJ permuted document test set and the various independent test sets of the previous SOTA UNC model and our XLNet based models. Except for the LMvLM results which are reported in terms of Krippendorff’s alpha agreement with human annotators, all other results are reported in terms of accuracy of the models in scoring the positive document higher than the negative document. All results are averaged over 5 runs with different seeds.

Model	WSJ	SUMMEVAL	LMvLM	INSTED-CNN	INSTED-WIKI	STORYCLOZE
UNC	94.11 $\pm$ 0.29	46.28 $\pm$ 0.80	0.463 $\pm$ 0.01	67.21 $\pm$ 0.55	55.97 $\pm$ 0.45	49.39 $\pm$ 1.81
Our - Pairwise (No FT)	71.70 $\pm$ 1.02	54.93 $\pm$ 1.91	0.421 $\pm$ 0.01	59.96 $\pm$ 3.15	53.45 $\pm$ 0.86	51.69 $\pm$ 1.32
Our - Pairwise	98.23 $\pm$ 0.20	64.83 $\pm$ 1.03	0.458 $\pm$ 0.02	91.96 $\pm$ 1.09	70.85 $\pm$ 1.85	71.84 $\pm$ 2.33
Our - Contrastive	98.59 $\pm$ 0.20	66.93 $\pm$ 1.10	0.468 $\pm$ 0.01	92.84 $\pm$ 0.61	71.86 $\pm$ 0.69	72.83 $\pm$ 2.89
Our - Full Model	98.58 $\pm$ 0.18	67.19 $\pm$ 0.63	0.473 $\pm$ 0.00	93.36 $\pm$ 0.49	72.04 $\pm$ 1.05	74.62 $\pm$ 2.79

### 3.3 CONTRASTIVE LEARNING

**Setup.** In the pairwise ranking setup, each positive sample is only compared to one negative sample at a time. Contrastive learning (Gutmann & Hyvärinen, 2010) makes it general, where a single positive sample can be compared to multiple negative samples, which can be particularly useful in the permuted document task where the number of possible incoherent samples per coherent document can be very large. The number of negatives considered and their quality can affect the model performance (Arora et al., 2019). Wu et al. (2020) show that contrastive loss maximizes a lower bound on the mutual information between representations. A larger number of negatives increases the tightness of the bound; learning with more negatives can better maximise the mutual information. We train our model with a margin-based contrastive loss defined as:

$$\mathcal{L}_\theta(\mathcal{D}^+, \mathcal{D}_1^-, \dots, \mathcal{D}_N^-) = -\log\left(\frac{e^{f_\theta(\mathcal{D}^+)}}{e^{f_\theta(\mathcal{D}^+)} + \sum_{j=1}^N e^{(f_\theta(\mathcal{D}_j^-) - \tau)}}\right) \quad (2)$$

where  $f_\theta(\mathcal{D}^+)$  is the coherence score of the positive document,  $f_\theta(\mathcal{D}_1^-), \dots, f_\theta(\mathcal{D}_N^-)$  are the scores of the  $N$  negative documents, and  $\tau$  is the margin.

**Training.** We use the same training data as the baseline models to train our contrastive model; the positive documents remain the same, while we use 5 negative documents per instance (instead of only 1 in the pairwise setup). Effectively, the model sees the same number of positive or coherent documents, but five times as many negative samples during training compared to the pairwise setting. See Appendix A.4 for the full set of our hyperparameters.

**Results.** From the results in Table 1, we see that the contrastive model (row 3) further improves the results across all the independent test sets; the results on the LMvLM dataset also improve, now surpassing the UNC model performance. Although the improvement on the WSJ permuted document task is small, the improvement in the generalizability of the model is more significant.

### 3.4 MOMENTUM ENCODER WITH HARD NEGATIVE MINING

While increasing the number of negative samples per instance has been shown to be effective for contrastive learning, resource constraints can limit the number of negatives that can be considered per instance. One solution is to consider other positive instances in the same training batch as negatives (Karpukhin et al., 2020; Chen et al., 2020). However, this method is not suitable for the permuted document task since the negatives are instance-specific. While a permuted document is still independently incoherent, training with permuted versions of other documents will not provide the same cues for coherence modeling as the original self-supervision.

Another solution is to maintain a large global queue of negative samples that are independent of the current training instance. During training, negative samples (more specifically, their representations) from the latest batch are enqueued to build a queue upto some size  $l$ . As training continues, the negative samples from the oldest batch are dequeued to accommodate newer samples. However, representations of the documents will evolve through training as the model parameters get updated; this will make the negative samples in the queue inconsistent with each other and the training instances in the current batch. Moreover, the issue of mismatched self-supervision with negatives that are permuted versions of other documents still remains.

