Is Everything in Order? A Simple Way to Order Sentences

Somnath Basu Roy Chowdhury[1*] **Faeze Brahman**[2*] **Snigdha Chaturvedi**[1]
[1]UNC Chapel Hill, [2]University of California, Santa Cruz
{somnath, snigdha}@cs.unc.edu fbrahman@ucsc.edu

Abstract

The task of organizing a shuffled set of sentences into a coherent text has been used to evaluate a machine's understanding of causal and temporal relations. We formulate the *sentence ordering* task as a conditional text-to-marker generation problem. We present **Re**order-**BART** (RE-BART) that leverages a pre-trained Transformer-based model to identify a coherent order for a given set of shuffled sentences. The model takes a set of shuffled sentences with sentence-specific markers as input and generates a sequence of position markers of the sentences in the ordered text. RE-BART achieves the state-of-the-art performance across 7 datasets in Perfect Match Ratio (PMR) and Kendall's tau (τ). We perform evaluations in a zero-shot setting, showcasing that our model is able to generalize well across other datasets. We additionally perform several experiments to understand the functioning and limitations of our framework.

1 Introduction

Constructing coherent text requires an understanding of entities, events, and their relationships. Automatically understanding such relationships among nearby sentences in a multi-sentence text has been a longstanding challenge in NLP.

Sentence ordering task was proposed to test the ability of automatic models to reconstruct a coherent text given a set of shuffled sentences (Barzilay and Lapata, 2005). Coherence modeling has wide applications in natural language generation like extraction-based multi-document summarization (Barzilay and Elhadad, 2002; Galanis et al., 2012; Nallapati et al., 2017), retrieval dependent QA (Yu et al., 2018; Liu et al., 2018) and concept-to-text generation (Schwartz et al., 2017).

Earlier studies on coherence modeling and sentence ordering focused on exploiting different categories of features like coreference clues (Elsner and Charniak, 2008), entity grids (Lapata and Barzilay, 2005; Barzilay and Lapata, 2005), named-entity categories (Elsner and Charniak, 2011), and syntactic features (Louis and Nenkova, 2012) among others. With the advent of deep learning, researchers leveraged distributed sentence representations learned through recurrent neural networks (Li and Hovy, 2014). Recent works adopted ranking-based algorithms to solve the task (Chen et al., 2016; Kumar et al., 2020; Prabhumoye et al., 2020).

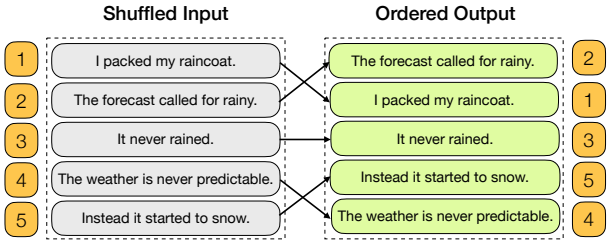

Figure 1: An example of the sentence ordering task. The goal is to reorder a set of shuffled sentences (left) into a coherent sequence of text (right).

In this paper, we present RE-BART (for **Re**order-**BART**) to solve the sentence ordering as a conditional text-to-marker generation where the input is a shuffled set of sentences and output is a sequence of position markers for the coherent sentence order.

Sentence ordering can be viewed as a task of reconstructing the correct text from a noisy input. For this reason we use BART (Lewis et al., 2020) as the underlying generation module for RE-BART. BART is pre-trained as a denoising autoencoder where one of the objective involves generating the coherent text from corrupted input sequences. Prior works encode sentences individually or in a pairwise manner and then compute the position of a sentence in the paragraph. We instead encode the entire shuffled sequence at once, which results in learning better token representations with respect to the entire input context. This helps the model in capturing interactions among all sentences and identifying the relative order among them.

[*] Authors contributed equally.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 10769–10779
November 7–11, 2021.

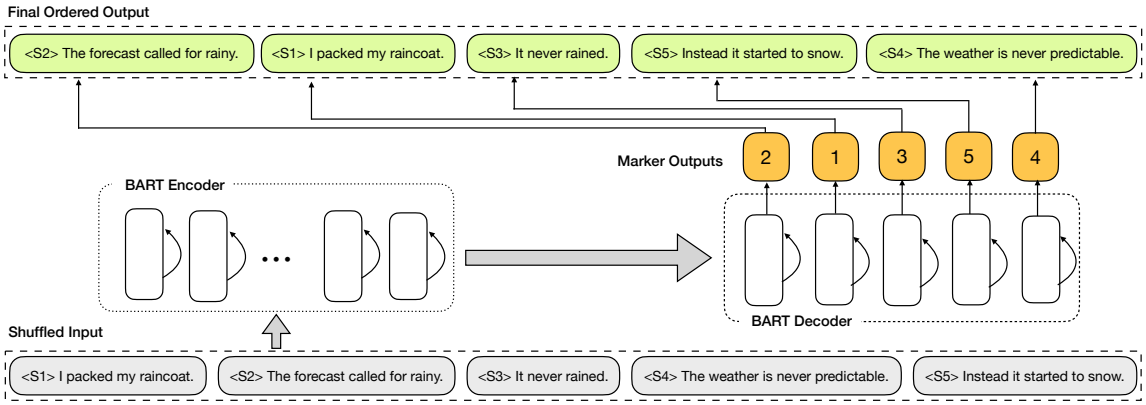

Figure 2: Proposed RE-BART framework. Given a shuffled set of input sentences, RE-BART generates Marker Outputs (position markers of sentences) which is then used to reconstruct the ordered output.

