

Content Planning for Neural Story Generation with Aristotelian Rescoring

Seraphina Goldfarb-Tarrant^{*†}, Tuhin Chakrabarty^{*‡}, Ralph Weischedel^{*}, Nanyun Peng^{*§}

^{*} Information Sciences Institute, University of Southern California

[†] University of Edinburgh, [‡] Columbia University, [§] University of California, Los Angeles

s.tarrant@ed.ac.uk, tuhin.chakr@cs.columbia.edu,

weisched@isi.edu, violetpeng@cs.ucla.edu

Abstract

Long-form narrative text generated from large language models manages a fluent impersonation of human writing, but only at the local sentence level, and lacks structure or global cohesion. We posit that many of the problems of story generation can be addressed via high-quality content planning, and present a system that focuses on how to learn good *plot structures* to guide story generation. We utilize a plot-generation language model along with an ensemble of rescoring models that each implement an aspect of good story-writing as detailed in Aristotle’s *Poetics*. We find that stories written with our more principled plot-structure are both more relevant to a given prompt and higher quality than baselines that do not content plan, or that plan in an unprincipled way.¹

1 Introduction

Despite many recent advances in Natural Language Generation, successful creative narrative composition remains elusive. Current neural approaches are plagued by difficulty in mastering structure, will veer between topics, and lack long-range cohesion. They successfully imitate the fluency and style of human writing, but on closer inspection sentences do not fit together to form a whole, and the reader is left with the impression that the generation has no content (See et al., 2019). This lack of structure also degrades the relevance of generations conditioned on a prompt or other source text - a strong language model will repeat key phrases from a given prompt but will not remain on topic. These issues are illustrated in the Naive Generated Story in Table 1, where many of the sentences individually are fine, but do not fit together as one story, and do not all relate to the prompt.

We hypothesise that this problem can be addressed with a focus on deeper latent narrative structures. In Aristotle’s *Poetics*, one of the most enduring treatises on the craft of writing good stories, the philosopher lays out the elements of story in order of importance. They are:

1. event choice and arrangement
2. character
3. relevant content²
4. diction

An amateur masters skills later in the list, but mastery of event choice and event arrangement is what distinguishes a good writer (Aristotle). Next is character, then relevance, and only finally do style and diction matter.

This philosophical framework fits remarkably well into the traditional Natural Language Generation Pipeline approach that emphasizes Content Planning (Reiter and Dale, 1997). The pipeline divides generation into three steps: *Content Planning*, *Microplanning* and *Surface Realization*, where at each step input is modified and refined, getting closer to the final textual output. Incorporating a plot in order to generate stories can then be viewed as a proxy for Content Planning/MicroPlanning before a language model makes use of it to convert it to a readable and grammatically correct natural language output (Surface Realization).

Inspired by both the Aristotelian and Content Planning Frameworks, we develop a novel system for story generation. We focus on developing a system that can learn to expertly select events, characters, and relevant content, and write good *plot structures*. After the work on the plot is complete, a large language model can then do what it does best and fill in the descriptions, details, and local specifics of each story.

¹Code at <https://github.com/PlusLabNLP/story-gen-BART>

²This directly translates as *thought* in the original, but is explained to clearly mean the selection of relevant information

Prompt: Write the most beautiful end to the universe that you can imagine

Naive Plot with Semantic Role Labelling Tags:

<A1> The universe <V> end </s> </s> <A0> ent 0 <V> see <A1> ent 3 </s> </s> <V> dying <A1> ent 1 # <A0> ent 1 <V> left <A1> ent 0 </s> <A0> the last human <V> live </s> <A1> ent 6 <V> end # <A1> ent 2 <V> come </s> <A1> the last one <V> die </s> <A1> a universe of life <V> left

Revised Plot with Semantic Role Labelling Tags:

<A2> The light <V> filled <A2> the sky </s> </s> </s> <A2> A bright flash <V> lit </s> </s> # <V> began <A2> to grow # <A2> the stars <V> grow </s> </s> <V> began <A2> ent 0 <A2> to fade # <A2> The stars <V> fade </s> </s> <A0> ent 2 <V> looked <A2> ent 1 me </s> <V> dying <A2> star

Naive Generated Story:

“You’re dead. I’ll be dead, and there’ll always be no more of you.” * “No, my dear. It’s not true!” - The last of the world is dead, * “It’ll never be over. *The universe* will n’t end. Not yet. But I’ll see it, one way or another. There is no other way.” * “My last thought, the *dying* man *left* me. He told that I was *the last human* to live. And he told me the universe would end, and the end will come soon. That your death would be *the last one* to die, and the first one to be the most beautiful. * “There is *a universe of life left*, my dear, and that’s all there is.