**Momentum Encoder.** To address these issues, we add an auxiliary momentum encoder (He et al., 2019), which is also XLNet (Yang et al., 2019). Figure 1 shows the overall architecture. Keeping the

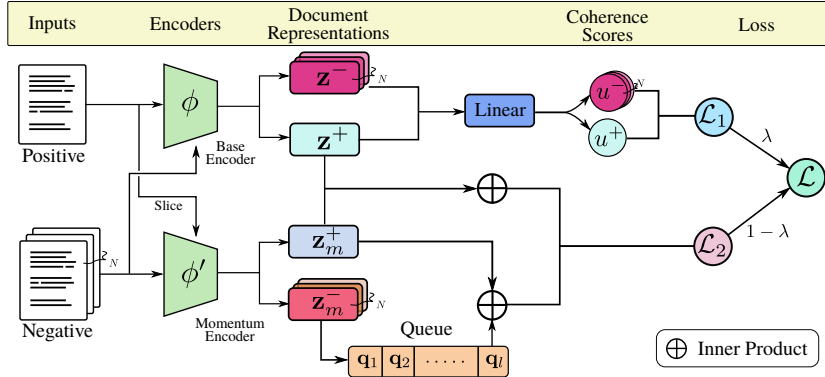


Figure 1: Our coherence model with the auxiliary momentum encoder.  $\phi$  is our base encoder similar to our setup in §3.3, while  $\phi'$  is our momentum encoder.  $u^+ = f_\theta(\mathcal{D}^+)$  and  $u^- = f_\theta(\mathcal{D}^-)$  are the coherence scores of the positive and negative documents respectively. Note that only the parameters of  $\phi$  and the linear layer are updated through backpropagation.

base contrastive setup the same (the upper part), we add an additional contrastive objective based on representations from the momentum encoder. Specifically, we re-encode the positive and negative samples through the momentum encoder; the negative samples thus encoded are used to build the queue. We train the model to promote the similarity between the positive representations from the momentum encoder and the positive representations from our base encoder over the similarity with the negative samples from the queue,  $\mathcal{Q}$ . Specifically, we define a momentum loss  $\mathcal{L}_\theta^{\text{mom}}$  as:

$$c^+ = \frac{(\mathbf{z}^+)^T (\mathbf{z}_m^+)}{\|\mathbf{z}^+\| \|\mathbf{z}_m^+\|}; \quad c_j^- = \frac{(\mathbf{z}_m^+)^T \mathbf{q}_j}{\|\mathbf{z}_m^+\| \|\mathbf{q}_j\|}; \quad \mathcal{L}_\theta^{\text{mom}} = -\log \left( \frac{e^{c^+}}{e^{c^+} + \sum_{j=1}^l e^{(c_j^- - \tau)}} \right) \quad (3)$$

where  $\mathbf{z}^+$  and  $\mathbf{z}_m^+$  are the positive representations from the base encoder ( $\phi$ ) and the momentum encoder ( $\phi'$ ) respectively,  $\mathbf{q}_1, \dots, \mathbf{q}_l$  indexed by  $j$  are the negative representations from  $\phi'$  in the queue, and  $\tau$  is the margin. The momentum encoder  $\phi'$  is updated based on the base encoder  $\phi$  as:

$$\phi' \leftarrow \mu * \phi' + (1 - \mu) * \phi \quad (4)$$

where  $\mu \in [0, 1)$  is the momentum coefficient; only  $\phi$  is updated through backpropagation.

Our full model is trained with a combination of the original contrastive learning objective (Eq. 2) and the momentum encoded contrastive similarity objective (Eq. 3):

$$\mathcal{L}_\theta = \lambda \mathcal{L}_\theta + (1 - \lambda) \mathcal{L}_\theta^{\text{mom}} \quad (5)$$

where  $\lambda$  is a weighting hyperparameter. **The momentum encoder can be considered as a *temporal ensemble* model consisting of exponential-moving-average versions of the base model. Due to this, gradients from the momentum loss (Eq. 3) also help in stabilising the overall training (§4).**

**Length Invariance Training.** In the permuted document task, both the positive and the negative samples have the same number of sentences. This is not necessarily the case for real world applications. In order to incorporate length invariance into our model, we encode a random contiguous slice of the positive document through the momentum encoder  $\phi'$ .<sup>3</sup>

**Hard Negative Mining.** It has been shown that the difficulty of the negative samples used for contrastive training can strongly influence model success (Wu et al., 2020). We therefore automatically mine hard negative samples during training. For the permuted document task, we can take advantage of the fact that the negative sample space can be huge; for a document with  $n$  sentences, the candidate pool of permutations has  $n! - 1$  incoherent documents from which we can mine hard negatives. For the problem of dense text retrieval, Xiong et al. (2020) find *global* hard negatives by computing document encodings using a recent checkpoint to build an asynchronous index of the entire corpus, and sampling negative documents from the index. However, the huge candidate pool for permuted documents also makes it infeasible to mine global negatives in our case.

Instead, we perform *local* negative sample ranking. For each positive instance in the training data, we sample a larger number of permuted documents ( $h$ ) per instance than we need for training (*i.e.*,

<sup>3</sup>Minimum sentence length is 4 and maximum is full document length.

