

NAREOR: The Narrative Reordering Problem

Varun Gangal^{* 1} Steven Y. Feng^{* 1} Malihe Alikhani² Teruko Mitamura¹ Eduard Hovy¹

¹ Language Technologies Institute, Carnegie Mellon University
{vgangal, syfeng, teruko, hovy}@cs.cmu.edu

² School of Computing and Information, University of Pittsburgh
malihe@pitt.edu

Abstract

Many implicit inferences exist in text depending on how it is structured that can critically impact the text’s interpretation and meaning. One such structural aspect present in text with chronology is the order of its presentation. For narratives or stories, this is known as the *narrative order*. Reordering a narrative can impact the temporal, causal, event-based, and other inferences readers draw from it, which in turn can have strong effects both on its interpretation and interestingness. In this paper, we propose and investigate the task of *Narrative Reordering* (NAREOR) which involves rewriting a given story in a different narrative order while preserving its plot. We present a dataset, NAREORC, with human rewritings of stories within ROCStories in non-linear orders, and conduct a detailed analysis of it. Further, we propose novel task-specific training methods with suitable evaluation metrics. We perform experiments on NAREORC using state-of-the-art models such as BART and T5 and conduct extensive automatic and human evaluations. We demonstrate that although our models can perform decently, NAREOR is a challenging task with potential for further exploration. We also investigate two applications of NAREOR: generation of more interesting variations of stories and serving as adversarial sets for temporal/event-related tasks, besides discussing other prospective ones, such as for pedagogical setups related to language skills like essay writing and applications to medicine involving clinical narratives.

1 Introduction

From the onset of language, storytelling has been crucial to the transmission of knowledge (Ramanujan 1991). It has been well-established that readers remember only an abstract representation of stories (Schank 1972). Before the printing press, classes engaged with oral teaching of scriptures, such as rabbis, underwent extensive training to reproduce them with no distortion (Bos 1995). Formally analyzing story structure commenced with the ancients, through works like Aristotle’s *Poetics* (Halliwell et al. 1998). These studies led to the concept of a *narrative*, distinct from story events.

For a story, there are two *orders*: the chronological order of events as they happened and their order as presented in text. These have been analyzed under different names (Propp 2010). We refer to them as *story order* and *narrative order*,

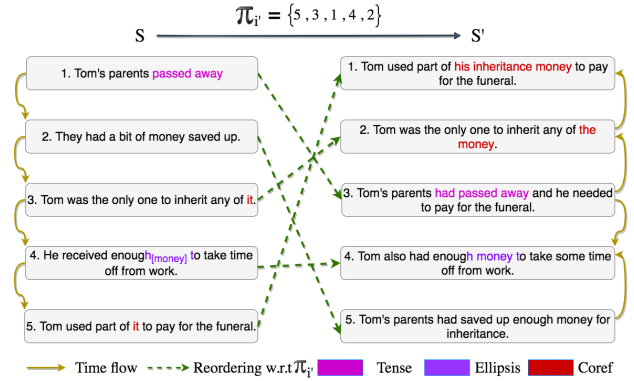


Figure 1: Example of our task and dataset, with original input story S on the left, target narrative order π_i on the top, and human rewritten story S' on the right.

or *story* and *narrative*, respectively. Genette (1983) enlists typical orders observed in writing. A *linear* order narrates events in same sequence as story order. The *in medias res* order starts with events in the middle, goes back to the start, then proceeds to the end. Changing from near-*linear* to more “interesting” orders is prevalent in cinema, e.g. *The Imitation Game* starts with Turing’s post-WWII 1951 interrogation. *Memento* and *Naked Lunch* are known for their esoteric narrative orders - loosely described as retrograde (reverse of linear) and syllepsis (lacking chronological logic), respectively.

Morgan (2017) explains how narratives surpass “mere chronicle”. Narrative orders of presenting materials in scientific *explanations* directly affects how researchers *interpret* and *understand* them since the order implies not only temporal but other inferences about causality, processes of change, etc. Narrative order can thus influence *model explainability*, especially for explanation generation (Rajani et al. 2019), a recent area-of-interest (Wiegrefe and Marasovic 2021).

In this work, we do not delve into the complex and somewhat subjective question of *which* narrative order is most suitable or “interesting”. We focus on *how* a given story in *linear* narrative order can be rendered in a specified, *non-linear*, *target* order while preserving plot. We call this *Narrative Reordering*, or NAREOR. To the best of our knowledge, we are the first to propose and investigate this task.

^{*} Equal Contribution by the two authors

Our work is not entirely adrift from past research in this vein. Montfort (2007) tries generating fiction narratives from basic existent-event info with a special focus on narrative order, using a rule and planning approach. Unlike our work, their rule-based system does not involve learning. Moreover, being generation in a given narrative order from unstructured story elements rather than reordering an existing story, their setting does not require solving challenges such as disentangling events from stories which are inherent in NAREOR.

Formally, NAREOR involves reordering a story S with sentences s_1, s_2, \dots, s_n to a reordered story S' with sentences s'_1, s'_2, \dots, s'_n according to a given target narrative order $\pi_{i'}$. $\pi_{i'}$ is a permutation $\{\pi_{i'} | \pi_{i'} : i' \rightarrow f(i'); 1 \leq i' \leq n; f(i') = i\}$ mapping from target sentence¹ indices i' to original sentence indices i , where f is a one-to-one and onto function from $\{1, 2 \dots n\}$ to itself. In practice, we write $\pi_{i'}$ as the sequence $\{i = f(i')\}_{i'=1}^n$ (f and i' become implied).

NAREOR’s challenges are evident from the example in Figure 1. Simply reordering sentences is far from sufficient, as rewritten text must be adjusted to handle coreference, tense, and other discourse dependencies. For example, narrative order affects tense since it can change the first 2 of 3 Reichenbach times (Reichenbach 1947) that together determine tense - speech, reference, and event time. NAREOR involves pinpointed and critical edits; a single missed or incorrect edit can result in an entirely different or invalid plot. Since $\pi_{i'}$ can be seen as a *control*, NAREOR is a controllable generation task (see Appendix §A for discussion).