Generated Story (from revision):

It was the end of the universe. *The light filled the sky*. The galaxy. All the light in all the worlds, and all the stars. *A bright flash*. The first light. And then a bright flash of light... then a flash of darkness. Then a bright light. A flash of blackness and then the stars *began to grow*. It was a bright, beautiful bright, light. And then, slowly the *stars* started to *grow* in number. There were millions and millions of stars.. But, slowly but surely, all the galaxies *began to fade*. Slowly, but surely. Finally, it *looked* at me with its eyes. One *dying star*.

Gold Story:

It slept, and dreamt of twirling lights. It cradled countless twirling lights, all dancing in their flashing ways, dancing to the sound of a faraway flute. The light was vibrant and warm and spun all around It. The tiny lights grew with the tune, and more lights came to dance within the luminescence. It was surrounded by light, all waltzing in their ways to the distant music. It, too, swirled with the swelling symphony, watching the lights. There were lights of all different sizes and shades and brightness, but each was its own, unique light. Each one was brilliant. Each one was beautiful. All dancers to the Song...

Table 1: Our proposed plot and story generation structure. We generate a Naive Plot, revise it with Aristotelian rescorers, then generate a story. In plots, <V> denotes verbs while <A{0, 1, 2}> denote arguments. ent {0...n} are entities. We removed newline symbol <P> from the generated stories and detokenized for better display.

For plot generation, we employ a few event-choice and event-arrangement rescoring models which assist in building the arc and cohesion of the plot, a character rescoring model that helps select which characters appear where, and a relevance model that is responsible for keeping the plot structure and the story on topic. As both improving plot-generation via rescoring and using an Aristotelian framework for neural generation are novel concepts, there is no previous work on how to implement them in practice.

Our contributions are: 1) we propose to leverage the principled Aristotelian framework for content planning, 2) we propose an implementation of the framework using a revision-based approach via several rescoring models 3) we show strong experimental results against 4 baselines.

2 Background

Existing work in neural story generation has established the strength of adding a content planning stage to structure the generated content (Yao et al., 2019; Fan et al., 2019) (discussed in more detail in Section 7). Specifically, this line of work trains a pipeline with one model that generates from *prompt* \rightarrow *plot* and another that generates from *prompt*

+ *plot* \rightarrow *story*. It modifies the standard conditional generation task with a source $x = x_1 \dots x_n$ (in this case, a prompt) and target $y = y_1 \dots y_n$ (in this case, a story) to condition also on an abstract intermediate representation \mathbf{z} . Note that the approach is not truly modelling $p(\mathbf{y}|\mathbf{x})$ since that would involve summing over all \mathbf{z} . Instead, it models $p(\mathbf{y}, \mathbf{z}|\mathbf{x}) = p(\mathbf{z}|\mathbf{x})p(\mathbf{y}|\mathbf{z}, \mathbf{x})$, but only shows the generated story \mathbf{y} at inference time.

This is a more controllable task than open-domain generation conditioned on only a prompt \mathbf{x} , provided that a good interim structure \mathbf{z} can be learnt. We follow this line and explore ways to improve plot planning to close the gap between stories generated from gold plots and those from model-generated plots.

Plot Representation As there are no large datasets with parallel gold-standard plots and stories, all work on plot generation depends on silver-standard plots extracted from stories. We follow Fan et al. (2019) to represent plots in Semantic Role Labelling (SRL) format. We run coreference resolution to identify entities, and use a compression algorithm to discard less salient information.³

³Our representation is conceptually quite similar to that of Fan et al. (2019), but is not identical, as their code is un-