$h > N$ ). We score these negative documents using the model updated thus far and use the highest ranking negative documents for training. Specifically, the model is first trained with  $x$  instances ( $x$  is a hyperparameter) of data, by using 5 negative samples randomly chosen out of  $h$ . The updated model is then used to score all the  $h$  negative samples each for another set of  $x$  instances from the training data. The scores of the  $h$  negative samples are ranked and the top scoring 5 negative samples for each instance are used to train the model for the next  $x$  gradient steps. This process is repeated throughout training; the model therefore iteratively mines harder and harder negative samples as it improves. See Algorithm 1 in Appendix A.2 for the pseudocode.

We use the hard negative training in combination with the momentum encoder since we find that using hard negative samples directly leads to instability in model training (see §4). The global negatives queue  $Q$  is thus also constructed from the mined hard negative samples used for training. Our model is therefore trained to rely not only on comparative coherence cues from the traditional permuted document setup, but also to recognize more independent cues for coherence through the global queue, which is additionally enhanced by incorporating length invariance and automatically mined hard negative samples.

**Training.** We train the model with the same training data, this time sampling  $h = 50$  negatives<sup>4</sup> per instance for hard negative ranking, and setting the training steps (or instances)  $x = 200$ . We use a queue size of  $l = 1000$  and set our momentum coefficient  $\mu = 0.9999999$ , with loss weighting parameter  $\lambda = 0.85$ . Due to GPU memory constraints (24GB, Quadro RTX 6000), we train our model with a batch size of 1. See Appendix A.4 for the full set of hyperparameters.

**Results.** The results in Table 1 (last row) show that our momentum encoder model with hard negative mining outperforms all previous models across the independent testsets. This improvement comes despite a very similar performance on the WSJ test set; we believe that our model truly improves in generalizability without overfitting to the permuted document task. The improvements on the out-of-domain test sets, particularly on LMVLM and STORYCLOZE, support this conclusion.

## 4 ANALYSIS

### 4.1 HARD NEGATIVE TRAINING WITH MOMENTUM MODEL

We only train our complete model (*i.e.*, base contrastive plus momentum model) by mining hard negative samples (§3.4), because we find that training the base contrastive model directly with hard negatives leads to instability during training. Figure 2a plots development set accuracies of our base model trained with and without hard negative mining, and our complete model trained with hard negative mining (evaluated every 1000 steps). As seen in the figure, the contrastive model displays significant volatility when trained with hard negatives only, while the complete model is quite stable. This is inline with the finding of Xuan et al. (2020) who show that training with the hardest negative samples leads to bad local minima. This can be explained with the gradient analysis of such negatives which have a larger gradient norm (Xiong et al., 2020), resulting in abrupt gradient steps. The momentum encoder being a temporal ensemble of the base models has a regularizing effect, addressing this issue and leading to stable and improved results (see §3.4).

### 4.2 EFFECTS OF HYPERPARAMETERS

**Number of Ranked Negatives.** Figure 2b shows the results across the test sets for different numbers of negative samples considered for ranking ( $h$ ) during hard negative mining. We see that increasing the number of negatives considered improves results across the board, with results on out-of-domain test sets LMVLM and STORYCLOZE showing particular improvement.

**Momentum Coefficient.** Figure 2c shows the variation in the model performance across the test sets for different values of the momentum coefficient  $\mu$ . We see that apart from a slight drop on the INSTED-WIKI dataset at  $\mu = 0.9999999$ , overall an increasing  $\mu$  value leads to better generalization on the independent test sets, presumably due to a more consistent global negative queue.

<sup>4</sup>As previously described in §2, we ensure the sampled negative documents are unique even when the positive documents are repeated. This ensures that a much larger sample of the overall candidate pool is considered during training. Since we sample and rank 50 negative documents per positive instance, accounting for 20 repetitions of the positive documents,  $20 * 50 = 1000$  total negative documents are considered for hard negative mining. This is 10 times larger than the contrastive setup (100 unique negatives) and 50 times larger than the pairwise setup (only 20 unique negatives).