Our simple framework outperforms previous state-of-the-art by a large margin for all benchmark datasets. Specifically, we achieve 11.3%-36.2% relative improvements in Perfect Match Ratio (PMR) and 3.6%-13.4% relative improvements in Kendall's tau (τ) across all benchmarks. Our main contributions are:

- We formulate the *sentence ordering* as a conditional text generation problem and present a simple method to solve it.
- We empirically show that our model significantly outperforms existing approaches by a large margin and achieves state-of-the-art performances across all benchmark datasets.
- We conduct zero-shot evaluations showing our model trained on Movie Plots outperforms the previous in-domain trained state-of-the-art.
- We present a thorough analysis to evaluate sensitivity of our model to different input properties.

2 Related Work

The problem of sentence ordering can be formulated as finding an order with maximum coherence. Earlier works focused on modeling local coherence using linguistic features (Lapata and Barzilay, 2005; Barzilay and Lapata, 2005; Elsner and Charniak, 2011; Guinaudeau and Strube, 2013).

A line of work have leveraged neural networks to encode sentences and retrieve the final order using pointer network (Vinyals et al., 2015) by comparing them in a pairwise manner (Gong et al., 2016; Logeswaran et al., 2018a; Cui et al., 2018; Yin et al., 2019, 2020). HAN (Wang and Wan, 2019) and TGCM (Oh et al., 2019) used an attention based pointer network for decoding. B-TSort (Prabhumoye et al., 2020) uses topological sorting to retrieve the final order from sentence pairs. Zhu et al. (2021) encode sentence-level relationships as constraint graphs to enrich sentence representations. The state-of-the-art approach (Cui et al., 2020) introduced a novel pointer decoder with a deep relational module.

Other works considered reframing the task as a ranking problem. Chen et al. (2016) proposed a model which relies on a ranking framework to retrieve the order of sentence pairs. Kumar et al. (2020) utilized a BERT (Devlin et al., 2019) encoder to generate scores for each sentence which were used to sort them into the correct order.

Different from these approaches, we formulate sentence ordering as a conditional text generation task. We use a sequence-to-sequence model in our framework where the decoder encapsulates the functioning of a pointer network while generating output sentence positions. Our code is available at: https://github.com/fabrahman/ReBART.

3 RE-BART

Given a sequence of shuffled sentences $S' = \{s'_1, s'_2, \ldots, s'_{N_S}\}$, where s'_i denotes the i^{th} sentence and N_S denotes the number of input sentences, the task is to generate the ordered output sequence $S^* = \{s_1, s_2, \ldots, s_{N_S}\}$.

We solve the sentence ordering task using a text-to-marker framework shown in Figure 2. Specifically, taking a shuffled sequence of sentences (S') as input, we generate a sequence of position markers $Y = \{y_1, y_2, \ldots, y_{N_S}\}$ as output, where y_i denotes the position of i^{th} sentence of the corresponding ordered sequence (s_i) in the shuffled input. The ordered output sequence can then be reconstructed

Dataset	Split			Length		Tokens / sentence	
	Train	Dev	Test	Max	Avg	Max	Avg
NeurIPS	2.4K	0.4K	0.4K	15	6	158	24.4
AAN	8.5K	962	2.6K	20	5	543	20.7
NSF	96K	10K	21.5K	40	8.9	307	24.3
arXiv	885K	110K	110K	35	5.4	443	23.6
ROCStories	78K	9816	9816	5	5	21	9.1
SIND	78K	9.8K	9.8K	5	5	137	10.7
Movie Plots	27.9K	3.5K	3.5K	20	13.5	319	20.4

Table 1: Dataset statistics.

Dataset	epochs	learning rate	batch size
NeurIPS abstract	10	5e-6	4
AAN abstract	5	5e-6	4
NSF abstract	3	5e-6	2
arXiv abstract	3	5e-6	2
ROCStories	5	5e-6	4
SIND	5	5e-6	4
Wikipedia Movie Plots	5	5e-6	2

Table 2: Hyperparameter settings on each dataset.

as $\hat{S} = \{S'_{y_1}, S'_{y_2}, \ldots, S'_{y_{N_S}}\}$.

Our goal is to train a probabilistic model $P_\theta(\mathbf{Y}|\mathbf{S}')$ by optimizing:

$$\max_\theta \log P_\theta(\mathbf{Y}|\mathbf{S}') \quad (1)$$

The functioning of RE-BART model is shown in Figure 2. RE-BART consists of a sequence-to-sequence model with an encoder to receive a shuffled set of sentences, and a decoder to generate *position markers* (2, 1, 3 etc.), which is then used to retrieve the final ordered sequence. We use BART (Lewis et al., 2020) as the underlying sequence-to-sequence model, since our task can benefit from the *sentence permutation* pre-training objective. Additionally, to provide the model with a supervision signal to generate position markers, we append each sentence in the shuffled input with *sentence markers* (<S1>, <S2>, etc.).[1] Sentence markers were added as special tokens to the tokenizer. RE-BART learns to attend to the markers while generating the final order Y.

The proposed text-to-marker framework has two advantages over an alternate text-to-text framework, where the model directly generates the entire text sequence instead of marker outputs. First, the model performs better as the output space is much smaller. This also makes it less susceptible to neural text degeneration (Holtzman et al., 2020), as significantly fewer output tokens are generated. Second, when generating the entire text sequence in the text-to-text framework, we observe that the model often generates text which is not part of the input, rendering the output invalid for the task.

