# Rethinking Self-Supervision Objectives for Generalizable Coherence Modeling

**Prathyusha Jwalapuram**[†] and **Shafiq Joty**[†§] and **Xiang Lin**[†]
[†]Nanyang Technological University, Singapore
[§]Salesforce Research Asia, Singapore
{jwal0001,srjoty,linx0057}@ntu.edu.sg

## Abstract

Given the claims of improved text generation quality across various pre-trained neural models, we consider the coherence evaluation of machine generated text to be one of the principal applications of coherence models that needs to be investigated. Prior work in neural coherence modeling has primarily focused on devising new architectures for solving the permuted document 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 applications and show significant improvements in coherence evaluations of downstream tasks.[1]

## 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 and Hasan, 1976; Grosz and Sidner, 1986). **Coherence models** (Barzilay and 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 recent advancements in neural methods, 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 using the Penn Treebank (WSJ) corpus. 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 and Lapata (2005, 2008) was extended to incorporate entity-specific features (Elsner and Charniak, 2011), multiple ranks (Feng and Hirst, 2012), and coherence relations (Lin et al., 2011; Feng et al., 2014). Their neural extensions have also been proposed (Nguyen and 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.

[1] Our code and data are available at https://ntunlpsg.github.io/project/coherence-paradigm

*Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*
*Volume 1: Long Papers*, pages 6044 - 6059
May 22-27, 2022

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 coherence modeling capabilities.

In light of these recent findings, our aim is to propose a coherence model that generalizes well to downstream tasks. We train our model purely through *self-supervision*, without tailoring the model architecture specifically to the permuted document task or any other form of supervision.

Li and 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 and 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) 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 automatically mined hard negatives, coupled with a large global negative queue encoded by a momentum encoder (He et al., 2019).

In summary, our contributions are:

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

## 2 Datasets

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. 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 downstream tasks 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 divided into 1240/138/1053 documents for training/development/testing in the standard setup. We exclude documents with < 4 sentences and truncate them to a maximum length of 600 tokens. 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 larger training set. Moon et al. (2019) use 20 permutations of a document for training; since their setup is pairwise, it means the original positive document is repeated 20x. We regenerate the permuted documents similarly, sampling a larger set of permutations for our contrastive learning setup.[2] This gives us $46,522$ instances of positive and 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 $20,411$ pairs 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

[2] 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.

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 $1,046$ 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 handcrafted 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.

## 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. 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 $(s_1, s_2, \ldots, s_n)$ as input, our model uses the representations obtained through XLNet (parameterized by $\phi$) to assign a coherence score to the model. Specifically, for each sentence $s_i$ with $k$ tokens $(w_1, w_2 \ldots w_k)$, 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:

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

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.

**Baselines.** We compare our models against all three versions of the **L**ocal **C**oherence **D**iscriminator or LCD model (Xu et al., 2019)[3]: (*i*) LCD-G, that uses GloVe (Pennington et al., 2014) representations, (*ii*) LCD-I, that uses InferSent (Conneau et al., 2017) representations, and (*iii*) LCD-L, that uses representations from an RNN-based language model trained on the training data. We also compare against the **Un**ified **C**oherence model or UNC (Moon et al., 2019)[4], which is the previous SOTA on the WSJ permuted document task. Results from evaluation of existing coherence models by Pishdad et al. (2020) and Mohiuddin et al. (2021) indicate that UNC and LCD are the best-performing models (see Appendix A.4 for a full comparison). We retrain their models with our training data for comparison. 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.** The results for the baseline models are given in Table 1 (see top five rows). We see that despite accuracies of more than 90% on the WSJ permuted document task, the LCD models perform only a little above a random baseline of 50% on most of the independent test sets, with LCD-G being the best generalizing model out of the three. Similarly, despite a relatively high performance on the WSJ test set (94.11%), UNC's performance on the independent test sets is quite poor, even failing to do better than the random baseline of 50% in two out of five cases. Both the LCD and UNC models have slightly better success on the INSteD-CNN dataset, which is the same domain (news) as the training data, with the UNC model reaching 67.21% accuracy. Our XLNet-Pairwise model trained without fine-tuning the representations (No FT) performs no better than the baseline models. This shows that both the LCD-G and the UNC models are in fact strong baselines despite using GloVe and ELMo (Peters et al., 2018) pre-trained representations respectively.