NAREOR is also a novel form of story-level paraphrasing and can be used to generate more interesting variations of stories (§5.1). Outputs can also serve as challenge sets for temporal or event-based tasks such as sentence ordering to assess the temporal reasoning capabilities of models (§6). NAREOR can also be potentially useful for pedagogical setups related to language skills such as essay writing, and applications to medicine involving clinical narratives (§6).

To complement NAREOR, we present a dataset, NARE-ORC, with human rewritings of stories from ROCStories (Mostafazadeh et al. 2016a) in non-linear orders. We conduct a thorough analysis, examining various ways humans modify the text when reordering (§2). We perform experiments with BART, T5, and GPT-2 on NAREORC using novel, task-motivated training methods we propose (§3). We evaluate our models with both an automatic and human evaluation along with qualitative analysis (§5). We demonstrate that our proposed training methods are effective but have room for further improvement. We illustrate that NAREOR is indeed a challenging task with potential for further exploration.²

2 Dataset: NAREORC

2.1 Dataset Construction

Source Corpus: ROCStories has $\approx 98.5K$ five-sentence English stories. For the dev and test splits, each example contains a four-sentence story prefix with a one-sentence co-

herent and incoherent ending. We treat the coherent endings as the fifth sentences for NAREORC’s dev and test stories.

Assigning Target Narrative Orders: The target narrative order $\pi_{i'}$ is not part of the ROCStories input. We devise a randomized procedure to assign a reasonable $\pi_{i'}$ for each example. We sample 3 permutations from the set of non-identity $n!-1$ permutations.³ We find Kendall τ correlations (Kendall 1938) between identity permutation $I_n, \{1,2,3,4,5\}$, and each of the three permutations, retaining the lowest as $\pi_{i'}$. We prefer this to sampling at random because we want our examples to be sufficiently non-trivial w.r.t. the task.

Supervised & Unsupervised Splits: We set aside 600, 200, 200 stories from train, dev, and test splits of ROCStories. These act as NAREORC’s trainSup, devSup, and testSup splits, for which we collect human references. Remaining stories in each ROCStories split are retained as trainUnsup, devUnsup, and testUnsup of size 95161, 1671, 1671.

Human Annotation: For trainSup and devSup, we annotate one reference per example. For testSup, we collect two each to help reference-based metrics. We conduct our study on AMT. To understand task difficulty, we ask a “Hardness” question with options *VeryEasy, Easy, Moderate, Hard, VeryHard*. On average, annotators found $\approx 70\%$ of rewritings to be *Moderate* or *Hard*, demonstrating that NAREOR is quite difficult even for humans. More details in Appendix §B.

2.2 Dataset Analysis

Overall Statistics

We find human-rewritten stories S' are $\approx 1.2x$ as long as input stories S on avg in words and characters. We expect this given the narrative reordered story favors resolution of sentence-order dependent elements like ellipses (s_4 and s'_4 in Figure 1) and pronouns (s_3 and s'_2 in Figure 1) to explicit forms. It also requires insertion of time expressions (e.g. *Before that*, 3rd row, Table 1) to clarify the now disrupted flow.

Unique n -gram ratio $UR_n(S)$ is the fraction of unique n -grams of length n in S . We observe all three mean URs ($n = 1, 2, 3$) to decrease from input to reference story. UR_1 : $0.692 \rightarrow 0.669$, UR_2 : $0.940 \rightarrow 0.931$, UR_3 : $0.989 \rightarrow 0.984$. Increased n -gram repetition could have reasons similar to length increase, causing cross-sentence repetition. Figure 1 demonstrates this: S only has one instance of *money*. Conversion of *inherit any of it* (s_3) \rightarrow *inherit any of the money* (s'_2) and *enough to take time* (s_4) \rightarrow *enough money to take some time* (s'_4), among other changes, results in four in S' .

How Verb Forms Change

We note changes in occurrence distribution across verb-related *pos* tags from S to S' using NLTK’s *pos* tagger. Gerund fraction (*pos*=VBG) (e.g. *I like playing*) increases $7.7\% \rightarrow 9.5\%$. Past participle fraction (*pos*=VBN) (e.g. *He had broken it*) \approx doubles, $6.5\% \rightarrow 12.4\%$. Past tense fraction (*pos*=VBD) (e.g. *He broke it*) decreases $60.9\% \rightarrow 54.6\%$. Other verb-related *pos* fractions remain fairly constant. Increase in past participle can be explained by frequent con-

¹For simplicity, we assume *narrative* to break up into sentence units. Our task is still very challenging as shown through this paper.

²Code + data will be released shortly.

³In our case, $n = 5$ as we experiment with ROCStories.

version to past perfect tense during reordering (e.g. *parents passed away*→*parents had passed away* in Figure 1).

How Narrative Reordering Alters Sentences

We look at corresponding sentence pairs $\{s_i, s'_{i'}\}$ in each story, specifically 4 linguistic change types - ellipsis, tense, time expressions (timexes), coreference. We tried detecting these using off-the-shelf tools, and did not find any for ellipsis. Timex detectors like SUTime (Chang and Manning 2012) only mark strict timexes (e.g. *last Sunday*) but not others (e.g. *before midsems*). We hence hand-annotate these four for each $\{s_i, s'_{i'}\}$ per testSup example. These are further described in Table 1. We find over half (51.5%) the examples show ≥ 3 of 4 change types at once, and 89.5% show ≥ 2 . This shows that NAREOR requires performing different changes in tandem.

3 Methodology

3.1 Training Methods

We introduce two task-specific training methods.

NAR-denoise (NAR-d)

This is partially inspired by how humans rewrite; a common approach is to first reorder sentences naively (simply swap positions), then make other changes. NAR-d attempts to mimic this, learning to convert from naive orderings to high-quality text. It involves two stages of model training.