4 Experimental Setup

4.1 Datasets

We run our experiments on 7 publicly available English datasets from two domains: scientific paper abstracts and narratives.

Paper Abstracts: We evaluate our model on 4 datasets, obtained from abstracts of scholarly articles. The datasets include abstracts from NeurIPS, AAN, ACL, NSF Research Award, and arXiv (Logeswaran et al., 2018b; Gong et al., 2016; Chen et al., 2016).

Narratives: We evaluate our model on 3 datasets in the narrative domain. ROCStories (Huang et al., 2016) contains five-sentence long stories about everyday events. SIND[2] (Huang et al., 2016) is a visual storytelling dataset. Wikipedia Movie Plots [3] contains plot description of movies from Wikipedia.[4]

We randomly split ROCStories into train/test/validation in a 80:10:10 ratio. For the other datasets, we use the same train, test and validation sets as previous works. Dataset statistics are reported in Table 1.

4.2 Implementation Details

We use Huggingface library (Wolf et al., 2019) for our experiments. During inference we decode the output positions in a greedy manner by choosing the logit with the highest probability. The hyper-parameters used for each dataset are provided in Table 1 in the Appendix. The experiments are conducted in PyTorch framework using Quadro RTX 6000 GPU. The hyper-parameters for each dataset are provided in Table 2.

4.3 Evaluation Metrics

Following previous works (Cui et al., 2020; Kumar et al., 2020; Wang and Wan, 2019), we use the following metrics for evaluating our approach:

Accuracy (Acc): This is the fraction of output sentence positions predicted correctly averaged over

[1] We experimented with various combinations of sentence markers and position marker, and found out that the text-to-marker framework performs the best.

[2] http://visionandlanguage.net/VIST/dataset.html

[3] www.kaggle.com/jrobischon/wikipedia-movie-plots

[4] Movie plots contains instances with long paragraphs, we consider the first 20 sentences in every instance.

METHOD	NeurIPS abstract			AAN abstract			NSF abstract			arXiv abstract		
	Acc	PMR	τ	Acc	PMR	τ	Acc	PMR	τ	Acc	PMR	τ
Pairwise Model (Chen et al., 2016)	-	-	-	-	-	-	-	-	-	-	33.43	0.66
LSTM+PtrNet (Gong et al., 2016)	50.87	-	0.67	58.20	-	0.69	32.45	-	0.52	-	40.44	0.72
V-LSTM+PtrNet (Logeswaran et al., 2018a)	51.55	-	0.72	58.06	-	0.73	28.33	-	0.51	-	-	-
ATTOrderNet (Cui et al., 2018)	56.09	-	0.72	63.24	-	0.73	37.72	-	0.55	-	42.19	0.73
HAN (Wang and Wan, 2019)	-	-	-	-	-	-	-	-	-	-	44.55	0.75
SE-Graph (Yin et al., 2019)	57.27	-	0.75	64.64	-	0.78	-	-	-	-	44.33	0.75
FUDecoder (Yin et al., 2020)	-	-	-	-	-	-	-	-	-	-	46.58	0.77
TGCM (Oh et al., 2019)	59.43	31.44	0.75	65.16	36.69	0.75	42.67	22.35	0.55	58.31	44.28	0.75
RankTxNet (Kumar et al., 2020)	-	24.13	0.75	-	39.18	0.77	-	9.78	0.58	-	43.44	0.77
B-TSort (Prabhumoye et al., 2020)	61.48	32.59	0.81	69.22	50.76	0.83	35.21	10.44	0.66	-	-	-
BERSON (Cui et al., 2020)	73.87	48.01	0.85	78.03	59.79	0.85	50.02	23.07	0.67	**75.08**	56.06	0.83
BART (fine-tuned)	64.35	33.69	0.78	73.02	52.40	0.86	33.59	14.44	0.53	60.51	2.45	0.25
RE-BART	**77.41**	**57.03**	**0.89**	**84.28**	**73.50**	**0.91**	**50.23**	**29.74**	**0.76**	74.28	**62.40**	**0.86**

METHOD	SIND			ROCStories			Wikipedia Movie Plots		
	Acc	PMR	τ	Acc	PMR	τ	Acc	PMR	τ
Pairwise Model (Chen et al., 2016)	-	-	-	-	-	-	-	-	-
LSTM+PtrNet (Gong et al., 2016)	-	12.34	0.48	-	-	-	-	-	-
V-LSTM+PtrNet (Logeswaran et al., 2018a)	-	-	-	-	-	-	-	-	-
ATTOrderNet (Cui et al., 2018)	-	14.01	0.49	-	-	-	-	-	-
HAN (Wang and Wan, 2019)	-	15.01	0.50	-	39.62	0.73	-	-	-
SE-Graph (Yin et al., 2019)	-	16.22	0.52	-	-	-	-	-	-
FUDecoder (Yin et al., 2020)	-	17.37	0.53	-	46.00	0.77	-	-	-
TGCM (Oh et al., 2019)	38.71	15.18	0.53	-	-	-	-	-	-
RankTxNet (Kumar et al., 2020)	-	15.48	0.57	-	38.02	0.76	-	-	-
B-TSort (Prabhumoye et al., 2020),	52.23	20.32	0.60	-	-	-	-	-	-
BERSON (Cui et al., 2020)	58.91	31.69	0.65	82.86	68.23	0.88	-	-	-
BART (fine-tuned)	54.50	26.73	0.64	80.42	63.50	0.85	30.01	18.88	0.59
RE-BART	**64.99**	**43.15**	**0.72**	**90.78**	**81.88**	**0.94**	**42.04**	**25.76**	**0.77**

Table 3: Performance on abstracts (top) and narratives (bottom) datasets. The best and second-best scores are in bold and underlined. RE-BART achieves the state-of-the-art performance in PMR and τ for all datasets.

all test instances. It is defined as:

$$\text{Acc} = \mathop{\mathbb{E}}_{S' \sim \mathcal{D}} \left[\frac{1}{N_S} \sum_{i=1}^{N_S} \mathbb{I}(S'_{y_i} = s_i) \right]$$

where S' is a shuffled input from the dataset $\mathcal{D}$, s_i is the i^{th} sentence in the ordered sequence, y_i is the predicted sentence marker at position i and N_S is the number of sentences in the input.