Our fully-trained XLNet-Pairwise model not only outperforms the UNC model on the standard WSJ permuted document task, but also significantly outperforms all baseline models 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.

### 3.3 Contrastive Learning

**Setup.** In pairwise ranking, each positive sample is only compared to one negative at a time. Contrastive learning (Gutmann and Hyvärinen, 2010) makes it general, where a single positive sample can be compared to multiple negatives, 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 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:

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

[3] https://github.com/BorealisAI/cross_domain_coherence

[4] https://github.com/taasnim/unified-coherence-model

| Model | WSJ | SummEval | LMvLM | INSteD-CNN | INSteD-Wiki | StoryCloze |
|---|---|---|---|---|---|---|
| LCD-G | $90.39_{\pm 0.28}$ | $54.15_{\pm 0.83}$ | $0.419_{\pm 0.00}$ | $61.24_{\pm 0.71}$ | $55.09_{\pm 0.46}$ | $51.76_{\pm 1.22}$ |
| LCD-I | $91.56_{\pm 0.16}$ | $51.71_{\pm 0.99}$ | $0.420_{\pm 0.01}$ | $60.23_{\pm 0.86}$ | $53.50_{\pm 0.37}$ | $52.69_{\pm 0.69}$ |
| LCD-L | $90.24_{\pm 0.36}$ | $53.56_{\pm 1.20}$ | $0.404_{\pm 0.01}$ | $55.07_{\pm 0.26}$ | $51.04_{\pm 0.47}$ | $50.09_{\pm 1.57}$ |
| 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 | $98.58_{\pm 0.18}$ | $67.19_{\pm 0.63}$ | $0.473_{\pm 0.00}$ | $93.36_{\pm 0.49}$ | $72.04_{\pm 1.05}$ | $74.62_{\pm 2.79}$ |

Table 1: Results on the WSJ permuted document test set and the various independent test sets of LCD GloVe (LCD-G), LCD Infersent (LCD-I), LCD Language Model (LCD-L), UNC, and our XLNet based models. The XLNet representations are not fine-tuned during training for our Pairwise (No FT) model. 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.

where $f_\theta(\mathcal{D}^+)$ is the coherence score of the positive document, $f_\theta(\mathcal{D}_1^-), \cdots, f_\theta(\mathcal{D}_N^-)$ 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. Appendix A.5 gives the full set of hyperparameters.

**Results.** From the results in Table 1, we see that the contrastive model (second to last row) further improves the results across all the independent test sets; the results on the LMvLM dataset also improve, 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 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; Huang et al., 2020). We therefore automatically mine hard negatives 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.,* $h > N$). We score these negative documents using the model updated thus far and use the highest ranking negatives 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.

In practice however, we find that using hard negative samples directly leads to instability in model training (see §4.1). We therefore use hard negative training in combination with a momentum encoder, which we describe in the next subsection.

### 3.5 Hard Negatives with Momentum Encoder

While increasing the number of negative samples per instance has been shown to be effective for constrastive 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 nega-