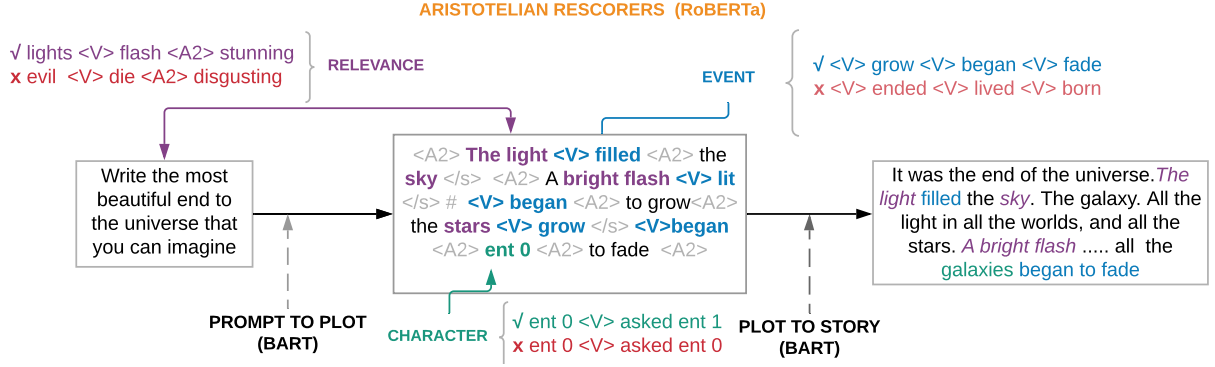


Figure 1: The full story generation pipeline. Base generation models in gray. Aristotelian rescorer input in color, with example positive & negative training pairs. Plot structure shown with SRL tags for verbs and their arguments.

3 Approach

We focus on learning the best interim plot structure between the input prompt and the story surface realisation. As such, we learn the plot model $p(\mathbf{z}|\mathbf{x})$ and the story model $p(\mathbf{y}|\mathbf{z}, \mathbf{x})$ by fine-tuning a pre-trained conditional language model BART (Lewis et al., 2019) on 1) pairs of prompts and extracted silver standard plot-structures, and 2) pairs of prompts + plots and stories, respectively. Full implementation details can be found in Appendix A.4.

We propose to modify the decoding objective to incorporate input from each Aristotelian rescoring model $a \in A$ (the complete set of rescoring models, detailed further in Section 3.1) and re-rank the original, or “naive” plot model hypotheses, bringing the plot representation closer to each rescoring model’s specialty and desirable story attribute. A diagram of our final system in Figure 1 shows each step of the generation process. The modified decoding objective becomes:

$$f_{\lambda}(\mathbf{x}, \mathbf{z}) = \sum_i^m -\log p(z|i, \mathbf{x}) + \sum_j^{|A|} \lambda_j a_j(\mathbf{x}, z_{i...m}) \quad (1)$$

where λ_j is the learned weight of the score given by a_j , as detailed in Section 3.3.

3.1 Aristotelian Rescoring Models

For all of our rescoring models, we train classifiers to *distinguish* positive examples - the silver extracted plots, and negative examples - plots that are worse with regard to the aspect that we desire to encode, given the prompt \mathbf{x} . The intuition is that if the rescoring model can learn a particular principle, it can assist the plot-generation language model in creating content that encapsulates

available. We used AllenNLP (Gardner et al., 2018) to run the SRL model (He et al., 2017) and Co-reference model (Lee et al., 2017) and determined our own compression algorithm experimentally. Further details in Appendix A.6

that principle. Mathematically, the classifiers learn $p(l|\mathbf{x}, \mathbf{z}) = \frac{p(\mathbf{x}, \mathbf{z}, l)}{p(\mathbf{x}, \mathbf{z})}$, and we use the probability of the plot being a positive example (a more plausible plot) as our Aristotelian rescoring model:

$$a_j(\mathbf{x}, z_{i...m}) = p_j(l_{pos}|\mathbf{x}, z_{i...m}). \quad (2)$$

What differs for each model a_j that specialises in a different Aristotelian aspect is the set of negative examples that we generated to capture the type of information it has learnt to discriminate, and the features it learns. We give more details about each Aristotelian rescorer as follows.

Event Rescorers. The SRL extracted plots provide us with a structure that is very similar to event representations in event extraction literature. We thus consider an *Event* to be composed of an action-based verb and its subject and object (a verb, subject, object tuple).⁴ We experiment with three different ways to construct positive and negative event examples. SRL based plots are structured, and a random shuffle would be trivial to distinguish, so we need more nuanced ways to learn good event choice and arrangement. We try:

- **inter-sentence shuffled events** we permute all sentences as a full chunk, and keep all events within a sentence together.
- **intra-sentence shuffled events** we permute the event tuples within a sentence, but keep each verb and its arguments together.
- **verb-shuffled events** we permute only the event verbs within a sentence, leaving their arguments and contexts unchanged.