Perfect Match Ratio (PMR): PMR measures the fraction of sentence orders exactly matching with the correct order across all input instances:

$$\text{PMR} = \frac{1}{N} \sum_{j=1}^{N} \left[\mathbb{I}(Y_j = Y_j^*) \right]$$

where Y_j and Y_j^* are the predicted and gold position marker sequences, respectively, and N is the number of instances in the dataset.

Kendall's Tau (τ): τ is a metric to evaluate the correlation between two sequences:

$$\tau = 1 - \frac{2\ (\#\text{ inversions})}{\binom{n}{2}}$$

In our setup, we evaluate τ between the predicted position marker sequence Y and gold position marker sequence Y^*. A higher score indicates a better performance for all metrics.

4.4 Baselines

We compare RE-BART with 11 previous sentence ordering frameworks including the current state-of-the-art BERSON (Cui et al., 2020). Other baselines include B-TSort (Prabhumoye et al., 2020), RankTxNet (Kumar et al., 2020), TGCM (Oh et al., 2019), FUDecoder (Yin et al., 2020), SE-Graph (Yin et al., 2019), HAN (Wang and Wan, 2019), ATTOrderNet (Cui et al., 2018), V-LSTM+PtrNet (Logeswaran et al., 2018a), LSTM+PtrNet (Gong et al., 2016) and Pairwise model (Chen et al., 2016).

Setup	BART			T5		
	Acc	PMR	τ	Acc	PMR	τ
text-to-text	80.42	63.50	0.85	62.75	34.49	0.71
text-to-marker	**90.78**	**81.88**	**0.94**	**82.36**	**64.85**	**0.88**

Table 4: Model performance using text-to-text and text-to-marker frameworks on ROCStories. A significant gain is observed using text-to-marker framework.

(a) BART embeddings (b) T5 embeddings

Figure 3: UMAP projections of pre-trained sentence representations from BART and T5 for ROCStories dataset. Embeddings of the sentences are colored based on their position in the ordered sequence S^*. It is easier to identify the gold sentence position from the pre-trained BART embeddings.

Apart from these baselines, we also include a text-to-text variant of our model, where we fine-tune a pre-trained BART model to generate the text sequences corresponding to sentences instead of their markers. We call this variant BART (fine-tuned).

5 Results

In this section, we evaluate the performance of RE-BART on several benchmark sentence ordering datasets. We also conduct a series of experiments to better understand the working of our model and investigate its generalization capability.

Table 3 reports the experimental results on all benchmark datasets.[5] RE-BART improves over all baselines by a significant margin and achieves the new state-of-the-art results in PMR and τ metrics for all datasets. In particular, RE-BART improves the previous state-of-the-art performance in PMR metric by a relative margin of 18.8% on NeurIPS, 22.9% on AAN, 28.9% on NSF, 11.3% on arXiv, 36.2% on SIND and 20% on ROCStories. We observe similar relative gains in τ: 4.7% on NeurIPS, 7.1% on AAN, 13.4% on NSF, 3.6% on arXiv, 10.8% on SIND and 6.8% on ROCStories.

We observe that RE-BART's performance on Wikipedia Movie Plots is relatively poor compared to other datasets. This could be because this dataset has relatively longer input sequences (Table 1), making the task more challenging for the model.

Comparison with text-to-text framework: Table 3 also shows that RE-BART outperforms BART (fine-tuned), our text-to-text baseline, for all datasets. BART (fine-tuned) performs reasonably well on the NeurIPS, AAN, SIND and ROCStories datasets where the average number of sentences (Table 1) is low. It struggles on NSF, arXiv and Movie Plots where input sequences are longer. Upon manual inspection, we found that BART (fine-tuned) model suffers from neural text degeneration (Holtzman et al., 2020) and produces output tokens which aren't present in the input.

We hypothesize that training in our proposed text-to-marker framework yields a performance gain over text-to-text framework, irrespective of the underlying sequence-to-sequence model. To verify this hypothesis, we compare two settings of our framework that use BART and T5 as the underlying sequence-to-sequence model. In Table 4, we observe significant gains for both BART and T5 using our text-to-marker framework. This shows the text-to-marker framework outperforms text-to-text baseline irrespective of the generation module.

From results in Table 3, we observe that our simple framework is effective and outperforms more complex baseline architectures. One explanation behind RE-BART's success could be the use of sentence markers. RE-BART is able to encapsulate the context in individual sentences (observed in generated attention maps in §6.5) and produce markers at the correct output position. Additionally, our text-to-marker framework is better at leveraging causal and temporal cues implicitly captured by BART during pre-training.

5.1 BART vs. T5

We want to study the effect of BART's pre-training objective on its performance in sentence ordering task. BART is pre-trained on multiple tasks including the rearrangement of permuted sentences, which is closely relevant to our task. To investigate if this pre-training objective provides an edge to BART, we conduct the following experiment on the ROCStories dataset. We visualize the UMAP (McInnes et al., 2018) projections of sentence representations obtained from pre-trained BART-*large* and T5-*large* models, and color code

[5]Prior results have been compiled from (Cui et al., 2020).