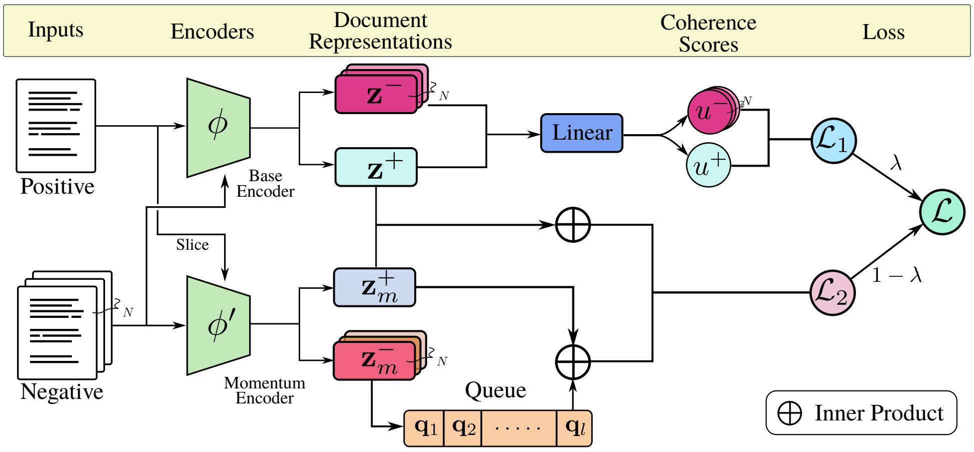


Figure 1: Our coherence model with the auxiliary momentum encoder. $\phi$ is our base encoder similar to our setup in §3.3, while $\phi'$ is our momentum encoder. $u^+ = f_\theta(\mathcal{D}^+)$ and $u^- = f_\theta(\mathcal{D}^-)$ are the coherence scores of the positive and negative documents respectively.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

tives (Karpukhin et al., 2020; Chen et al., 2020). However, it is not suitable for the permuted document task since the negatives are instance-specific. While a permuted document is still independently incoherent, training with permutations 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 (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 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, $\boldsymbol{Q}$. Specifically, we define a momentum loss $\mathcal{L}_\theta^{\text{mom}}$ as:

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

where $\mathbf{z}^+$ and $\mathbf{z}_m^+$ are the positive representations from the base encoder ($\phi$) and the momentum encoder ($\phi'$) respectively, $\mathbf{q}_1, \ldots, \mathbf{q}_l$ 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:

$$\phi' \leftarrow \mu * \phi' + (1-\mu) * \phi \tag{4}$$

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 objective (Eq. 2) and the momentum encoded contrastive similarity objective (Eq. 3):

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

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

**Length Invariance.** 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 downstream applications. To incorporate length invariance into our model, we encode a random contiguous slice of the positive document through the momentum encoder $\phi'$.[5]

The global negatives queue $\boldsymbol{Q}$ is 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[6] 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.5 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

We only train our complete model (*i.e.,* base contrastive plus momentum model) by mining hard negative samples (§3.5), 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 §3.5).

### 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.

**Queue Size.** Figure 2d shows the variation in model performance across different test sets for various sizes of the global negative queue $\boldsymbol{Q}$. 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

[5] Minimum is 4 and maximum is full document.

[6] 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).

