NAREOR: The Narrative Reordering Problem

Varun Gangal * 1 Steven Y. Feng * 1 Malihe Alikhani 2 Teruko Mitamura 1 Eduard Hovy 1

1 Language Technologies Institute, Carnegie Mellon University
2 School of Computing and Information, University of Pittsburgh
{vgangal,syfeng,teruko,hovy}@cs.cmu.edu, 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, 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 (Wiegreffe 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.

Our work is not entirely a drift 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 disentan-
gling events from stories which are inherent in NAREOR.

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

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 coherence,
tense, and other discourse dependencies. For example, narra-
tive order affects tense since it can change the first 2 of 3
Reichenbach times (Reichenbach 1947) that together deter-
mine tense - speech, reference, and event time. NAREOR
involves pinpointed and critical edits; a single missed or in-
correct edit can result in an entirely different or invalid plot.
Since \( \pi' \) 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, NAREOR-
C, with human rewritings of stories from ROCStories
(Mostafazadeh et al. 2016a) in non-linear orders. We conduct
a thorough analysis, examining various ways humans modi-
fy 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 \( \approx 98.5 \text{K} \) five-sentence
English stories. For the dev and test splits, each example

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

\(^2\) Code+data at github.com/vgtomahawk/NAREORCamReady.

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 \( \pi' \) is not part of the ROCStories input. We devise a
randomized procedure to assign a reasonable \( \pi' \) for each example. We sample 3 permutations from the set of non-
identity \( n-1 \) permutations.\(^3\) We find Kendall \( \tau \) correlations
(Kendall 1938) between identity permutation \( I_n \), \{1,2,3,4,5\},
and each of the three permutations, retaining the lowest as
\( \pi' \). 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 anno-
tate 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, Very-
Hard. 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 \( \approx 1.2 \times \) 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 ( \( s_3 \) and \( s'_3 \) in Figure
1) and pronouns ( \( s_3 \) and \( s'_3 \) 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.
In-
creased n-gram repetition could have reasons similar to length
increase, causing cross-sentence repetition. Figure 1 demon-
strates this: \( S \) only has one instance of money. Conversion
of inherit any of it ( \( s_3 \) ) to inherit any of the money ( \( s'_3 \) ) and
enough to take time ( \( s_3 \) ) to enough money to take some time
( \( s'_3 \) ) 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 frac-
tion (pos=VBD) (e.g. He broke it) decreases 60.9\% \( \rightarrow \) 54.6\%.

\(^3\) In our case, \( n = 5 \) as we experiment with ROCStories.
Other verb-related pos fractions remain fairly constant. Increase in past participle can be explained by frequent conversion 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 mindens). 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 noise 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-written reorderings 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 \( S_i \) are first reordered as per \( \pi_i \) to produce \( S'_{\text{naive}} \), 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 \( S'_{\text{naive}} \), and the outputs are the human reorderings 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 \( <a> \) to \( <e> \). We specify \( \pi_i' \) as a sequence of these, separated from \( S \) by \( <\text{sep}> \). 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 \( \{<X>\} \) 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 \( <\text{st}> \) to mark the end of the input. An illustration of the scheme follows: \( <a> \) *Since I had front seat tickets, I was able to directly see <XI>.* \( \langle b \rangle <\text{XI}> \) tried to reach out with <\text{X1}> <\text{X2}>, \( \langle c \rangle I \) grabbed <\text{X2}> and <\text{XI}> pulled me on stage. \( \langle d \rangle <\text{XI}> \) began to sing. \( \langle e \rangle The \) concert had started. \( \langle <\text{sep}> \rangle <\text{e}> \) \( \langle a \rangle <\text{b}> <\text{c}> <\text{XI}> The \) music artist <\text{X2}> her hand <\text{st}>.

- **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 \( S'_{\text{naive}} \) would hurt output fluency. We instead train in the inverse direction \( S_{\text{naive}}, \pi_i^{-1} \rightarrow S \), where \( \pi_i^{-1} : \pi_i^{-1}(\pi_i') = I \) is the inverse permutation of \( \pi_i' \). To reduce train-test mismatch, we use the inverse formulation for each 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.

- **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**

We 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 \( s'_1 \). 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

\(^4\)See §4 for further training/finetuning details.