1. **Denoise-1S:** Stage 1 is unsupervised training through story-level denoising. We use trainUnsup without human-written reorderings, and simulate them using the original human-written ROCStories (the outputs during training). Deletion and swapping of tokens are used to create inputs from these stories that simulate naive reorderings. This noising aims to emulate the reverse of the content editing that occurs during NAREOR. Specifically, we randomly delete 12.5% of tokens and swap another 12.5%. We found human-rewritten stories were, on average, in combination of token length (longer) and swappings, $\approx 25\%$ different from the originals. We split this between deletion and swapping to approximate naively-reordered stories. Story sentences S are first reordered as per $\pi_{i'}$ to produce S'_{naive} , then each is edited to fit the new narrative. We swap tokens as humans often swap words like coreferent mentions based on how the narrative order changes. Hence, this stage learns to denoise text by converting noised versions to human-written text.

2. **Denoise-2S:** The second stage is supervised training atop the model above. The inputs are the 600 original stories in trainSup, with sentences naively reordered as per target narrative order $\pi_{i'}$ to S'_{naive} , and the outputs are the human rewrites of these. The model learns to further translate from naively-reordered text to fluent human-written text.

NAR-reorder (NAR-r)

Unlike NAR-d, NAR-r models themselves handle reordering given the target order rather than naive reordering beforehand.

• **Input Encoding Scheme:** We describe how the task input $\{S, \pi_{i'}\}$ is encoded as a token sequence for both Stage-1 and 2 training. To enable the model to distinguish different sentences, we prefix each $s \in S$ with a tag from $\langle a \rangle$ to $\langle e \rangle$. We specify $\pi_{i'}$ as a sequence of these, separated from

S by $\langle sep \rangle$. NAREOR involves rearranging mention types among coreference chains (see §2.2), so we use NeuralCoref (HuggingFace 2020) to detect these chains. For each, we assign a unique uppercase tag ($\langle X \rangle$) to replace its mentions. At the end of the input, we list each tag and the head mention of its coreference chain in order. We then append $\langle st \rangle$ to mark the end of the input. An illustration of the scheme follows: $\langle a \rangle$ *Since I had front seat tickets, I was able to directly see* $\langle X1 \rangle$. $\langle b \rangle$ $\langle X1 \rangle$ *tried to reach out with* $\langle X1 \rangle$ $\langle X2 \rangle$. $\langle c \rangle$ *I grabbed* $\langle X2 \rangle$ *and* $\langle X1 \rangle$ *pulled me on stage.* $\langle d \rangle$ $\langle X1 \rangle$ *began to sing.* $\langle e \rangle$ *The concert had started.* $\langle sep \rangle$ $\langle e \rangle$ $\langle d \rangle$ $\langle a \rangle$ $\langle b \rangle$ $\langle c \rangle$ $\langle X1 \rangle$ *The music artist* $\langle X2 \rangle$ *her hand* $\langle st \rangle$

• **Reorder-1S:** We use examples from trainUnsup for stage 1. It is problematic to train for the forward direction of our task $S, \pi_{i'} \rightarrow S'$ since S' is not known. Approximating S' using S'_{naive} would hurt output fluency. We instead train in the inverse direction $S'_{naive}, \pi_{i'}^{-1} \rightarrow S$, where $\pi_{i'}^{-1}; \pi_{i'}^{-1}(\pi_{i'}) = I_n$ is the inverse permutation of $\pi_{i'}$. To reduce train-test mismatch, we use the inverse formulation half the time, and an autoencoding one, i.e. $S, I_n \rightarrow S$ the other half.

• **Reorder-2S:** trainSup examples are used to further finetune on reorder-1S. We train in the task direction $S, \pi_{i'} \rightarrow S'$.

3.2 Chosen Models

We choose several pretrained generation models: GPT-2, BART, and T5. We finetune all using both our training methods to produce denoise-1S (d-1S), denoise-2S (d-2S), reorder-1S (r-1S), and reorder-2S (r-2S) versions. GPT-2 (Radford et al. 2019) is a Transformer-based language model trained on *WebText*. BART (Lewis et al. 2020) and T5 (Raffel et al. 2020) are Transformer seq2seq models. BART is trained as a denoising autoencoder to reconstruct original from noised text. T5 is designed to be effective for transfer learning. We use HuggingFace’s implementations of their base versions.⁴

3.3 Automatic Evaluation Metrics

Reference-Based Metrics assess the similarity between generated text and human-written references. We use **BLEU** (Papineni et al. 2002), **METEOR** (Banerjee and Lavie 2005), and **BERTScore** (Zhang et al. 2019). We compare generated text with the two references per testSup example.⁵

Target Order Fidelity (TOF) is defined as how closely the reordered text matches the given target narrative order. E.g. given $S = \{s_1, s_2, s_3\}$, $\pi_{i'} = \{3, 2, 1\}$, and $S' = \{s'_1, s'_2, s'_3\}$, we wish to see if s_1 has correctly been translated to s'_3 . We introduce **TOF-METEOR** and **TOF-BERTScore**. These assess the average METEOR and BERTScore values for each aligned pair $\{s_i, s'_{i'}\} \forall i$ (where i' refers to the target index for s_i). Higher values correspond to more content preservation, where each output sentence is more likely in the correct position. Some drop is expected in modulating for $\pi_{i'}$, but the overall content should be faithful. These metrics serve more as validation, where reasonable values (e.g. > 50)⁶ are sufficient. Lower values indicate more changing of the text which may be necessary for certain narrative reorderings.

⁴See §4 for further training/finetuning details.

⁵Correlates well with human evaluation as shown in §5.

⁶Assuming the values are multiplied by 100.