Method	τ	PMR
Random	0.00	20.00
RE-BART (*shuffled output*)	0.00	19.97
RE-BART (*w/o sentence markers*)	0.79	84.59
RE-BART (*random markers*)	0.92	88.97
RE-BART (*final setup*)	**0.94**	**90.78**

Table 5: Ablations with training setups on ROCStories.

them according to their position in the ordered text S^*. For example, red color represents the *first* sentence of every instance. We compare with T5, which has a similar architecture but is not pre-trained with *sentence permutation* objective.

We observe that in case of BART, sentence embeddings belonging to an identical output position (s_i) are better clustered in space, making them easier to be identified as shown in Figure 3.[6] In case of T5, the overlap among embeddings at different sentence positions is higher. To quantify the overlap we measure cluster purity following Ravfogel et al. (2020). We perform k-means clustering on UMAP projections of sentence embeddings from pre-trained BART & T5 models ($k = 5$, ROCStories has 5 sentences per instance). We measure average purity in each cluster by computing the relative proportion of the most common sentence position. The mean cluster purity for BART: 35.9% and T5: 23.6%. This indicates that since pre-trained BART is already able to segregate sentences based on their original position, it finds it easier to reorder them given a shuffled set.

The impact on downstream performance is shown in Table 4, where BART outperforms T5 in both setups. We posit that *sentence permutation* denoising during pre-training gives BART an advantage in the sentence ordering task.

5.2 Ablations

We perform a series of ablation experiments with different setups to better understand the working of our model. All the experiments in this section were performed on ROCStories dataset.

In the first ablation test, we want to verify whether the model is able to capture coherence among sentences or is just over-fitting on the data. To this end, we train our model using an arbitrarily shuffled order as output instead of the ground-truth order. We observe near random prediction performance as shown in the second row of Table 5

Next, we examine whether the sentence-markers provide any strong supervision to the model during training. Our initial assumption was that the model can use these markers adequately to learn sentence ordering. To validate our assumption, we remove the sentence-markers from the input (the input is simply a sequence of shuffled sentences) and evaluate if the model is implicitly able to figure out the sentence positions. We observe a significant drop in τ (-14.97%) and PMR (-6.19%) comparing the third and last row in Table 5 . This result shows that sentence-markers are indeed helpful.

Finally, we investigate if the sequential nature of sentence markers have an impact on the performance. We append every sentence in an input with *random sentence markers* between 0-100 (e.g. `<S47>, <S78>` etc.). We observe that the model performance is quite close to the final setup (fourth row in Table 5). There is a slight drop in performance which can be attributed to inconsistent assignment of sentence markers across different instances. This shows that the model can still effectively exploit sentence markers and their sequential nature have little impact on the final performance.

5.3 Zero-shot Performance

We investigate how well our model is able to generalize across different datasets. To this end, we evaluate the zero-shot performance of our model on different datasets.

In our experiment, we train the RE-BART model on a single dataset and test it on all others in a zero-shot setup.[7] From the results in Table 6, we observe that in most zero-shot setups RE-BART is able to perform well across different domains. Particularly, RE-BART fine-tuned on Wikipedia Movie Plots generalizes well to other unseen datasets. Surprisingly it even outperforms the previous state-of-the-art BERSON, which was fine-tuned on in-domain data, on PMR score for all datasets except NSF abstract (see the last row for comparison). We posit that the presence of longer sentences with more complex language in the Movie Plots dataset helps the model generalize to other datasets.

[6] An interesting observation from Figure 3 is that both pre-trained BART and T5 embeddings have two distinct clusters. Upon closer inspection we found out that the smaller cluster on the right in both cases correspond to sentences that appear first in the shuffled input (starting with "`<S0>`").

[7] We do not report the results of zero-shot experiments for the arXiv dataset because the training data in arXiv may overlap with NeurIPS, AAN and NSF abstract test sets.

Evaluated → / Trained ↓	NeurIPS abstract			AAN abstract			NSF abstract			SIND			ROCStories			Movie Plots		
	Acc	PMR	τ	Acc	PMR	τ	Acc	PMR	τ	Acc	PMR	τ	Acc	PMR	τ	Acc	PMR	τ
NeurIPS abstract	**77.41**	57.03	**0.89**	78.42	55.38	0.80	29.04	11.37	0.57	55.19	24.97	0.59	76.99	58.62	0.89	17.46	9.39	0.40
AAN abstract	76.99	**58.62**	0.89	**84.28**	**73.50**	**0.91**	37.09	19.12	0.62	58.08	33.93	0.64	86.62	75.46	0.91	24.08	16.23	0.48
NSF abstract	77.32	57.82	0.88	81.15	61.10	0.81	**50.23**	**29.74**	**0.76**	57.26	28.46	0.60	86.62	75.22	0.90	30.86	17.89	0.76
SIND	59.95	34.75	0.77	75.36	53.29	0.78	34.32	17.27	0.61	**64.99**	**43.15**	**0.72**	86.56	75.02	0.91	21.49	13.87	0.45
ROCStories	15.84	0.27	0.12	21.43	0.45	0.09	6.50	0.12	0.07	50.09	21.03	0.54	**90.78**	**81.87**	**0.94**	2.88	0.06	0.03
Movie Plots	73.26	54.11	0.87	78.56	64.46	0.85	39.19	20.33	0.65	58.40	36.12	0.65	87.07	76.39	0.91	**42.05**	**25.76**	**0.77**
BERSON*	73.87	48.01	0.85	78.03	59.79	0.85	50.02	23.07	0.67	58.91	31.69	0.65	82.86	68.23	0.88	-	-	-

Table 6: Performance of our model when trained on a dataset and evaluated on another in a zero-shot setup. The best and second-best performance for any metric are in bold and underline respectively. *We include the performance of BERSON for comparison purposes, when it is evaluated on the same dataset it is fine-tuned on (from Table 3).