Each of these has specific strengths and weaknesses. *Inter-sentence shuffling* is closest to work

⁴With modal verbs generally excluded in a stop list, which is included in Appendix A.6

on Narrative Event Chains (Chambers and Jurafsky, 2008) and script-learning, which represent the fact that certain events are more likely to causally follow other events rather than precede them. However, since inter-sentence noising scopes globally over the entire plot structure, it is a harder task and may be difficult for the model to discriminate patterns. *Intra-sentence shuffling* is the same task but restricted to a more local scale, which makes the patterns clearer and more learnable but cannot capture long-distance Event Chains inter-sententially. It is also more sensitive to the style of a given story, as stories have a variable number of events per sentence. Finally, *verb-shuffling* focuses on verbs as the salient element of an event, and should teach both principles of verb ordering and of verb suitability for context, and avoid artifacts from reordering arguments. However, since verbs are shuffled naively, the task can in some cases be too easy due to differences in verb selectional preferences.⁵

Character Rescorers. We represent character trajectory by distinguishing which character should appear at what point in the story. We create training examples by taking each entity and all the preceding plot tokens up until the entity, and having the rescoring model choose between the true entity and a randomly sampled entity. The character rescorer must then distinguish between points in the plot when a pre-existing entity should reappear, and if so which one, or whether a new entity should be introduced. The intuition is that this should encode typical patterns of a character’s actions and relationships in particular contexts.

Relevance Rescorers. We take an approach similar to prior work on learning to discriminate between random and true continuations of story sentences (Holtzman et al., 2018). We consider pairs of prompts and plots, where a positive example is the true plot and the negative is a randomly selected plot from elsewhere in the training data. This prompt and plot pairing is a much more difficult task than pairing context and continuation sentences, but once trained, this rescorer is expected to tell which kinds of plot words, verbs, and SRL patterns belong with which kinds of prompts.

3.2 Rescoring Model Architecture

There is an inherent tension in training a useful rescoring model: discrimination tasks for which

⁵For instance, an exclusively transitive verb could be swapped into the space of an intransitive one, and the edit would be trivially recognisable based on grammatical patterns.

even simple models can perform well may have inherent data artifacts and therefore not be helpful for modeling plots. However, discrimination tasks that are so hard that all models have low accuracy are also likely to be unhelpful. We experiment with three different architectures for our rescorsers. We start with ngram-baseline models⁶ to better judge the baseline difficulty of a given task and take artifacts of data creation into account. This is more informative than random chance accuracy. We also experiment with augmented versions of the CNN architectures used in Holtzman et al. (2018), and RoBERTa models (Liu et al., 2019), and find RoBERTa to have the best performance for each Aristotelian concept.

1. **XGBoost with ngrams:** We used n-grams in range of (1,4) as features and trained a XGBoost model with 200 gradient boosted trees with a maximum tree depth of 5.
2. **CNN with maxpooling:** We used a CNN-based architecture (Holtzman et al., 2018) but augmented it with BART position and subword encodings because our event tasks are ordered, so pooled or averaged representations that don’t represent permutations differently would be indistinguishable.
3. **RoBERTa-large** (Liu et al., 2019) has shown excellent performance on various sentence pair classification tasks. We expect this large pre-trained language model to be more efficient in being able to discriminate between a well-formed sequence of words and a poorer one. To this end we finetune RoBERTa-large with a classification specific final layer as the final option to build rescorer models.

Accuracies for different rescorer architectures by aspect are shown in Table 2. As we hypothesised from the nature of many of the tasks (Section 3.1), the inter-sentence shuffled task is more difficult because the noising is global. This is reflected in the barely above chance scores of the ngram-baseline. The Verb-shuffling high ngram-baseline performance shows that our suspicions about this task being easier were also correct. Intra-shuffling was the only surprise, and turned out to be more difficult than we expected and to have the largest gap between baseline ngram and CNN performance. RoBERTa scores are high across the board, so we use RoBERTa for all models in the final system.

⁶Implemented in Scikit-learn (Pedregosa et al., 2011)

Plot Elements		XGB	CNN	RoBERTa
Relevance		55.2	56.0	89.3
Entity		64.1	76.0	92.3
Event	Inter-Shuffled	55.4	53.0	96.6
	Intra-Shuffled	58.5	75.0	97.3
	Verb-Shuffled	76.1	75.0	95.8

Table 2: Accuracy for training Rescorers using three different architectures.

3.3 Mixture Weight Training and Ablations

We learn optimal weights for rescorers online during decoding using a held-out validation set V , and use these weights during inference via sampling. We minimize Margin Ranking Loss of the negative log probability of each validation sample between the gold (\mathbf{z}) and hypothesised ($\hat{\mathbf{z}}$) plot structures.

$$\mathcal{L}_{mix} = \sum_{\lambda} \sum_{\mathbf{x}, \mathbf{z} \in V} \sum_{i=1}^n \max(0, -(f_{\lambda}(\mathbf{x}, \mathbf{z}_{<i}) - f_{\lambda}(\mathbf{x}, \hat{\mathbf{z}}_{<i}))) \quad (3)$$

where i indexes the word position, n denotes the plot length, f_{λ} is the same as in Equation 1, and we are training the λ weights with this objective.⁷ We train mixture weights both for combinations of rescorers and for ablations using each rescorer individually, to isolate the contribution of each one.