Change Type	Story Examples with Changes Highlighted
Ellipsis (Sent: 5.7%) (Stor: 27.5%)	S : 1. All of the Ross family has red hair, except Henry. 2. Henry has blonde hair that is very curly. 3. Henry’s father often teases Henry’s mother about the mailman. 4. The mailman has blonde, curly hair, but he is very ugly. 5. <i>His dad’s teasing makes Henry feel bad.</i> ; $\pi_{i'}$: {1, 5, 4, 2, 3} S' : 1. All of the Ross family has red hair, except Henry. 2. <i>His dad’s teasing about the mailman makes Henry feel very bad.</i> 3. This is because the mailman has blonde, curly hair, but he is very ugly. 4. Henry also has blonde hair that is very curly. 5. Henry’s father often teases Henry’s mother about the mailman.
Tense (Sent: 19.1%) (Stor: 64.0%)	S : 1. Sam bought a new SUV. 2. It was all wheel drive. 3. He figured he would take it off road. 4. <i>He hit a few hard bumps and broke his suspension.</i> 5. Sheepishly, he brought it to the dealership for repair. ; $\pi_{i'}$: {2, 3, 5, 1, 4} S' : 1. Sam’s SUV was an all wheel drive. 2. He thought he could take it for a spin off road. 3. Embarrassed by the outcome of his drive, Sam took the car to the dealership for repair. 4. He had just bought the SUV. 5. <i>The car had hit a few hard bumps and the suspension broke when Sam took it off road.</i>
Timexes (Sent: 34.0%) (Stor: 85.5%)	S : 1. There was once a kitten that did not have a home. 2. <i>The poor kitten walked around cold and hungry.</i> 3. One day, a nice lady let the kitten into her home. 4. The woman gave the kitten food and a bed. 5. The kitten was happy to be adopted. ; $\pi_{i'}$: {4, 2, 5, 1, 3} S' : 1. A woman gave a home to a cat. 2. <i>Before that it was cold and hungry.</i> 3. It made the cat happy to have a home. 4. The little cat originally was homeless. 5. But in the end, it met the nice woman and she let it in.
Coreference (Sent: 20.7%) (Stor: 71.5%)	S : 1. Jimmy wandered around the city looking for a place for a soda. 2. Before he knew it, he was in an unfamiliar area. 3. He was scared of strangers and didn’t want to ask anyone. 4. Soon a policeman came by and asked if he was lost. 5. <i>He told him that he was lost.</i> ; $\pi_{i'}$: {5, 4, 2, 1, 3} S' : 1. <i>Jimmy</i> told a police officer that he was lost. 2. He was lucky the police showed up in the first place. 3. He had no idea where he was. 4. He had wandered off when trying to find somewhere to buy a soda. 5. It was pretty terrifying being all alone in a mysterious area with strangers.

Table 1: Sentence pairs in testSup stories are annotated for 4 linguistic change types common in NAREORC. Sent denotes % of sentence pairs showing that change type. Stor denotes story pairs (S, S') where \geq one sentence pair shows that change type.

4 Experiments

Model Finetuning and Generation

For finetuning our models, we try different combinations of learning rates (LR) for both stages. We look at either the loss (for BART and T5) or perplexity (for GPT-2) on the respective validation splits (devUnsup for 1st stage and devSup for 2nd), and choose the epoch with the lowest.

We evaluate each model on testSup, where we can directly compare results to NAREORC’s human rewritings. We generate a single output per test example. The inputs are the original examples to NAR-r models and the S'_{naive} of the examples to NAR-d models. See §3.1 for more details.

We only keep the first five sentences of each output. For BART and T5, we use beam search with a width of 5.⁷ For GPT-2, we use a nucleus sampling budget (Holtzman et al. 2019) of 0.9 and output length limit of 500. We try various softmax temperatures and find 0.9 performs best. For GPT-2, during finetuning, it is given the concatenation of the input plus output. During generation, it is only fed the input for which it generates a continuation (the output). We noticed that many GPT-2 generations included trailing exclamation marks, and strip these if more than four occur in a row.⁸

Human Evaluation

Annotators evaluate 100 testSup examples each from the original stories, human rewritings, outputs from our two-stage models, and a subset of one-stage models. Each example is evaluated by two annotators. See Appendix §D for more.

They evaluate fluency, coherence, logic, and plot preservation (plot-pres) on 1-5 scales. Fluency is a measure of how

fluent and readable a text is. Coherence is how well individual sentences fit together (Barzilay and Lapata 2008). Logic is the plausibility of described events. Plot-pres is how well reordered text preserves the plot of the original. This includes details about characters, events, and interactions between them, encompassing its semantic and temporal aspects.

We also conduct an interestingness (interest) study on human rewritings and outputs from our BART-2S and T5-2S models. Each reordered story’s interestingness w.r.t. suspense and time flow compared to the original are evaluated from 1-5 by two annotators. We ask the following: “On a scale of 1-5, with 1 being most decrease in interestingness and 3 being same level of interestingness and 5 being most increase in interestingness, how interesting is the suspense and flow of time in the story S , compared to the original story O ? How exciting did you find the story as you read through it?”

5 Results and Analysis

We present evaluation results of our 2S and subset of 1S models on testSup compared to human rewritings and original stories. Tables 2 and 3 contain human evaluation results, and Table 4 automatic evaluation results. Correlations between automatic and human metrics are in Table 5. Table 6 contains qualitative examples, with more in Appendix §E.

5.1 Analysis of Human Evaluation Results

We begin by analyzing human evaluation performance through results in Tables 2 and 3.

Fluency, Coherence, Logic: Original stories are the highest for all three metrics⁹ with human rewritings second for co-

⁷Nucleus sampling did not work as well for BART and T5.

⁸See Appendix §C for more finetuning/generation details.