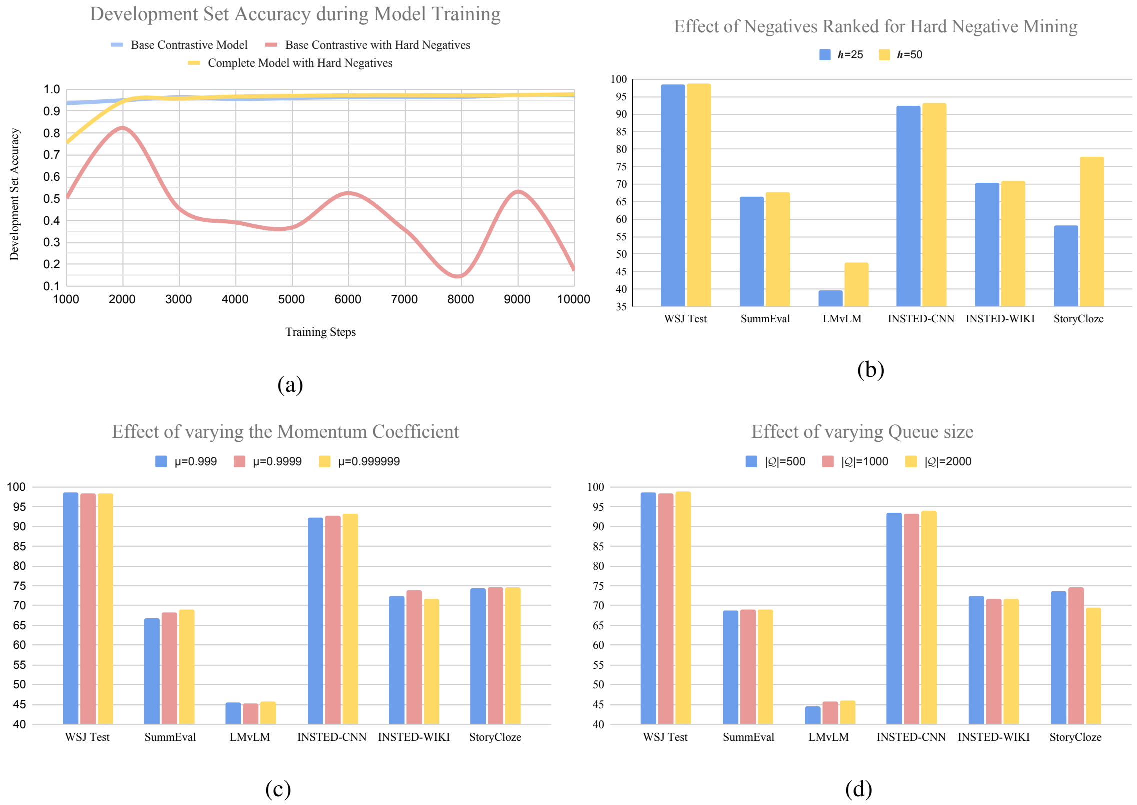


Figure 2: (a) A plot of the development accuracy during training our contrastive model with and without hard negative mining, and our complete model with hard negative mining. The accuracies are evaluated after every 1000 gradient steps. (b) Results on the various test sets for our model trained with 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 $\boldsymbol{Q}$ 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).

most prior work (Elsner and Charniak, 2011; Moon et al., 2019). We now test the effectiveness of other datasets, 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 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. Specifically for machine generated texts, results in Table 2 show that 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 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

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

| Linguistic Probe | LCD | UNC | Our | | Example |
|---|---|---|---|---|---|
| Pronoun Animacy Downgrade | 87.0 | 76.0 | 100.0 | ✔ | She→It was the mother of twins Lakshmana and Shatrughna. |
| Pronoun Animacy Upgrade | 46.0 | 63.0 | 100.0 | ✔ | It→She has been collected in two tankōbon volumes. |
| Pronoun Gender Flip | 49.0 | 55.0 | 100.0 | ✔ | She→He is also well known for her→his role as Mary, the mother of Jesus. |
| Past to Future Flip | 68.0 | 86.0 | 96.0 | ✘ | The Danes finished→will finish first in the 2014 World Junior Hockey Championship. |
| Single Determiner Flip | 57.9 | 62.1 | 83.2 | ✘ | In 1969, he was again sold, this→these time to the Milwaukee Bucks. |
| Number | 56.0 | 58.0 | 80.0 | ✘ | He had a career record of 67→6.7 wins and 62→-6.2 losses. |
| Conjunction Flip | 54.0 | 55.0 | 78.0 | ✘ | The school was founded in 1908, and→but has been a non-profit organization since 1956. |
| Negation | 46.0 | 60.0 | 78.0 | ✘ | He was not named as the Australian squad captain and was not captain of the Wallabies. |

Table 3: Accuracies of the best performing LCD-G, 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.

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 model trained on the WSJ permuted document task generalizes well.

### 4.4 Linguistic Probe Analysis

Shen et al. (2021) create 8 hand-crafted linguistic probe test sets by manually modifying words in coherent texts based on various linguistic phenomena, ensuring 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.[7]

We evaluate the best performing LCD-G, UNC and our full models on these test sets. The results are shown in Table 3 along with some examples from the dataset. The LCD-G model has mixed success across the test sets. The UNC model has the most success with the tense agreement test set and is moderately successful 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

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 open source our code base to encourage research in a new paradigm of coherence modeling.

## Acknowledgements

We would like to thank the Senior Area Chairs of ACL 2022 for evaluating our paper on its merits, and the reviewers and meta-reviewer of ARR for their reviews. We would also like to thank our colleagues Mathieu Ravaut and Han Cheol Moon for their valuable inputs.

[7] Except for Single Determiner Flip, which has 95.

## Ethics Statement

### 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.

### 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.

### Applicability Across Languages

All our experiments are conducted using data for the English language. However, as coherence and discourse relations in text are a universal concept, and our training data is automatically generated, we expect the permuted document task to be easily extensible to other languages.

## References

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. 2019. A theoretical analysis of contrastive unsupervised representation learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5628–5637. PMLR.

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

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

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. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. 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 1 (Long Papers)*, pages 1662–1675, New Orleans, Louisiana. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709.

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

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*.