Mixture weight training accuracy is in Table 3,⁸ which we report as Ranking Accuracy, the number of samples where the generation has higher probability than the gold. There we also include our automated plot metrics on the validation set for each ablation (further detail on those metrics in Section 4.2). As Table 3 shows, Inter-event is the strongest of the individual rescorers, though all five together achieve the best performance. This seems to indicate that each method of creating negative event examples is encoding a separate helpful piece of information, rather than one of them alone being the best approach.

4 Experimental Setup

Dataset. We use the Writing Prompts dataset (Fan et al., 2018), which is a large collection of user-generated stories along with their associated prompts from Reddit, to benchmark our models. It

⁷We experiment with learning on each token (in which case, each validation sample will contain a few hundred updates) vs. on each complete sample (an entire plot). We find both approaches to perform similarly (with every-token gaining 1% accuracy) and use an every-token approach in for our final system. Mixture weight hyperparameters can be found in Appendix A.5

⁸Our 4-scorer ensemble subtracts out the Intra-event model as it is conceptually a more local version of the Inter-event model, and has a lesser effect on Ranking Accuracy

Rescorer	RA \uparrow	V:T \uparrow	E \uparrow
All 5	0.43	2.07	7.15
All 4 (-intra)	0.35	1.72	8.78
Inter-event	0.37	1.76	8.87
Intra-event	0.16	2.01	6.58
Verb-event	0.15	1.66	8.89
Entity	0.17	1.68	9.18
Relevance	0.17	1.71	8.89

Table 3: Ranking accuracy (RA) and generated plot metrics i.e Verb:Token ratio (V:T), Entities(E) for rescorer ablations

is particularly suited to this task since it is both hierarchical (contains pairs of titles and stories, which enables the use of a plot as an interim step) and contains many diverse long-form stories that are very challenging to learn to structure.⁹

4.1 Baselines

We compare our generations to the two strongest recent story generation systems as well as two ablated versions of our own system.¹⁰

Targeted Common Sense Grounding Model.

Mao et al. (2019) propose a multi-task learning scheme to achieve quantitatively better common sense reasoning in pre-trained language models by leveraging auxiliary training signals from datasets designed to provide common sense grounding.

Knowledge-Enhanced Commonsense Model.

Guan et al. (2020) devise a knowledge-enhanced pretraining model for commonsense story generation leveraging external knowledge bases on the ROCStories dataset (Mostafazadeh et al., 2016). To further capture causal and temporal dependencies between sentences in a reasonable story, they employ multi-task learning which combines a discriminative objective to distinguish true and fake stories during fine-tuning.

Prompt to Story. This fine tunes the BART model directly with the prompt and story pairs without access to a plot structure.

Naive Plot. This utilizes a plot structure to write a story, but does not incorporate the Aristotelian Rescorer ensemble.

4.2 Metrics

We use automatic metrics to evaluate 1000 randomly selected prompts and their associated plot

⁹Dataset statistics are in Appendix A.2

¹⁰We do not compare to the strong system of Fan et al. (2019), as they do not release their code and the authors were unable to make their output available after being contacted. However Guan et al. (2020) is one of our baselines and achieves better results on Mostafazadeh et al. (2016) than their approach.

structures and stories from the WritingPrompts test set. As automatic metrics are known to be overly coarse for open domain NLG (Novikova et al., 2017) we also report human evaluation results on 110 stories, with each story evaluated by 3 separate judges.¹¹ All metrics are averaged over all stories within the evaluation set.

Plot Structure Metrics. For plot structures we use purely automated metrics, as it is difficult for humans to read the abstract interim representation of SRL tags and entity anonymisation:

- Vocab:Token Ratio
- Entities per story

We choose to focus on these metrics as indicative of performance as they show the largest gap between gold and generated stories.

Automatic Story Metrics.

- Vocab:Token Ratio
- Unique Verbs & % Diverse Verbs: % of verbs not in the top 5 most frequent¹².
- Intra-story Trigram Repetition: trigram repetition *within* a story
- Inter-story Trigram Repetition: trigram repetition *between* stories

We again use Vocab:Token ratio as a rough diversity metric. We report also inter-story trigram repetition rates and intra-story trigram repetition rates (Yao et al., 2019).¹³

For all of these metrics, there is a tension between diversity metrics, which bring a generation closer to human quality, and fluency metrics, which can degrade as diversity increases.