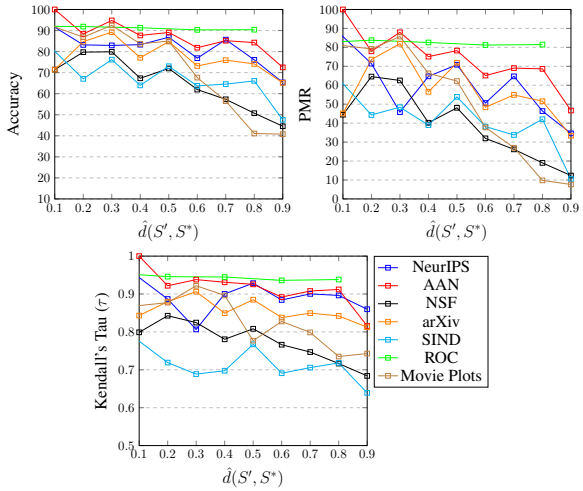

Figure 4: Variation in performance metrics with relative degree of shuffling $\hat{d}(S', S^*)$. A decline in performance is observed with a higher degree of shuffling across datasets.

RE-BART trained on ROCStories performs the worst across all datasets. Its poor performance can be attributed to the fact that ROCStories has fixed length stories with short sentences and simpler language, which makes transfer to other complex datasets harder. However, it performs reasonably well on SIND, where the data is from a similar domain and most instances are five-sentence long.

From the results in Table 6, we also observe that RE-BART performs equally well across domains (narrative → abstract, abstract → narrative). The model trained on Wikipedia Movie Plots (narrative domain), achieve the best zero-shot performance on AAN and NSF abstract (abstract domain). We also observe good performance during (narrative → abstract) transfer, when RE-BART trained on AAN and NSF is tested on ROCStories. From these experiments, we show that our model is able to generalize across domains and is not restricted to the domain of the dataset it is trained on.

6 Analysis

In this section, we perform experiments to explore RE-BART's functioning with variation in different aspects of inputs.

6.1 Effect of Shuffling

We analyze if RE-BART's performance is sensitive to the degree of shuffling in the input. To this end, we define the degree of shuffling $d(S', S^*)$ as the minimum number of swaps required to reconstruct the ordered sequence S^* from S'. Lower $d(S', S^*)$ indicates that the input S' is more similar to the ordered output sequence S^*. To effectively compare the performance across all datasets, we compute the normalized degree of shuffling as:

$$\hat{d}(S', S^*) = \frac{d(S', S^*)}{|S^*|}$$

In Figure 4, we observe a gradual decline in performance across all metrics with an increase in the normalized degree of shuffling, $\hat{d}(S', S^*)$. Overall, the results show that RE-BART performance is higher when $\hat{d}(S', S^*)$ is lower. This could be because a lower degree of shuffling means more coherent and meaningful input, resulting in an easier task for the model.

6.2 Effect of Input Length

In this experiment, we analyse how RE-BART performance varies with the number of sentences in the input. Figure 5 shows RE-BART's performance for inputs with different number of sentences, N_S. We observe a general declining trend in performance with increasing input length across different datasets.[8] This shows that the model finds it dif-

[8]We do not show the results on ROC and SIND, because these datasets mostly have a fixed number of input sentences.

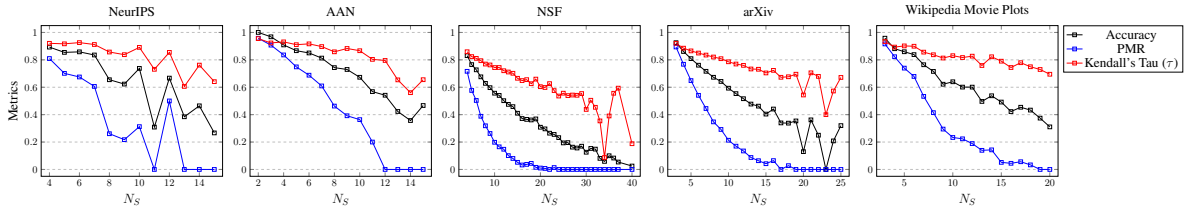

Figure 5: Variation of performance metrics (Accuracy, PMR and Kendall's Tau) with the number of input sentences N_S across different datasets. A decline in performance is observed when the number of input sentences increases.

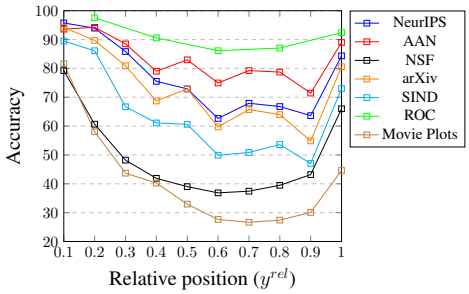

Figure 6: Position-wise accuracy. A higher prediction accuracy is observed for first and last sentences across all datasets.