Zihang Dai, Z. Yang, Yiming Yang, J. Carbonell, Quoc V. Le, and R. Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL*.

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

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2020. Summeval: Re-evaluating summarization evaluation. *arXiv preprint arXiv:2007.12626*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

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

Vanessa Wei Feng, Ziheng Lin, and Graeme Hirst. 2014. The impact of deep hierarchical discourse structures in the evaluation of text coherence. In *COLING*.

B. Grosz and C. Sidner. 1986. Attention, intentions, and the structure of discourse. *Comput. Linguistics*, 12:175–204.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy. PMLR.

Michael Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*, chapter xx. Longman, London.

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. 2018. Achieving human parity on automatic chinese to english news translation. *ArXiv*, abs/1803.05567.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2019. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*.

K. Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, W. Kay, Mustafa Suleyman, and P. Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery  Data Mining*, KDD '20, page 2553–2561, New York, NY, USA. Association for Computing Machinery.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *ArXiv*, abs/2004.04906.

K. Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *EMNLP*.

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

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2011. Automatically evaluating text coherence using discourse relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 997–1006, Portland, Oregon. Association for Computational Linguistics.

Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li. 2017. Generative adversarial network for abstractive text summarization. *ArXiv*, abs/1711.09357.

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

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

Tasnim Mohiuddin, Prathyusha Jwalapuram, Xiang Lin, and Shafiq Joty. 2021. 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*, pages 3528–3539, Online. Association for Computational Linguistics.

Han Cheol Moon, Tasnim Mohiuddin, Shafiq R. Joty, and Xiaofei Chi. 2019. 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*, pages 2262–2272.

Dat Nguyen and Shafiq Joty. 2017. A neural local coherence model. In *Proceedings of the 55th Annual*

*Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1320–1330. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and R. Socher. 2018. A deep reinforced model for abstractive summarization. *ArXiv*, 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.

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

L. Pishdad, Federico Fancellu, Ran Zhang, and A. Fazly. 2020. How coherent are neural models of coherence? In *COLING*.

M. Popel, M. Tomková, J. Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondrej Bojar, and Z. Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11.

Alec Radford, Jeffrey Wu, Dario Amodei, Daniela Amodei, Jack Clark, Miles Brundage, and Ilya Sutskever. 2019. Better language models and their implications. *OpenAI Blog*.

Ehud Reiter. 2018. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.

Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. 2019. 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)*, pages 3271–3282.

Rishi Sharma, J. Allen, Omid Bakhshandeh, and N. Mostafazadeh. 2018. Tackling the story ending biases in the story cloze test. In *ACL*.

Aili Shen, Meladel Mistica, Bahar Salehi, Hang Li, Timothy Baldwin, and Jianzhong Qi. 2021. Evaluating Document Coherence Modeling. *Transactions of the Association for Computational Linguistics*, 9:621–640.

Aäron van den Oord, Y. Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

M. Wu, Chengxu Zhuang, M. Mosse, D. Yamins, and Noah D. Goodman. 2020. On mutual information in contrastive learning for visual representations. *ArXiv*, abs/2005.13149.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *ICLR*, abs/2007.00808.

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

Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. 2020. Hard negative examples are hard, but useful. In *ECCV*.

Z. Yang, Zihang Dai, Yiming Yang, J. Carbonell, R. Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation. In *ACL, system demonstration*.

# A Appendix

## A.1 WSJ Permuted Document Task

The examples for the permuted document task on the WSJ data are shown in Table 5.

## A.2 Hard Negative Ranking Pseudocode

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

## 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 Comparison of Existing State-of-The-Art Coherence Models

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

## A.5 Hyperparameters

The hyperparameters used in our experiments are given in Table 4.

| Parameters | Values |
|---|---|
| **Margin-based Pairwise Ranking (without XLnet fine-tuning)** | |
| - 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 |
| **Margin-based Pairwise Ranking** | |
| - 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 |
| **Contrastive Learning** | |
| - 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 |
| **Momentum Encoder with Hard Negative Mining** | |
| - 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 |