Human Metrics. We run two separate experiments to measure improvement in our target areas of Relevance and Overall Quality. We postprocess stories for human review by detokenizing¹⁴, removing special end-of-sentence tokens, and truncating to 250 (whitespace separated) words. We have Mechanical Turk workers rate all systems’

¹¹Workers were paid \$12/hr and instructed to disregard punctuation and spelling. Surveys, as well as further details on compensation rates, are in Appendix C

¹²We report these for comparability to Fan et al. (2019); as they do we identify verbs via <https://spacy.io/>

¹³The former is a diversity metric - if stories look fine but inter-story repetition is high, it means the language model has learned to tell only very similar stories even when conditioned on diverse prompts. Intra-story trigram repetition is a fluency metric, and measures the proportion of trigrams within a single story that are repeated.

¹⁴We use MosesDetokenizer (Koehn et al., 2000)

System	Voc:Tok ratio	Entities	Avg Tok
Naive Plot	1.52	8.25	199
Aristotelian Plot	1.81	7.49	168
Gold Plot	3.59	9.26	371

Table 4: Metrics for plots

System	Automatic Evaluation			
	Voc: Tok↑	Diverse Verb %↑	Intra-Rep↓	Inter-Rep↓
Mao et al. (2019)	6.6	81.8	5.68	27.5
Guan et al. (2020)	2.3	71.9	0.60	56.1
Prompt-to-Story	1.5	68.9	0.22	65.1
Naive Plot	1.4	76.4	0.11	63.8
Aristotelian Plot	1.5	74.8	0.12	64.1

Table 5: Automated metrics for all models.

outputs on the same prompt comparatively on a *likert score* (1-5) across both metrics. But since likert scores are well known to exhibit a central tendency bias, it is likely to be unreliable on distinguishing between systems that are close in performance, particularly as reading 5 long generations introduces significant cognitive load. Therefore, we further conduct *pairwise comparisons* between the top 3 systems in the likert experiment.

Test Data Selection. In contrast to previous work on this dataset, our 110 human titles are randomly sampled from a *filtered version* of the test set. Writing Prompts has a one-to-many relationship between the prompts and stories.¹⁵ The dataset also contains an artifact of the sort of topic that is upvoted on reddit¹⁶, so many test prompts are minor variations on the same topic. We hypothesise that some of the gap between reported performance in papers on this dataset and performance in the wild is due to the artificially high similarity between training and test prompts, so we randomly sample from the test set, but exclude prompts with extremely high lexical overlap with training prompts.¹⁷

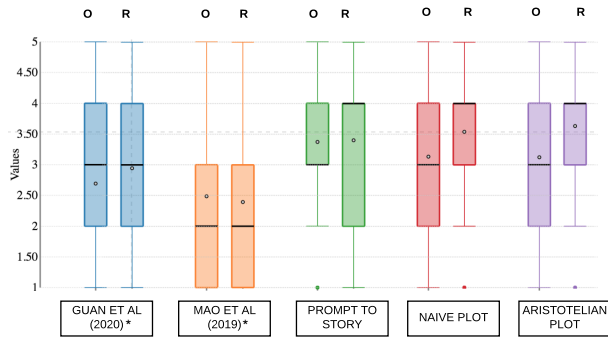
5 Results

Plots. As is shown in Table 4, the Aristotelian plot brings the generated plot structure closer to the gold plot Vocab:Token ratio, though there is still a large gap. This improvement comes at a slight expense of number of entities per plot, which is

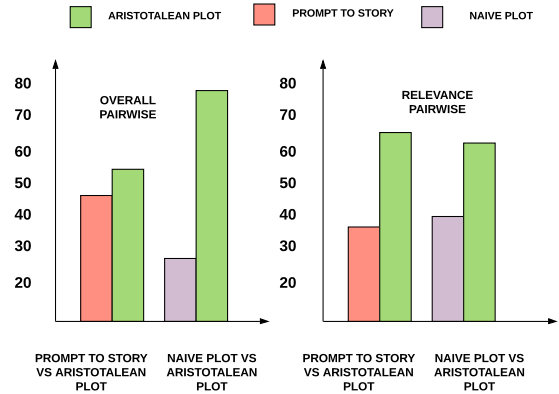
¹⁵Of the 303358 prompts, only 1/3 (107665) are unique

¹⁶Mostly aliens

¹⁷72% of prompts were excluded. We used sequence-matcher <https://docs.python.org/3/library/difflib.html> and spacy vector similarity <https://spacy.io/> to exclude prompts with a similarity of 1 to any prompt in the training data when stopwords are removed. The two systems gave identical results.