⁹Although these metrics slightly decrease for reordered stories, we note that NAREOR’s main purpose is for more interesting

Method \ Metric	Fluency	Coherence	Logic	Plot-pres
Original stories	4.209	4.0	3.851	N/A
Human rewritings	3.797	3.723	3.784	3.972
GPT2-d-2S	3.635	3.399	3.399	3.708
GPT2-r-2S	3.595	3.378	3.291	3.375
BART-d-1S	3.628	3.412	3.318	3.847
BART-d-2S	3.818	3.507	3.493	3.722
BART-r-2S	3.757	3.439	3.493	<u>3.861</u>
T5-d-2S	3.764	3.419	<u>3.5</u>	3.889
T5-r-1S	3.655	3.378	3.486	3.847
T5-r-2S	<u>3.784</u>	3.595	3.520	<u>3.861</u>

Table 2: Average human evaluation results on testSup (excl. interestingness), rated from 1-5. Bold corresponds to best model performance per metric, and underline second-best model performance.

Method:	Human	BART-d	BART-r	T5-d	T5-r
Interest	3.75	3.367	3.483	<u>3.533</u>	3.3

Table 3: Average interestingness results on testSup, rated from 1-5 (3 represents equal to original story). Models are 2S versions. Bold corresponds to best performance, and underline second-best.

herence and logic, beating the models by a noticeable degree. BART-d-2S and T5-r-2S are generally the best-performing models here. BART-d-2S slightly outperforms human rewritings on fluency, with T5-r-2S closely behind, demonstrating that these models are quite fluent. These models also outdo their 1S variants. GPT-2 models perform worst on all metrics.

Plot-pres: We see that human rewritings best preserve the plot of the original stories. T5-d-2S is the best performing model on plot-pres, followed by BART-r-2S and T5-r-2S. GPT-2 models perform the worst at preserving the plot of the original stories (which we show qualitatively in §5.3).

Interestingness: Human rewritings score highest on interest. Humans rewrite the text in more creative ways, whereas BART and T5 models are more conservative (see §5.2 TOF and §5.3). Narrative reorderings for all methods are more interesting, on average, than original stories. NAREOR can indeed be used to generate more interesting story variations.

5.2 Analysis of Automatic Evaluation Results

We now analyze the automatic evaluation performance of the different methods in Table 4.

BERTScore, BLEU, METEOR: We see from Table 5 that these reference-based metrics correlate quite well with human eval metrics, particularly plot-pres. T5-d-2S performs best followed by BART-d-2S. Similar to the human evaluation, 2S models outperform their 1S variants, and GPT-2 models perform worst overall. Denoise outperforms reorder variants and generate more similar text, on avg, to human references.

Target Order Fidelity (TOF): It appears all approaches are reasonable (e.g. > 50 for TOF metrics), and outputs are likely in the correct target orders. Human rewritings have the lowest TOF; humans are less conservative while rewriting (shown in §5.3). GPT-2 models modify text second heaviest,

tellings of the same story which we do achieve (see Table 3).

but perform worst overall. They introduce more errors, e.g. repeating or hallucinating to degrade text quality and plot-pres (§5.3). BART and T5 models are more conservative. It appears they have learned to perform minimal but effective edits (§5.3). They lag behind humans and heavier editing may be required to further improve. Lastly, it appears the reorder models modify text more heavily than their denoise variants.

5.3 Qualitative Analysis

From Table 6, we see that humans modify text heavily to suit the reorderings and are sometimes quite creative, e.g. phrasing Fred as having *grown accustomed to the bird being his alarm clock* (ex. 2). Humans successfully handle necessary coreferences, tenses, time expressions (timexes), etc.

GPT-2 modifies text quite heavily but suffers from incorrect coreference while introducing spurious tokens, repetition, or hallucinations. For ex. 2, GPT2-r changes the plot greatly, stating *Fred woke him up for work* and *This was because he liked Fred* (likely due to poor coreference), and hallucinating *This bird, however, did not like Fred*. For ex. 4, it repeats Joey’s excitement many times, while hallucinating a *roller coaster* that was absent in the original story.

BART and T5 models are more conservative, but their edits are important and effective. They handle coreference, tense, and timexes quite well. These pinpointed and critical edits are required to maintain plot. For ex. 1, they modify *He told him that he was lost* to *Jimmy told a/the policeman that he was lost* given that sentence is now at the beginning. BART-d impressively modifies tense by converting *Soon a policeman came by and asked if he was lost* to *The policeman had come by and asked if he had been lost*. For ex. 2, T5-d converts *enjoyed* to *had enjoyed* since the bird no longer singing is now prior information, and adds the timex *After a while* to the beginning of the last output sentence. BART-r successfully changes *Fred began to like the bird* to *He had begun to like the bird*. For ex. 3, BART-d inserts the timex *Earlier* at the beginning of the second output sentence, correctly and unambiguously conveying its underlying temporality w.r.t. the first. BART-d correctly changes *saw a turtle* to *had seen a turtle*, while BART-r does so for *stepped to had stepped*. For ex. 4, BART and T5 models all resolve the *Disneyland* ellipsis by converting *Joey had a great time* to *Joey had a great time at Disneyland*, while GPT2-d cannot.

However, the BART and T5 models are imperfect. For ex. 1, BART-r hallucinates *lost his wallet* (original story does not involve a wallet), T5-d inserts an incorrect timex of *Soon after* at the beginning of the second output sentence, and T5-r hallucinates *asked if he had a soda* (this is not asked in the original story). For ex. 2, BART-r incorrectly converts *the bird no longer sang* to *Fred no longer sang*, likely due to coreference difficulties. For ex. 3, T5-r does not convert *Suddenly* to *Earlier* like BART-d, giving a false interpretation that Eric slipped after his rescuer’s arrival. BART-r does not mislead with *Suddenly*, but is ambiguous with no timex at all. Further, BART and T5 are more conservative than humans.