Method	arXiv		SIND	
	head	tail	head	tail
Pairwise Model	84.85	62.37	-	-
LSTM+PtrNet	90.47	66.49	74.66	53.30
ATTOrderNet	91.00	68.08	76.00	54.42
SE-Graph	92.28	70.45	78.12	56.68
FUDecoder	92.76	71.49	78.08	57.32
TGCM	92.46	69.45	78.98	56.24
RankTxNet	92.97	69.13	80.32	59.68
BERSON	94.75	76.69	84.95	64.87
RE-BART	**96.46**	**80.62**	**87.97**	**73.02**

Table 7: Accuracy of predicting the first and the last sentences on arXiv and SIND datasets. RE-BART achieves the best performance for both datasets.

ficult to tackle longer input instances. The drop in performance is more pronounced for NSF and arXiv which have instances with higher number of sentences compared to other datasets. For all datasets, we observe that the rate of decline in τ is much less than Accuracy and PMR. From this observation, we infer that even if the predicted positions of individual sentences are incorrect, our model produces sentence order which are correlated with the original order.

6.3 Position-wise Performance

Here, we explore how the performance of RE-BART varies while predicting sentences at different positions in the ordered output. To uniformly investigate this across all datasets, we measure performance using a *relative output position* defined as $y^{rel} = \frac{y_i}{|S|}$. We consider y^{rel} correct to 1 decimal place and compute the prediction accuracy for each y^{rel}. The position-wise prediction accuracy for all datasets is shown in Figure 6. We observe that prediction accuracy is the highest for the first sentence, then there is a steady decline till it starts to rise again towards the end of the output sequence.

We conjecture that RE-BART is able to pick up on shallow stylistic cues, often present in the first and last sentences enabling it to have higher prediction accuracies for these positions. For example, in ROCStories all first sentences have a proper noun and introduce the protagonist of the story. In the abstracts, many papers start with similar phrases like "In this paper,", "We present " and ends with "Our contributions are ", "We achieve ", etc. For Movie plots, last sentence accuracy is significantly less than other datasets because we consider the first 20 sentences only.

Following previous works (Gong et al., 2016; Cui et al., 2018), we report the prediction accuracy of the head and tail (first and last) sentences for arXiv and SIND in Table 7. RE-BART outperforms all baselines by a large margin on both datasets.

6.4 Prediction Displacement

For instances where the model prediction was wrong $Y \neq Y^*$, we investigate how far was the model prediction Y was from the gold label Y^*. To evaluate this we compute $d(Y, Y^*)$, the minimum number of swaps required to retrieve Y^* from Y. We experiment on Wikipedia Movie Plots dataset where the performance of RE-BART was not as good as other datasets. From Figure 8, we observe that most of the incorrectly predicted samples had a low $d(Y, Y^*)$, with 70% of the incorrect predictions having $d(Y, Y^*) \leq 6$. This shows that even

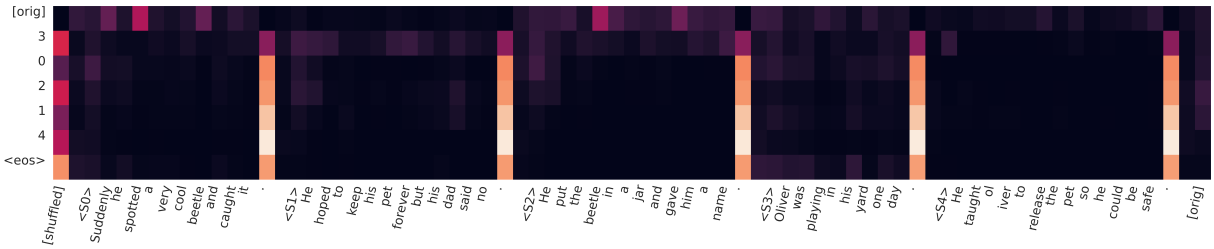

Figure 7: Visualization of cross-attention in the trained RE-BART model for an input instance from ROCStories. The y-axis shows output tokens, x-axis shows input tokens, and colorized cells denote the cross-attention between tokens at a position (x, y). Lighter color indicates higher attention values. The model learns to attend around sentence markers and other special tokens.

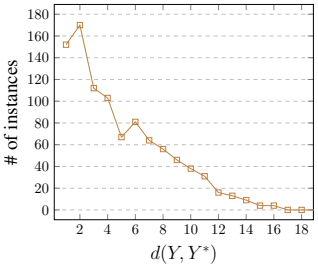

Figure 8: Plot shows how many instances were incorrectly predicted for each $d(Y, Y^*)$ for Movie Plots.

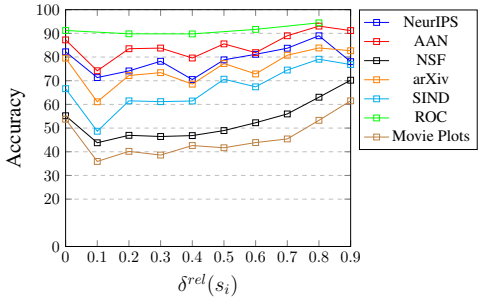

Figure 9: Accuracy at a position based on how far it is from the original position. Accuracy doesn't change much with $\delta^{rel}(s_i)$.

if the model make a wrong prediction, it mostly misses a few positions and does not get the entire order wrong.