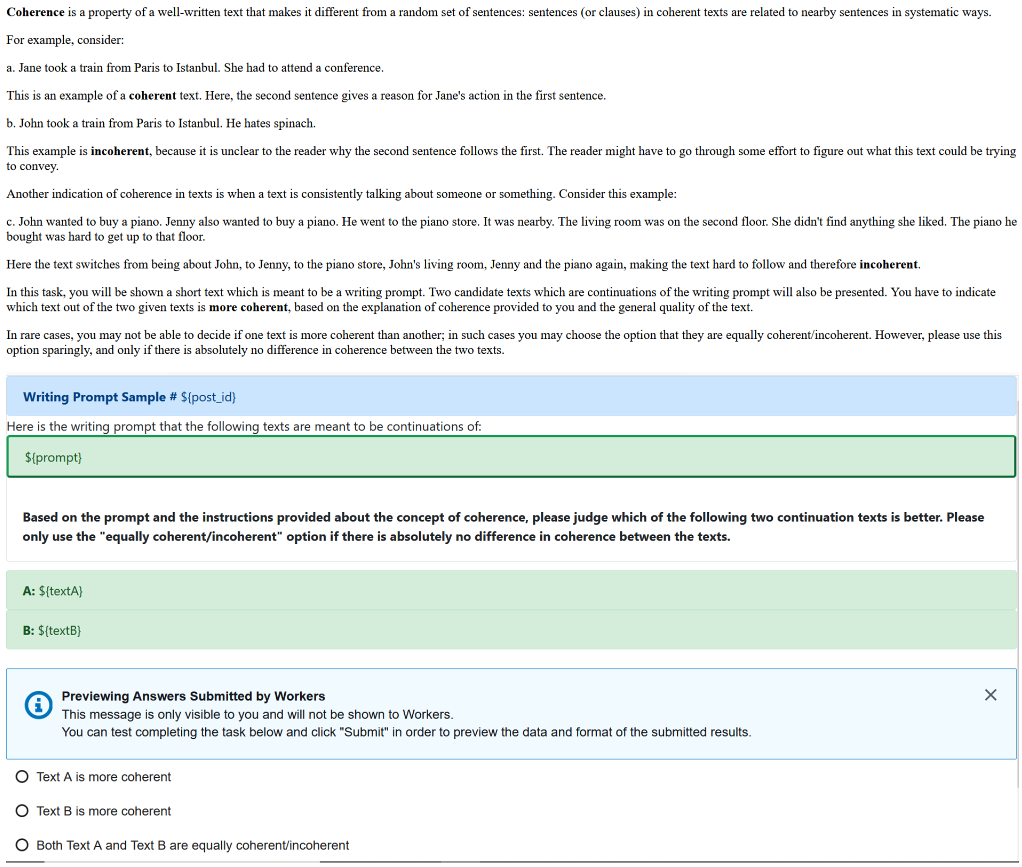
Table 4: Configuration parameters for training

| | 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. |

Table 5: Examples showing the original coherent document and the incoherent document created by permuting the sentences of the original. Text taken from WSJ-1778.

**Algorithm 1** Local Negative Sample Ranking

```
Require: Training data $D$ in which each instance consists of a positive document and $h$ negative
    documents, model $\theta$
Initialize empty hard negative array $\hat{D}^-$ for each instance $\in D$
procedure HARDNEGATIVERANKING($\theta, D$)
    Partition the dataset into sets of $x$ instances $D_1 \dots D_r$
    for $i = 1 \dots r$ do
        if i==0 then                                              ▷ No hard negatives for first iteration
            for $j = 1 \dots x$ do
                Randomly sample $N$ negatives from $D^-_{(i,j)}$ and store in $\hat{D}^-_{(i,j)}$
        Train $\theta$ with $(D^+_i, \hat{D}^-_i)$
        for $j = 1 \dots x$ do
            Score all the $h$ negative documents in $D^-_{(i+1,j)}$
            Sort $D^-_{(i+1,j)}$ in descending order of scores
            Get $N$ top scoring negative documents and store in $\hat{D}^-_{(i+1,j)}$
                                                                  ▷ Store hard negatives for the next iteration
```



**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.

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

Table 6: Results reported by Mohiuddin et al. (2021) and Pishdad et al. (2020) on various tasks and datasets that compare the Moon et al. (2019) (UNC) model to two other SOTA neural coherence models proposed by Mesgar and Strube (2018) and Xu et al. (2019) (LCD). 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.