(a) Likert ratings for all 5 systems scored together. O is Overall, R is Relevance, solid bars are median, dots are mean. * denotes systems with significant differences ($\alpha < 0.05$, Wilcoxon signed-rank test). The three BART based systems at right are not statistically significantly distinguishable via this setup due to central tendency bias.



(b) Pairwise win rates for the Aristotelian Plot system and two best baselines. Significant ($\alpha < 0.05$) via Wilcoxon signed-rank test.

Figure 2: Human evaluation for five systems on 110 prompts, with each sample evaluated by 3 judges. The scores for the likert experiment are averaged over all the instances; the pairwise comparison takes the majority vote of the three judges for each instance.

likely because rescored plots are on average shorter.

Stories. The combination of automatic and human evaluation for stories highlights both the coarseness of automatic evaluation metrics and the blind spots of human evaluation (Hashimoto et al., 2019). From just the lexical metrics, the two baseline systems of Mao et al. (2019) and Guan et al. (2020) would appear to have the best performance (though Guan et al. (2020) suffers in Diverse Verbs), and the three BART-based systems are indistinguishable save an improvement in Diverse Verbs with the introduction of plot structures, as observed in the plot experiments of Fan et al. (2019).¹⁸ However, intra-story trigram repetition reveals the weakness of the Prompt to Story system and the Guan et al. (2020) system. It also shows the extreme lack of fluency in the Mao et al. (2019) system, which exhibits an unnatural level of repetition characteristic of degenerate output.

Human Evaluation. The human metrics in Figure 2a then reinforce that the high lexical scores of the Guan et al. (2020) and Mao et al. (2019) systems are a result of those models sometimes deteriorating into nonsensical text, as both of those systems have extremely low human judgement

scores.¹⁹ The likert scores favor the Aristotelian Plot system with regard to relevance but favor the Prompt to Story baseline for Overall Quality - though as can be seen from the variance in 2a, all three BART systems are too close together to be reliably distinguished via likert metrics. The pairwise comparisons in Figure 2b do differentiate the three systems with strong statistical significance, showing the superiority of Aristotelian Plot over both the Prompt to Story and Naive Plot systems with respect to both Relevance and the Overall Quality of the final stories.

6 Analysis

We analysed the patterns in reported user confidence and found that 8% of prompts are low user confidence (<3) and 8% are high confidence (>4.5), so we look further into these as examples where the top three systems are minimally and maximally distinguishable. We include examples of outputs for these prompts for all three models in Appendix A.3.

For Overall Quality, there are no low confidence titles, and the Aristotelian plot system is preferred for all of the high confidence ones. In Relevance, the low confidence prompts have split win-rates across all models (essentially random) and the prompts show that all systems struggle with the meta-level concepts. The Writing Prompts dataset varies from concrete (*A story of a cat who saves*

¹⁸Though we do observe a similar magnitude increase in Diverse Verbs with the introduction of plots, our baseline Prompt-to-Story model has a higher % than their best model, reflecting the recent performance improvements in pretrained language models. We report Unique Verbs for comparability, but do not find it to be useful metric since it is not normalised by length or token count.

¹⁹Examples from each baseline system for the prompt in Table 1 can be found in Appendix B

System	Word Incorp %	Verb Incorp %
Naive Plot	75.5	75.1
Aristotelian Plot	72.4	72.0

Table 6: Metrics for incorporation rates for stories.

a girl from her depression) to prompts requiring other types of meta-knowledge. Prototypical low confidence titles are of two forms: 1) *A day in this life in a world where everything is written in poetry*, and 2) *Write a story where I can't tell whether the protagonist is a hero or a villain*. The type 1 stylistic instruction prompt requires knowledge about the distinction between instructions for style and content, and all models fail. The type 2 meta-content type of prompt requires a finer level of control of the structure of the plot and story than the Aristotelian system or any of the other models can manage. This kind of prompt presents an interesting case for future story generation work, as it is simple enough for human authors to be popular on a forum, but far beyond the capabilities of current language models. At the high confidence Relevance prompts, the Aristotelian system wins all but one. Those stories highlight the way that adding and then improving on plot structures assists relevance by keeping a story on topic throughout an entire generation (see Appendix A.3 Table 8).

To assess the cases where the Aristotelian Plot did not improve over the BART baselines, we measure both word and verb incorporation rates, in Table 6²⁰. These measure the Levenshtein distance over the sequence of plot words (excluding verbs) or plot verbs that are included within the final story. While the high incorporation rates show that the story model does utilize the plot, there is a gap between the current utilisation and the possible upper bound. The focus of our work is on improving plot generation, but we hypothesise that modifications to the story model to improve incorporation rates would further widen that performance gap between the three systems, as it will give the plot more influence over the story surface realisation and ensure that plot improvements appear downstream.