5.4 Overall Takeaways

Humans modify text greatly while successfully performing NAREOR. BART and T5 models perform decently with

<i>Method\Metric</i>	BERTScore	BLEU	METEOR	TOF-BERTScore	TOF-METEOR
Human rewritings	N/A	N/A	N/A	66.85	56.79
GPT2-d-2S	60.75	37.01	45.20	79.23	74.23
GPT2-r-2S	58.03	32.57	40.85	73.04	63.00
BART-d-1S	67.14	44.73	49.88	95.61	93.43
BART-d-2S	<u>67.93</u>	<u>46.03</u>	<u>50.54</u>	93.55	90.81
BART-r-2S	67.16	44.63	49.16	91.32	86.43
T5-d-2S	67.99	46.95	51.12	94.20	91.83
T5-r-1S	66.24	43.40	48.20	89.85	84.26
T5-r-2S	66.62	44.30	49.00	91.61	86.16

Table 4: Average automatic evaluation results on testSup (values multiplied by 100). Bold corresponds to best performance per metric, and underline second-best (excluding the TOF metrics which are mainly for validation).

<i>Metric</i>	<i>Correlation</i>	Fluency	Coherence	Logic	Plot-pres	Interest
BERTScore	Pearson	0.130 (4e-04)	0.139 (1e-04)	0.125 (0.001)	0.255 (1e-06)	0.111 (0.226)
	Spearman	0.106 (0.004)	0.124 (0.001)	0.127 (0.001)	0.211 (5e-05)	0.117 (0.201)
BLEU	Pearson	0.144 (9e-05)	0.140 (1e-04)	0.113 (0.002)	0.219 (3e-05)	0.174 (0.047)
	Spearman	0.130 (4e-04)	0.129 (4e-04)	0.123 (0.001)	0.179 (0.001)	0.171 (0.049)
METEOR	Pearson	0.107 (0.003)	0.125 (0.001)	0.108 (0.003)	0.203 (1e-04)	0.120 (0.191)
	Spearman	0.098 (0.008)	0.114 (0.002)	0.122 (0.001)	0.164 (0.002)	0.121 (0.187)

Table 5: Pearson and Spearman correlations between automatic and human evaluation metrics, with p-values in brackets. TOF metrics excluded as they are mainly for validation. Bold corresponds to highest correlation per human evaluation metric.

minimal but effective edits. GPT-2 models tend to repeat, hallucinate, and reduce text quality and plot preservation.

Based on human (§5.1) and automatic (§5.2) evaluation, BART-d-2S and T5-d-2S are the best models overall. BART-d-2S outdoes its reorder variant, possibly due to BART’s pretraining as a denoising autoencoder, closer to our denoise training method. For T5, both methods perform quite well and show potential. However, T5-d outperforms on plot-pres (Table 2), interest (Table 3), and automatic metrics (Table 4). The denoise training method appears to be slightly more effective, possibly because it is partially inspired by how humans perform NAREOR (see §3.1). These are the first two task-specific training methods for NAREOR which we propose ourselves, each approaching the task differently (see §3.1). 2S models also mostly outperform 1S ones, demonstrating that second stage finetuning improves upon the first.

BART and T5 models are quite effective, excelling at fluency, but have further room for improvement in coherence, logic, plot-pres, and interest. §5.3 shows they still suffer from several issues. Their conservative tendency may limit their NAREOR ability compared to humans. Overall, these models serve as strong initial baselines for NAREOR while underscoring the task’s difficulty and potential for exploration.

6 Applications of NAREOR

Sentence ordering involves reconstructing original sentence order of an unordered sentence set (Barzilay and Lapata 2008). NAREORC’s reordered stories could serve as a challenge set for sentence reordering models due to their non-linear narrative structure underrepresented in corpora. We use the implementation of Prabhumoye, Salakhutdinov, and Black (2020) to train i) M_{ext} , an external model on the SIS corpus (Huang et al. 2016), ii) M_{iid} , an in-domain model on first 20% of ROCStories’ train split. We test each on i) Control set $\{s_i\}_{i=1}^{i=n}$, input stories from testSup, ii) Challenge set

$\{s'_i\}_{i=1}^{i=n}$, reordered stories from testSup. Table 7 shows drastic drops across metrics (higher is better - see Prabhumoye, Salakhutdinov, and Black (2020)) for both M_{ext} and M_{iid} from control to challenge set, confirming our hypothesis.

Systems with ability to manipulate narrative variables like order could be important for automating pedagogical setups, especially for fine-grained language skills such as *argumentation in essay writing*. As Wingate (2012) explains, tutor understanding is found deficient and methods of feedback for students are inconsistent or vague. Language in school texts follows a characteristic register, which often differs from registers students handle in everyday conversation (Schleppegrell 2001). Models (e.g. NAREOR ones) which can control elements of register, e.g narrative order, can be used to tailor such content to intended settings and bridge this gap.

Systems that can generate event timelines for clinical narratives, e.g. admission notes and physical reports, is important for applications like medical document summarization (Bramsen et al. 2006; Reichert et al. 2010) and clinical decision making (Demner-Fushman, Chapman, and McDonald 2009). Raghavan et al. (2014) demonstrate that cross-narrative temporal ordering of medical events is vital to generating a comprehensive timeline over a patient’s history. Aligning multiple medical event sequences using coreference information and temporal relations has a large impact on their presentation and effectiveness. Our NAREOR models may be effective here and improve upon existing systems.

7 Related Work

There exists work on the sentence ordering task discussed in §6. For example, Chen, Qiu, and Huang (2016) learn pairwise orderings of sentences using a ranking model. Unlike sentence ordering, NAREOR involves reordering and rewriting a sequence of sentences to fit a new narrative order.