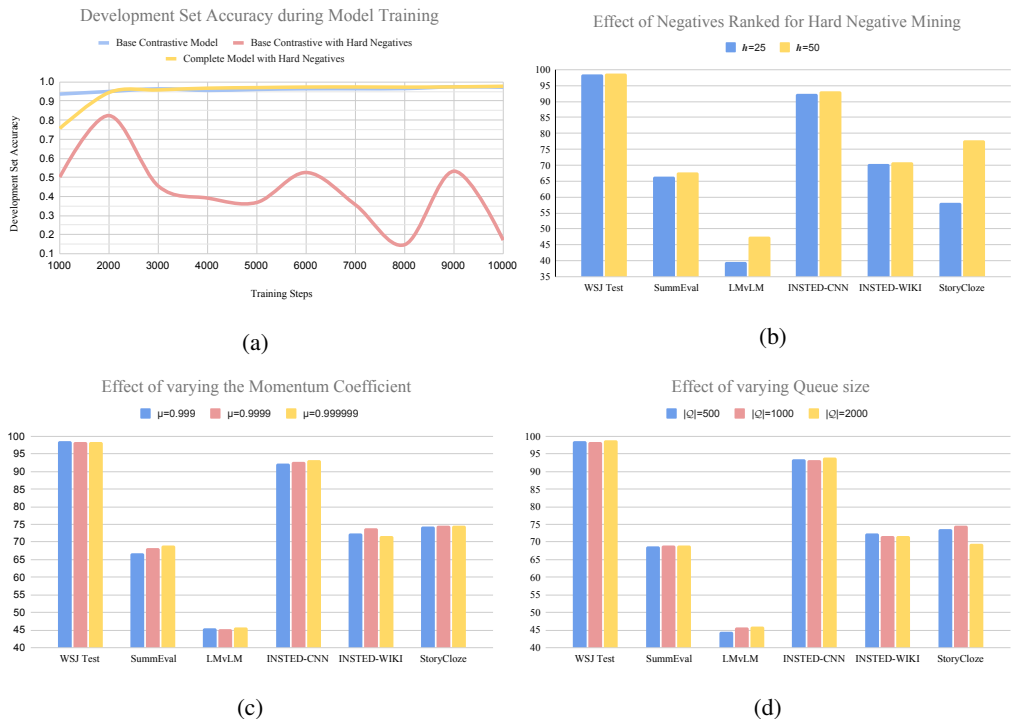


Figure 2: (a) A plot of the development accuracy during training our contrastive model with and without hard negative mining, and our complete model with hard negative mining. The accuracies are evaluated after every 1000 gradient steps. (b) Results on the various test sets for our model trained with hard negative mining by sampling different number of negatives ( $h$ ) for ranking. (c) Results on the various test sets for our complete model trained with different momentum coefficient ( $\mu$ ) values. (d) Results on the various test sets for our model trained with different global queue  $Q$  sizes. Please note that the agreement values for LMvLM test set have been scaled by a factor of 100 to facilitate visualization in figures (b), (c) and (d).

**Queue Size.** Figure 2d shows the variation in model performance across different test sets for various sizes of the global negative queue  $Q$ . We see that while increasing the queue size generally leads to an improvement in scores, at high queue sizes the improvement is limited to test sets from the same domain (WSJ, SUMMEVAL and INSTED-CNN), and the model’s generalizability is affected.

### 4.3 EFFECTS OF VARYING TASK & DATASET

So far, we have reported the results of training our model on the permuted document task using documents from the WSJ corpus as was done by most prior work (Elsner & Charniak, 2011; Moon et al., 2019). We now test the effectiveness of other datasets, both by varying the task itself and by using a different dataset for the permuted document task.

**Sentence Intrusion.** As described in §2.3, Shen et al. (2021) propose a sentence intrusion task to test coherence modeling capabilities of pre-trained language models. We adapt their dataset to the pairwise setting by pairing the original coherent document (positive) with the corrupted (negative) document; setting aside 10% of the data for development gives us 25,852 positive-negative training pairs for INSTED-CNN and 41,135 pairs for INSTED-WIKI. We train our pairwise (§3.2) model on this task. From the results in Table 2 (first two rows), we see that the performance on the same domain/task (as the training) and the performance on the LMvLM dataset is high, but the models trained on this task generalize poorly to the other independent test sets.

**Permuted Document Task with INSTED** We now train our model on the permuted document task using the INSTED datasets. We generate 52,607 and 66,679 positive-negative pairs for INSTED-CNN and INSTED-WIKI respectively by sampling permutations, similar to our training data (see §2.1), and train our pairwise model with this data. The results are shown in Table 2, highlighted in blue. Specifically for machine generated texts, the sentence intrusion task training does better on the LMvLM dataset. On the other hand, the permuted document task training does better on SUMMEVAL. This could be because the documents in SUMMEVAL are summaries of the same



Table 2: Results on the WSJ permuted document test set and other independent test sets on the pairwise and contrastive models trained on different datasets. All results are averaged over 5 runs with different seeds.

Train Dataset	Neg. Type	Model	WSJ	SUMMEVAL	LMVLM	INSTED-CNN	INSTED-WIKI	STORYCLOZE
INSTED-WIKI	Intrusion	Pairwise	95.24 $\pm$ 0.37	53.03 $\pm$ 1.49	0.490 $\pm$ 0.01	94.07 $\pm$ 0.29	82.01 $\pm$ 0.24	64.21 $\pm$ 1.98
INSTED-CNN	Intrusion	Pairwise	95.48 $\pm$ 0.47	57.85 $\pm$ 2.47	0.502 $\pm$ 0.01	97.83 $\pm$ 0.15	73.52 $\pm$ 1.17	71.75 $\pm$ 1.81
INSTED-WIKI	Permuted	Pairwise	96.89 $\pm$ 0.23	64.53 $\pm$ 0.82	0.491 $\pm$ 0.01	84.17 $\pm$ 1.50	71.35 $\pm$ 0.88	69.09 $\pm$ 2.29
INSTED-CNN	Permuted	Pairwise	97.03 $\pm$ 0.12	66.63 $\pm$ 0.97	0.483 $\pm$ 0.01	92.61 $\pm$ 0.62	69.88 $\pm$ 0.64	68.95 $\pm$ 1.02
WSJ	Permuted	Pairwise	98.23 $\pm$ 0.20	64.83 $\pm$ 1.03	0.458 $\pm$ 0.02	91.96 $\pm$ 1.09	70.85 $\pm$ 1.85	71.84 $\pm$ 2.33