6.5 Attention Visualization

We visualize the norm-based cross-attention map (Kobayashi et al., 2020), between the decoded output and encoder input, of one of the attention head in Figure 7. Lighter color indicates higher attention values. We append all input instances with special tokens `[shuffled]` and `[orig]` at the beginning and end respectively, along with sentence markers at the start of each sentence. In Figure 7, we observe that the model attends to tokens near those special tokens. This shows that during decoding the model finds only tokens next to the sentence markers useful. We hypothesize this is due to the fact that these tokens are able to encapsulate the context of the corresponding sentence. We observe similar maps across different attention heads.

6.6 Effect of Sentence Displacement

We investigate if there is any variation in performance if a sentence is placed too far from its position in the shuffled sentence. We compute relative distance from the original position $\delta^{rel}(s_i)$ as:

$$\delta^{rel}(s_i) = \frac{|i-j|}{|S^*|} \text{ s.t. } s_i = s'_j$$

Figure 9 shows how the performance varies with respect to $\delta^{rel}(s_i)$. We observe that accuracy doesn't change much with relative displacement. We infer that local sentence-level relative displacement doesn't dictate the performance as much as global input-level factors like degree of shuffling and input length.

7 Conclusion

In this work, we address the task of sentence ordering by formulating it as a conditional text generation problem. We observe that simply generating output text from shuffled input sequences is difficult due to neural text degeneration. We solve this problem by proposing RE-BART, a text-to-marker generation framework. RE-BART achieves the state-of-the-art performance on 7 benchmark datasets and is able to generalize well across different domains in a zero-shot setup. We investigated the limitations of our model, and found that RE-BART is sensitive to various factors like number of input sentences and degree of shuffling. Future works can focus on developing models which are robust to such factors.

References

Regina Barzilay and Noemie Elhadad. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

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

Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2016. Neural sentence ordering. *arXiv preprint arXiv:1607.06952*.

Baiyun Cui, Yingming Li, Ming Chen, and Zhongfei Zhang. 2018. Deep attentive sentence ordering network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4340–4349, Brussels, Belgium. Association for Computational Linguistics.

Baiyun Cui, Yingming Li, and Zhongfei Zhang. 2020. BERT-enhanced relational sentence ordering network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6310–6320, Online. Association for Computational Linguistics.

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

Micha Elsner and Eugene Charniak. 2008. Coreference-inspired coherence modeling. In *Proceedings of ACL-08: HLT, Short Papers*, pages 41–44, Columbus, Ohio. Association for Computational Linguistics.

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*, pages 125–129.

Dimitrios Galanis, Gerasimos Lampouras, and Ion Androutsopoulos. 2012. Extractive multi-document summarization with integer linear programming and support vector regression. In *Proceedings of COLING 2012*, pages 911–926, Mumbai, India. The COLING 2012 Organizing Committee.

Jingjing Gong, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2016. End-to-end neural sentence ordering using pointer network. *arXiv preprint arXiv:1611.04953*.

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. 2016. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1233–1239, San Diego, California. Association for Computational Linguistics.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Pawan Kumar, Dhanajit Brahma, Harish Karnick, and Piyush Rai. 2020. Deep attentive ranking networks for learning to order sentences. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8115–8122. AAAI Press.

Mirella Lapata and Regina Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pages 1085–1090. Professional Book Center.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiwei Li and Eduard Hovy. 2014. A model of coherence based on distributed sentence representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2039–2048, Doha, Qatar. Association for Computational Linguistics.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2018. Stochastic answer networks for machine reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1694–1704, Melbourne, Australia. Association for Computational Linguistics.

Lajanugen Logeswaran, Honglak Lee, and Dragomir R. Radev. 2018a. Sentence ordering and coherence modeling using recurrent neural networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5285–5292. AAAI Press.

Lajanugen Logeswaran, Honglak Lee, and Dragomir R. Radev. 2018b. Sentence ordering and coherence modeling using recurrent neural networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5285–5292. AAAI Press.

Annie Louis and Ani Nenkova. 2012. A coherence model based on syntactic patterns. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1157–1168, Jeju Island, Korea. Association for Computational Linguistics.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3075–3081. AAAI Press.

Byungkook Oh, Seungmin Seo, Cheolheon Shin, Eunju Jo, and Kyong-Ho Lee. 2019. Topic-guided coherence modeling for sentence ordering by preserving global and local information. 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 2273–2283, Hong Kong, China. Association for Computational Linguistics.

Shrimai Prabhumoye, Ruslan Salakhutdinov, and Alan W Black. 2020. Topological sort for sentence ordering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2783–2792, Online. Association for Computational Linguistics.

Shauli Ravfogel, Yanai Elazar, Jacob Goldberger, and Yoav Goldberg. 2020. Unsupervised distillation of syntactic information from contextualized word representations. *arXiv preprint arXiv:2010.05265*.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. 2017. The effect of different writing tasks on linguistic style: A case study of the ROC story cloze task. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 15–25, Vancouver, Canada. Association for Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2692–2700.

Tianming Wang and Xiaojun Wan. 2019. Hierarchical attention networks for sentence ordering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 01, pages 7184–7191. AAAI Press.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv*.

Yongjing Yin, Fandong Meng, Jinsong Su, Yubin Ge, Lingeng Song, Jie Zhou, and Jiebo Luo. 2020. Enhancing pointer network for sentence ordering with pairwise ordering predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9482–9489. AAAI Press.

Yongjing Yin, Linfeng Song, Jinsong Su, Jiali Zeng, Chulun Zhou, and Jiebo Luo. 2019. Graph-based neural sentence ordering. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5387–5393. ijcai.org.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Yutao Zhu, Kun Zhou, Jian-Yun Nie, Shengchao Liu, and Zhicheng Dou. 2021. Neural sentence ordering based on constraint graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14656–14664.