TALESPIN (Meehan 1975) was an early goal-based story

Method	Text
Original (ex. 1)	Jimmy wandered around town looking for a soda. Before he knew it, he was in a strange area. He was scared of strangers and didn't want to ask anyone. Soon a policeman came by and asked if he was lost. He told him that he was lost. {5,4,2,1,3}
BART-d	Jimmy told a policeman that he was lost. The policeman had come by and asked if he had been lost. Before he knew it, he was in a strange area. Jimmy had wandered around town looking for a soda. He was scared of strangers and didn't want to ask anyone.
BART-r	Jimmy told the policeman that he was lost. The policeman came by and asked if he had lost his wallet . Before Jimmy knew it, he was in a strange area. He had wandered around town looking for a soda. He was scared of strangers and didn't want to ask anyone.
T5-d	Jimmy told a policeman that he was lost. Soon after , a policeman came by and asked if he was lost. Before he knew it, he was in a strange area. He had wandered around town looking for a soda. He was scared of strangers and didn't want to ask anyone.
T5-r	Jimmy told a policeman that he was lost. A policeman came by and asked if he had a soda . Before he knew it, he was in a strange area. He had wandered around town looking for a soda. He was scared of strangers and didn't want to ask anyone.
Original (ex. 2)	Every morning, a bird woke Fred up for work at 6 AM. Fred began to like the bird. And he enjoyed the sound of its singing. One day, the bird no longer sang. He hoped nothing bad had happened to the bird. {4,1,3,5,2}
Human	Fred didn't hear the bird singing this morning which is unusual. The bird had been waking him up every single day at 6 AM for quite some time. He had grown accustomed to the bird being his alarm clock . Now he's worried that something might have happened to the poor bird. He almost considers the bird a friend at this point.
GPT2-r	One day, Fred's bird began to sing. Every morning, Fred woke him up for work at 6 AM. This was because he liked Fred . He hoped nothing bad had happened to Fred. This bird, however, did not like Fred .
BART-r	Fred no longer sang. Every morning, he woke Fred up for work at 6 AM. He enjoyed the sound of his singing. He hoped nothing bad had happened to him. He had begun to like the bird.
T5-d	One day, the bird no longer sang. Every morning, a bird woke Fred up for work at 6 AM. He had enjoyed the sound of its singing. He had hoped nothing bad had happened to the bird. After a while , Fred began to like the bird.
Original (ex. 3)	Eric was walking around a pond at a local park. He saw a turtle on a rock few feet offshore. He stepped into the water to try to catch the turtle. Suddenly, he sank knee deep into the mud and got stuck. Someone walking on the shore ran over to rescue Eric. {5,4,1,3,2}
Human	Someone who had been walking on the shore ultimately ran over and rescued Eric from the mud. Eric's knee had sunk deep into the mud, rendering him stuck. Earlier, Eric had been walking around a pond while at a local park. Trying to catch a turtle in the pond, Eric stepped into the water. Eric did so because he had seen a turtle poached on a rock few feet offshore.
BART-d	Someone walking on the shore ran over to rescue Eric. Earlier , he had sunk knee deep into the mud and got stuck. Eric was walking around a pond at a local park. He stepped into the water to try to catch the turtle. He had seen a turtle on a rock few feet offshore.
BART-r	Someone walking on the shore ran over to rescue Eric. Eric sank knee deep into the mud and got stuck. He was walking around a pond at a local park. He had stepped into the water to try to catch the turtle. He saw the turtle few feet offshore.
T5-r	Someone walking on the shore ran over to rescue Eric. Suddenly , Eric sank knee deep into the mud and got stuck. He was walking around a pond at a local park. He stepped into the water to try to catch the turtle. He saw the turtle few feet offshore.
Original (ex. 4)	Joey went to Disneyland with his family. He was so excited to meet Mickey Mouse. When he met Mickey Mouse he was so excited he couldn't speak! Joey finally said hi to Mickey and it was great! Joey had a great time. {5,2,4,1,3}
GPT2-d	Joey had a great time. He was so excited to meet Mickey Mouse. He had finally said hi to Mickey and it was great! He had gone to Disneyland with his family. When he met Mickey Mouse he was so excited he couldn't speak
GPT2-r	Joey had a great time on the roller coaster at Disneyland. He was so excited to meet Mickey Mouse he couldn't speak! (...) He was so excited that he went to the park with his family. He was so excited when he met Mickey Mouse and he was so excited .
BART-r	Joey had a great time at Disneyland . He was so excited to meet Mickey Mouse. He finally said hi to Mickey and it was great! He had gone to Disneyland with his family. When he met Mickey he was excited he couldn't speak!
T5-d	Joey had a great time at Disneyland . He was so excited to meet Mickey Mouse. He had finally said hi to Mickey and it was great! He had gone to Disneyland with his family. When he met Mickey Mouse he was so excited he couldn't speak!

Table 6: Qualitative examples for testSup. Target perms are in brackets beside original stories. *d* refers to denoise, and *r* to reorder (2S models).

Model	TestSet	SentAcc	Rouge-S	LCS	Kendall τ
M_{ext}	Control	76.35	48	59.1	0.57
	Challenge	52.4	24.7	29.7	0.12
M_{iid}	Control	66.4	85.3	84.8	0.75
	Challenge	21.9	49.6	58	0.03

Table 7: Sentence ordering on control vs. challenge sets.

generator. There has since been work on related tasks like story cloze test (Mostafazadeh et al. 2016b, 2017) and generation from prompts (Fan, Lewis, and Dauphin 2018; See et al. 2019). Some works explore controllable variants, e.g. with keywords as control (Peng et al. 2018). NAREOR is distinct as it aims to preserve the underlying plot while controlling a story-level aspect for an already-complete story.

There is also narrative order visualization work. For example, Kim et al. (2017) annotate story order for movie scripts and visualize narrative order as a function of story order.

8 Conclusion and Future Work

We proposed the task of Narrative Reordering (NAREOR) and introduced a dataset, NAREORC, with task-specific training methods and evaluation metrics, and experimented with T5, BART, and GPT-2. Extensive evaluation and qualitative analysis demonstrated that our models are quite effective but can be further improved, and that NAREOR is challenging with potential for further exploration. We showed that NAREOR can be used to produce more interesting story variations and as a challenge set for tasks like sentence ordering.

Future directions include exploring training ideas better emulating human rewrites. NAREOR can be explored as document-level paraphrasing for applications like data augmentation for document tasks, adversarial sets for more temporal tasks, and applications for education and medicine discussed in §6. We also hope our work drives investigation of more challenging task variations (e.g. sub-sentential). Lastly, NAREOR's controllability aspect can be investigated further.