Table 3: Accuracies of the best performing UNC and our full model on the hand-crafted linguistic probe dataset constructed by Shen et al. (2021). Examples (abridged for brevity) shown indicate the manual changes made to make the text incoherent; the original words are shown in blue while the modified/added words are shown in red. Checks (✓) indicate our model correctly scored the coherent text higher for that example, while crosses (✗) indicate that our model failed to do so.

Linguistic Probe	UNC	Our	Example
Pronoun Animacy Downgrade	76.0	100.0	✓ She→It was the mother of twins Lakshmana and Shatrughna.
Pronoun Animacy Upgrade	63.0	100.0	✓ It→She has been collected in two tankōbon volumes.
Pronoun Gender Flip	55.0	100.0	✓ She→He is also well known for her→his role as Mary, the mother of Jesus.
Past to Future Flip	86.0	96.0	✗ The Danes finished→will finish first in the 2014 World Junior Hockey Championship.
Single Determiner Flip	62.1	83.2	✗ In 1969, he was again sold, this→these time to the Milwaukee Bucks.
Number	58.0	80.0	✗ He had a career record of 67→6.7 wins and 62→6.2 losses.
Conjunction Flip	55.0	78.0	✗ The school was founded in 1908, and→but has been a non-profit organization since 1956.
Negation	60.0	78.0	✗ He was not named as the Australian squad captain and was not captain of the Wallabies.

source article and therefore similar in content (detecting incoherence through permutations might help here), while the text generated by language models even for the same prompt tends to differ in content more significantly (detecting intruder sentences might help here). Additionally, the performance of our WSJ model on the INSTED-CNN and INSTED-WIKI datasets is comparable to the performance of the respective in-domain pairwise models, while outperforming both the other models on the STORYCLOZE dataset. Overall, the WSJ model generalizes well.

#### 4.4 LINGUISTIC PROBE ANALYSIS

Shen et al. (2021) create eight hand-crafted linguistic probe test sets by manually modifying words in coherent texts based on various linguistic phenomena, ensuring that the incoherent text produced as a result remains syntactically correct. Except for the words targeted by the probe, the rest of the text remains identical. Each test set has 100 samples each.<sup>5</sup>

We evaluate the best performing UNC and our full models on these test sets. The results are shown in Table 3 along with some examples from the dataset. The UNC model has the most success with the tense agreement test set and mixed success on the pronoun test sets. We see that our model has perfect accuracy on all pronoun-related test sets and near-perfect accuracy on the tense agreement test set. This shows that our model is indeed capturing the discourse-level phenomena that constitute coherence. Where our model falters is in cases which may require commonsense knowledge, such as identifying that 6.7 wins is not possible. Overall, our model is quite successful in detecting several kinds of incoherence.

## 5 CONCLUSION

With the goal of making our coherence model generalizable and useful for off-the-shelf evaluations, in this work we have explored self-supervision objectives to improve coherence models without adapting our model architecture to a specific training task like previous work. We upgrade the self-supervision objective from the existing pairwise ranking paradigm to a contrastive learning setup. We further enhance this model with a momentum encoder to maintain a large global queue of negative samples, and also perform hard negative mining to refine the quality of the negative samples. We show empirically that increasing the ratio and quality of negative samples improves the generalizability of the coherence model. We also test our model on a wide-ranging collection of independent test sets that resemble downstream applications, including machine generated text, on which our model significantly outperforms the previous SOTA model. Our work thus also sets a new evaluation standard for future research in coherence modeling. We will open source our code base to encourage research in a new paradigm of coherence modeling.

<sup>5</sup>Except for the test set with determiner flipping, which has 95.

## REPRODUCIBILITY STATEMENT

### CODE AND HYPERPARAMETERS

Code for the various models will be open-sourced. Specific hyperparameters used for experiments are described in §3.3 and §3.4, while a full list of hyperparameters is included in Appendix A.4.

### DATA

A description of the data pre-processing is provided in §2.1. Datasets that we created will be open-sourced. In the case of the WSJ dataset, the data is licensed for use only to members by the Linguistic Data Consortium. Consequently, we only release scripts to generate the data we use and not the data itself. We highlight however that the permuted document self-supervision task that we train on is independent of the dataset used and the task can be reproduced on any other corpus; see also §4.3. All other datasets we use are licensed freely for academic use.