\(^5\)Correlates well with human evaluation as shown in §5.

\(^6\)Assuming the values are multiplied by 100.
sufficient. Lower values indicate more changing of the text which may be necessary for certain narrative reorderings.

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 $S_i^{m2}$ 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.\footnote{Nucleus sampling did not work as well for BART and T5.} 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.\footnote{See Appendix C for more finetuning/generation details.}

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.
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.

**Fluency, Coherence, Logic:** Original stories are the highest for all three metrics with human rewritings second for coherence 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

| Method/Metric | Fluency | Coherence | Logic | Plot-pres |
|---------------|---------|-----------|-------|-----------|
| Original stories | 4.299   | 4.0       | 3.853 | 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   | 3.503     | 3.493 | 3.722     |
| BART-r-2S | 3.757   | 3.439     | 3.495 | 3.861     |
| T5-d-2S | 3.764   | 3.419     | 3.5   | 3.889     |
| T5-r-1S | 3.655   | 3.378     | 3.486 | 3.847     |
| T5-r-2S | 3.784   | 3.595     | 3.520 | 3.861     |

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

| Method | Interest | Human | BART-d | BART-r | T5-d | T5-r |
|--------|----------|-------|--------|--------|------|------|
|        |          |       |        |        |      |      |
|        | 3.75     | 3.56   | 3.485  | 3.533  | 3.35 |

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

**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, 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 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 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 and has no timex.
| Method | Metric | 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 | 95.43 |
| BART-d-2S | 67.93 | 46.03 | 50.54 | 93.55 | 93.81 |
| 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).

| Metric | Correlation | 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 (3e-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 (9e-04) | 0.125 (0.001) | 0.179 (0.001) | 0.171 (0.049) |
| METEOR | Pearson | 0.107 (1e-03) | 0.125 (0.001) | 0.108 (0.001) | 0.205 (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.

5.4 Overall Takeaways

Humans modify text greatly while successfully performing NAREOR. BART and T5 models perform decently with 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 denoising 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) $M_{idd}$, an in-domain model on first 20% of ROCStories’ train set. We test each on i) Control set $\{s_i\}_{i=1}^n$, input stories from testSup, ii) Challenge set $\{s'_i\}_{i=1}^n$, 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 $M_{idd}$ 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, are important for applications like medical document summarization (Bransen 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 — e.g., Chen, Qiu, and Huang (2016) learn pairwise orderings of sentences using a ranking model. Unlike sen-
Kendall

TestSet

LCS

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 couldn’t speak! (...)

Someone walking on the shore ran over to rescue Eric.
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 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.

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.

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 had hoped nothing bad had happened to Fred. This bird, however, did not like Fred.

Jimmy told the policeman that he was lost.

Challenge:

Control

SentAcc  Rouge-S  LCS  Kendall r

M_{ext}

Control

70.35  48  59.1  0.57

Challenge

52.4  24.7  29.7  0.12

M_{lid}

Control

66.4  85.3  84.8  0.75

Challenge

21.9  49.6  58  0.03

Table 6: Qualitative testSup examples. Target perms in brackets along original stories. d & r refer to denoise & reorder.

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

narrative order visualization work. For example, Kim et al. (2017) visualize narrative order as a function of story order.

8 Conclusion and Future Work

We proposed the NAREOR task and introduced a dataset, NAREORC, with task-specific training methods and evaluation metrics, and experimented with T5, BART, and GPT-2. Extensive evaluation and analysis showed that our models are effective but can be further improved, and that NAREOR is challenging with further exploration potential. We showed that NAREOR can create interesting story variations and challenge sets for tasks like sentence ordering.

Future directions include exploring training ideas better emulating human rewrites. NAREOR can be explored as document-level paraphrasing for data augmentation, adversarial sets for more temporal tasks, and applications for education/medicine (see §6). We also hope our work drives inquiry into harder task variations of NAREOR (e.g. sub-sentential).
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.

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.

Halliwell, 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. github.com/huggingface/neuralcoref. [Online; accessed 29-September-2020].

Kendall, M. G. 1938. A new measure of rank correlation. Biometrika, 30(1/2): 81–93.

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 commonsense 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 multitask 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.

Wiegreffe, 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.