References

- Banerjee, S.; and Lavie, A. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 65–72.
- Barzilay, R.; and Lapata, M. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1): 1–34.
- Bos, G. 1995. Jewish Traditions on Strengthening Memory and Leone Modena’s Evaluation. *Jewish Studies Quarterly*, 2(1): 39–58.
- Bramsen, P.; Deshpande, P.; Lee, Y. K.; and Barzilay, R. 2006. Inducing Temporal Graphs. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 189–198. Sydney, Australia: Association for Computational Linguistics.
- Chang, A. X.; and Manning, C. D. 2012. SUTIME: A library for recognizing and normalizing time expressions. In *LREC*, volume 2012, 3735–3740.
- Chen, X.; Qiu, X.; and Huang, X. 2016. Neural sentence ordering. *arXiv preprint arXiv:1607.06952*.
- Demner-Fushman, D.; Chapman, W. W.; and McDonald, C. J. 2009. What can natural language processing do for clinical decision support? *Journal of Biomedical Informatics*, 42(5): 760–772. Biomedical Natural Language Processing.
- Elder, H.; Gehrmann, S.; O’Connor, A.; and Liu, Q. 2018. E2E nlg challenge submission: Towards controllable generation of diverse natural language. In *Proceedings of the 11th International Conference on Natural Language Generation*, 457–462.
- Fan, A.; Lewis, M.; and Dauphin, Y. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 889–898.
- Genette, G. 1983. *Narrative discourse: An essay in method*, volume 3. Cornell University Press.
- Halliwel, S.; et al. 1998. *Aristotle’s poetics*. University of Chicago Press.
- Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2019. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*.
- Huang, T.-H. K.; Ferraro, F.; Mostafazadeh, N.; Misra, I.; Agrawal, A.; Devlin, J.; Girshick, R.; He, X.; Kohli, P.; Batra, D.; Zitnick, C. L.; Parikh, D.; Vanderwende, L.; Galley, M.; and Mitchell, M. 2016. Visual Storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1233–1239. San Diego, California: Association for Computational Linguistics.
- HuggingFace. 2020. NeuralCoref. [Online; accessed 29-September-2020].
- Kendall, M. G. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2): 81–93.
- Keskar, N. S.; McCann, B.; Varshney, L. R.; Xiong, C.; and Socher, R. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.
- Kim, N. W.; Bach, B.; Im, H.; Schriber, S.; Gross, M.; and Pfister, H. 2017. Visualizing nonlinear narratives with story curves. *IEEE transactions on visualization and computer graphics*, 24(1): 595–604.
- Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 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*, 7871–7880. Online: Association for Computational Linguistics.
- Meehan, J. R. 1975. Using Planning Structures to Generate Stories. *American Journal of Computational Linguistics*, 78–94. Microfiche 33.
- Montfort, N. 2007. Ordering Events in Interactive Fiction Narratives. In *AAAI Fall Symposium: Intelligent Narrative Technologies*, 87–94.
- Morgan, M. S. 2017. Narrative ordering and explanation. *Studies in History and Philosophy of Science Part A*, 62: 86–97.
- Mostafazadeh, N.; Chambers, N.; He, X.; Parikh, D.; Batra, D.; Vanderwende, L.; Kohli, P.; and Allen, J. 2016a. A corpus and cloze evaluation for deeper understanding of common-sense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 839–849.
- Mostafazadeh, N.; Misra, I.; Devlin, J.; Mitchell, M.; He, X.; and Vanderwende, L. 2016b. Generating Natural Questions About an Image. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1802–1813. Berlin, Germany: Association for Computational Linguistics.
- Mostafazadeh, N.; Roth, M.; Louis, A.; Chambers, N.; and Allen, J. 2017. LSDSem 2017 Shared Task: The Story Cloze Test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, 46–51. Valencia, Spain: Association for Computational Linguistics.
- Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.
- Peng, N.; Ghazvininejad, M.; May, J.; and Knight, K. 2018. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, 43–49.
- Prabhumoye, S.; Salakhutdinov, R.; and Black, A. W. 2020. Topological Sort for Sentence Ordering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2783–2792.
- Propp, V. 2010. *Morphology of the Folktale*, volume 9. University of Texas Press.

- Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multi-task learners. *OpenAI Blog*, 1(8): 9.
- Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140): 1–67.
- Raghavan, P.; Fosler-Lussier, E.; Elhadad, N.; and Lai, A. M. 2014. Cross-narrative Temporal Ordering of Medical Events. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 998–1008. Baltimore, Maryland: Association for Computational Linguistics.
- Rajani, N. F.; McCann, B.; Xiong, C.; and Socher, R. 2019. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4932–4942. Florence, Italy: Association for Computational Linguistics.
- Ramanujan, A. K. 1991. Three hundred Ramayanas: Five examples and three thoughts on translation. *Many Ramayanas: The Diversity of a Narrative Tradition in South Asia*, 22–49.
- Reichenbach, H. 1947. Elements of symbolic logic. *London: Dover Publications (1947)*.
- Reichert, D.; Kaufman, D.; Bloxham, B.; Chase, H.; and Elhadad, N. 2010. Cognitive analysis of the summarization of longitudinal patient records. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, 2010: 667–71.
- Schank, R. C. 1972. Conceptual dependency: A theory of natural language understanding. *Cognitive psychology*, 3(4): 552–631.
- Schleppegrell, M. J. 2001. Linguistic features of the language of schooling. *Linguistics and education*, 12(4): 431–459.
- See, A.; Pappu, A.; Saxena, R.; Yerukola, A.; and Manning, C. D. 2019. Do Massively Pretrained Language Models Make Better Storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 843–861. Hong Kong, China: Association for Computational Linguistics.
- Wiegrefe, S.; and Marasovic, A. 2021. Teach Me to Explain: A Review of Datasets for Explainable Natural Language Processing. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.
- Wingate, U. 2012. ‘Argument!’ helping students understand what essay writing is about. *Journal of English for academic purposes*, 11(2): 145–154.
- Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2019. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.