## ETHICS STATEMENT

### ANNOTATION OF LMVLM DATASET

We conduct a user study to collect pairwise coherence judgments on our language model output dataset. As part of our crowd-sourced user study on Amazon Mechanical Turk to collect these coherence judgements, we do not collect any personal information from the participants. Based on the average time spent to perform the tasks, participants were paid the equivalent of 16 USD per hour for their work. The annotation instructions and interface provided to the participants are included in Appendix A.3.

One potential issue is that the language model output that we generate from prompts may lead to malicious text generation by the models. We flagged the task to warn the workers that there may be potentially offensive content, and manually checked the final dataset post curation.

## REFERENCES

- Tushar Abhishek, Daksh Rawat, Manish Gupta, and Vasudeva Varma. Transformer models for text coherence assessment. *ArXiv*, abs/2109.02176, 2021.
- Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5628–5637. PMLR, 09–15 Jun 2019. URL <http://proceedings.mlr.press/v97/saunshi19a.html>.
- R. Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34:1–34, 2008.
- Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL ’05*, pp. 141–148, Ann Arbor, Michigan, 2005. Association for Computational Linguistics.
- T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, T. Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.
- Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume*

- I (Long Papers)*, pp. 1662–1675, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1150. URL <https://www.aclweb.org/anthology/N18-1150>.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.
- Zihang Dai, Z. Yang, Yiming Yang, J. Carbonell, Quoc V. Le, and R. Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL*, 2019.
- Micha Elsner and Eugene Charniak. 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, pp. 125–129, Portland, Oregon, 2011. Association for Computational Linguistics.
- Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation. *arXiv preprint arXiv:2007.12626*, 2020.
- Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1082. URL <https://www.aclweb.org/anthology/P18-1082>.
- Vanessa Wei Feng and Graeme Hirst. 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, pp. 315–324, Avignon, France, 2012. Association for Computational Linguistics.
- Vanessa Wei Feng, Ziheng Lin, and Graeme Hirst. The impact of deep hierarchical discourse structures in the evaluation of text coherence. In *COLING*, 2014.
- B. Grosz and C. Sidner. Attention, intentions, and the structure of discourse. *Comput. Linguistics*, 12:175–204, 1986.
- B. Grosz, A. Joshi, and S. Weinstein. Centering: A framework for modeling the local coherence of discourse. *Comput. Linguistics*, 21:203–225, 1995.
- Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterton (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL <http://proceedings.mlr.press/v9/gutmann10a.html>.
- Michael Halliday and Ruqaiya Hasan. *Cohesion in English*, chapter xx. Longman, London, 1976.
- Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan R. Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, T. M. Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. Achieving human parity on automatic chinese to english news translation. *ArXiv*, abs/1803.05567, 2018.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.
- K. Hermann, Tomáš Kociský, Edward Grefenstette, Lasse Espeholt, W. Kay, Mustafa Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In *NIPS*, 2015.

- Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. A simple language model for task-oriented dialogue, 2020.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering. *ArXiv*, abs/2004.04906, 2020.
- K. Krippendorff. Computing krippendorff’s alpha-reliability. 2011.
- Samuel Läubli, Rico Sennrich, and Martin Volk. Has machine translation achieved human parity? a case for document-level evaluation. In *EMNLP*, 2018.
- Jiwei Li and Dan Jurafsky. Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 198–209, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
- Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 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*, pp. 997–1006, Portland, Oregon, 2011. Association for Computational Linguistics.
- Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li. Generative adversarial network for abstractive text summarization. *ArXiv*, abs/1711.09357, 2017.
- Mohsen Mesgar and Michael Strube. A neural local coherence model for text quality assessment. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4328–4339, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/D18-1464>.
- Muhammad Tasnim Mohiuddin, Shafiq Joty, and Dat Tien Nguyen. 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)*, pp. 558–568, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/P18-1052>.
- Tasnim Mohiuddin, Prathyusha Jwalapuram, Xiang Lin, and Shafiq Joty. Rethinking coherence modeling: Synthetic vs. downstream tasks. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 3528–3539, Online, April 2021. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/2021.eacl-main.308>.
- Han Cheol Moon, Tasnim Mohiuddin, Shafiq R. Joty, and Xiaofei Chi. A unified neural coherence model. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 2262–2272, 2019. URL <https://www.aclweb.org/anthology/D19-1231.pdf>.
- Dat Nguyen and Shafiq Joty. A neural local coherence model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1320–1330. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1121. URL <http://www.aclweb.org/anthology/P17-1121>.
- Romain Paulus, Caiming Xiong, and R. Socher. A deep reinforced model for abstractive summarization. *ArXiv*, abs/1705.04304, 2018.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.
- L. Pishdad, Federico Fancellu, Ran Zhang, and A. Fazly. How coherent are neural models of coherence? In *COLING*, 2020.
- M. Popel, M. Tomková, J. Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondrej Bojar, and Z. Žabokrtský. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11, 2020.

- Alec Radford, Jeffrey Wu, Dario Amodei, Daniela Amodei, Jack Clark, Miles Brundage, and Ilya Sutskever. Better language models and their implications. *OpenAI Blog*, 2019. URL <https://openai.com/blog/better-language-models/>.
- Ehud Reiter. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401, 2018.
- Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. An entity-driven framework for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3271–3282, 2019.
- Rishi Sharma, J. Allen, Omid Bakhshandeh, and N. Mostafazadeh. Tackling the story ending biases in the story cloze test. In *ACL*, 2018.
- Aili Shen, Meladel Mistica, Bahar Salehi, Hang Li, Timothy Baldwin, and Jianzhong Qi. Evaluating Document Coherence Modeling. *Transactions of the Association for Computational Linguistics*, 9:621–640, 07 2021. ISSN 2307-387X. doi: 10.1162/tacl\_a.00388. URL [https://doi.org/10.1162/tacl\\_a\\_00388](https://doi.org/10.1162/tacl_a_00388).
- Aäron van den Oord, Y. Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.
- M. Wu, Chengxu Zhuang, M. Mosse, D. Yamins, and Noah D. Goodman. On mutual information in contrastive learning for visual representations. *ArXiv*, abs/2005.13149, 2020.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *ICLR*, abs/2007.00808, 2020.
- Peng Xu, Hamidreza Saghir, Jin Sung Kang, Teng Long, Avishek Joey Bose, Yanshuai Cao, and Jackie Chi Kit Cheung. A cross-domain transferable neural coherence model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 678–687, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1067.
- Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. Hard negative examples are hard, but useful. In *ECCV*, 2020.
- Z. Yang, Zihang Dai, Yiming Yang, J. Carbonell, R. Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, 2019.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. Dialogpt: Large-scale generative pre-training for conversational response generation. In *ACL, system demonstration*, 2020.

## A APPENDIX

### A.1 WSJ PERMUTED DOCUMENT TASK

The examples for the permuted document task on the WSJ data are shown in Table 4.

### A.2 HARD NEGATIVE RANKING PSEUDOCODE

The pseudocode for our hard negative mining through local sample ranking is given in Algorithm 1.

Table 4: Examples showing the original coherent document and the incoherent document created by permuting the sentences of the original. Text taken from WSJ-1778.

Original Document
(S1) Judy and I were in our back yard when the lawn started rolling like ocean waves.
(S2) We ran into the house to get Mame, but the next tremor threw me in the air and bounced me as I tried to get to my feet.
(S3) We are all fine here, although Mame was extremely freaked.
(S4) Books and tapes all over my room.
(S5) Not one thing in the house is where it is supposed to be, but the structure is fine.
Permuted Document
(S4) Books and tapes all over my room.
(S3) We are all fine here, although Mame was extremely freaked.
(S2) We ran into the house to get Mame, but the next tremor threw me in the air and bounced me as I tried to get to my feet.
(S5) Not one thing in the house is where it is supposed to be, but the structure is fine.
(S1) Judy and I were in our back yard when the lawn started rolling like ocean waves.

**Algorithm 1** Local Negative Sample Ranking

**Require:** Training data  $D$  in which each instance consists of a positive document and  $h$  negative documents, model  $\theta$

- 1: Initialize empty hard negative array  $\hat{D}^-$  for each instance  $\in D$
- 2: **procedure** HARDNEGATIVERANKING( $\theta, D$ )
- 3:   Partition the dataset into sets of  $x$  instances  $D_1 \dots D_r$
- 4:   **for**  $i = 1 \dots r$  **do**
- 5:     **if**  $i == 0$  **then** ▷ No hard negatives for first iteration
- 6:       **for**  $j = 1 \dots x$  **do**
- 7:          Randomly sample  $N$  negatives from  $D_{(i,j)}^-$  and store in  $\hat{D}_{(i,j)}^-$
- 8:   Train  $\theta$  with  $(D_i^+, \hat{D}_i^-)$
- 9:   **for**  $j = 1 \dots x$  **do**
- 10:     Score all the  $h$  negative documents in  $D_{(i+1,j)}^-$
- 11:     Sort  $D_{(i+1,j)}^-$  in descending order of scores
- 12:     Get  $N$  top scoring negative documents and store in  $\hat{D}_{(i+1,j)}^-$
- 13:     ▷ Store hard negatives for the next iteration

## A.3 LMVLM USER STUDY

The instructions and the interface provided to the workers in the user study comparing pairs of language model outputs is given in Figure 3. Workers were restricted to the native English speaking regions of Canada, United Kingdom and the United States and could only participate in our task if they had completed  $> 10,000$  HITs with a  $> 98\%$  acceptance rate. Each task was estimated to take 2 minutes, and workers were paid the equivalent of 16 USD per hour.

## A.4 HYPERPARAMETERS

The hyperparameters used in our experiments are given in Table 5.

## A.5 COMPARISON OF EXISTING STATE-OF-THE-ART COHERENCE MODELS

We report the results obtained by Mohiuddin et al. (2021) and Pishdad et al. (2020) on their evaluation tasks for SOTA neural coherence models in Table 6.

**Coherence** is a property of a well-written text that makes it different from a random set of sentences: sentences (or clauses) in coherent texts are related to nearby sentences in systematic ways. For example, consider:

a. Jane took a train from Paris to Istanbul. She had to attend a conference.

This is an example of a **coherent** text. Here, the second sentence gives a reason for Jane's action in the first sentence.

b. John took a train from Paris to Istanbul. He hates spinach.

This example is **incoherent**, because it is unclear to the reader why the second sentence follows the first. The reader might have to go through some effort to figure out what this text could be trying to convey.

Another indication of coherence in texts is when a text is consistently talking about someone or something. Consider this example:

c. John wanted to buy a piano. Jenny also wanted to buy a piano. He went to the piano store. It was nearby. The living room was on the second floor. She didn't find anything she liked. The piano he bought was hard to get up to that floor.

Here the text switches from being about John, to Jenny, to the piano store, John's living room, Jenny and the piano again, making the text hard to follow and therefore **incoherent**.

In this task, you will be shown a short text which is meant to be a writing prompt. Two candidate texts which are continuations of the writing prompt will also be presented. You have to indicate which text out of the two given texts is **more coherent**, based on the explanation of coherence provided to you and the general quality of the text.

In rare cases, you may not be able to decide if one text is more coherent than another; in such cases you may choose the option that they are equally coherent/incoherent. However, please use this option sparingly, and only if there is absolutely no difference in coherence between the two texts.

**Writing Prompt Sample #** `#{post_id}`


Here is the writing prompt that the following texts are meant to be continuations of:

`$(prompt)`

**Based on the prompt and the instructions provided about the concept of coherence, please judge which of the following two continuation texts is better. Please only use the "equally coherent/incoherent" option if there is absolutely no difference in coherence between the texts.**

**A:** `$(textA)`

**B:** `$(textB)`

 **Previewing Answers Submitted by Workers** ×

This message is only visible to you and will not be shown to Workers.  
You can test completing the task below and click "Submit" in order to preview the data and format of the submitted results.

Text A is more coherent

Text B is more coherent

Both Text A and Text B are equally coherent/incoherent

Figure 3: Instructions and study interface for the user study conducted on language model outputs.



Table 5: Configuration parameters for training

Parameters	Values
<b>Margin-based Pairwise Ranking</b>	
- margin	0.1
- optimizer	AdamW
- scheduler	SWALR
- learning rate	5e-6
- annealed to	1e-6
- anneal rate	5000 steps
- batch-size	1
- XLNet model	base
- dimension size	768
<b>Contrastive Learning</b>	
- margin	0.1
- optimizer	AdamW
- scheduler	SWALR
- learning rate	5e-6
- annealed to	1e-6
- anneal rate	5000 steps
- batch-size	1
- XLNet model	base
- dimension size	768
<b>Momentum Encoder with Hard Negative Mining</b>	
- margin	0.1
- optimizer	AdamW
- scheduler	SWALR
- learning rate	5e-6
- annealed to	1e-6
- anneal rate	1000 steps
- batch-size	1
- XLNet model	base
- dimension size	768

Table 6: Results reported by Mohiuddin et al. (2021) and Pishdad et al. (2020) on various tasks and datasets that compare the UNC model to two other SOTA neural coherence models proposed by Xu et al. (2019) and Mesgar & Strube (2018). Except those marked by (Agr.) which report agreement with humans, all other tasks report accuracies. We only include tasks that directly test discourse coherence phenomena.

As reported by Pishdad et al. (2020)			
Task	Dataset	UNC	Mesgar & Strube (2018)
Permuted Document	Visual Storytelling	<b>88.42</b>	82.25
Permuted Document	ROCStories	<b>94.80</b>	89.55
Permuted Document	Dialogue	<b>97.21</b>	90.79
Permuted Document	HellaSwag	<b>83.92</b>	69.38
Permuted Document	PDTB	<b>92.85</b>	61.96
Connective Substitution	PDTB	<b>96.46</b>	84.99
Topic Switching	Visual Storytelling	<b>92.10</b>	64.81
Topic Switching	ROCStories	<b>94.62</b>	67.85
Topic Switching	Dialogue	<b>71.74</b>	68.41
Topic Switching	PDTB	<b>70.89</b>	52.33
As reported by Mohiuddin et al. (2021)			
Task	Dataset	UNC	Xu et al. (2019)
Permuted Document	WSJ	<b>93.19</b>	91.77
Abstractive Summarization (Agr.)	CNN	<b>0.68</b>	0.55
Extractive Summarization (Agr.)	DUC	0.35	<b>0.38</b>
Machine Translation (Agr.)	WMT	0.77	<b>0.78</b>
(Trained) Machine Translation (Agr.)	WMT	<b>0.83</b>	0.75