²⁰Yao et al. (2019) also use word incorporation rates, but theirs are not comparable to ours, as we both include verbs as their own separate metric (due to their importance in our structure) and make this an *ordered* metric (rather than set intersection as they do), which is necessary because our generated stories are much longer

7 Related Work

Story Generation without Plots. Diverse efforts have focused on generating stories. Fan et al. (2018) re-purpose an approach for Neural Machine Translation to translate from prompt to a story via Convolutional Seq2Seq models. Guan et al. (2020); Mao et al. (2019) use a similar approach, however they incorporate structured commonsense knowledge from external datasets or knowledge bases to improve a story generated from a prompt.

Story Generation with Plots. Riedl and Young (2010) use refinement search as a technique to balance between character and plot for solving the narrative generation problem. Li et al. (2013) use plot graphs for story generation that model the intended logical flow of events in the virtual world as a set of precedence constraints between plot events. Martin et al. (2018) decompose the problem of story generation into generation of successive events (event2event) followed by generation of natural language sentences from events (event2sentence). Yao et al. (2019) improve their LSTM-generated ROC-Stories by extracting keyword-based plot-like structures, or *storylines*, and using these in a pipeline to generate storylines and then stories. Fan et al. (2019) experiment with numerous techniques for representing story plots on the WritingPrompts dataset, and find Semantic Role Labelling with Entity Anonymization and Compression to work best. More recently Chen et al. (2019) propose a latent variable model to learn how to generate outlines for neural story generation.

Learning Story Aspects via Rescoring. Holtzman et al. (2018) generate continuation sentences from context sentences, and introduce using a mix of collaborative discriminators that each learn one Gricean maxim of conversation, and use them to rerank their RNN story model output. Goldfarb-Tarrant et al. (2019) use those discriminators with the system of Yao et al. (2019) as part of a collaborative story writing task with an LSTM and human writers. However, none of them apply rescorsers to the plot. There is no other work that uses discriminators or rescorsers for plot structures or to try to train them based on different principles.

8 Conclusion and Future Work

We have shown that Content Planning via an interim *plot structure* representation can be combined with the use of rescoring models to inject Aristotelian story-writing principles into the plot struc-

ture. We found that this results in generated stories that are both more relevant and higher quality than stories that are generated directly from prompts or that use plots without Aristotelian rescoring. Our findings also suggest future work on additional ways to incorporate story principles into plot generation. Although our Aristotelian plots improved over the naive plot, there remains gaps in quality between generated and gold plot structures. There is also further work to be done in investigating what story models are best able to utilise improved plots, which would enable plot improvements to be even more effective.

Acknowledgments

We thank the anonymous reviewers for their feedback, and Adam Lopez for his excellent editing suggestions. This work is supported by Contract W911NF-15-1-0543 with the US Defense Advanced Research Projects Agency (DARPA).

References

- Poetics Aristotle. The loeb classical library, eds. te page, at al., and trans. w. hamilton fyfe.
- Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *ACL*.
- Gang Chen, Yang Liu, Huanbo Luan, Meng Zhang, Qun Liu, and Maosong Sun. 2019. Learning to predict explainable plots for neural story generation. *arXiv preprint arXiv:1912.02395*.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *ACL*.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for structuring story generation. In *ACL*.
- Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*.
- Seraphina Goldfarb-Tarrant, Haining Feng, and Nanyun Peng. 2019. Plan, write, and revise: an interactive system for open-domain story generation. In *NAACL-HLT*.
- Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A knowledge-enhanced pre-training model for commonsense story generation. *arXiv preprint arXiv:2001.05139*.
- Jian Guan, Yansen Wang, and Minlie Huang. 2019. Story ending generation with incremental encoding and commonsense knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6473–6480.
- Tatsunori B Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. *arXiv preprint arXiv:1904.02792*.
- Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and whats next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483.
- Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *Proceedings of ACL*.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 200. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.
- Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ArXiv*, abs/1910.13461.
- Boyang Li, Stephen Lee-Urban, George Johnston, and Mark Riedl. 2013. Story generation with crowd-sourced plot graphs. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Huanru Henry Mao, Bodhisattwa Prasad Majumder, Julian McAuley, and Garrison W Cottrell. 2019. Improving neural story generation by targeted common sense grounding. *arXiv preprint arXiv:1908.09451*.
- Lara J Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark O Riedl. 2018. Event representations for automated story generation with deep neural nets. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

- Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of NAACL-HLT*, pages 839–849.
- Jekaterina Novikova, Ondrej Duek, Amanda Cercas Curry, and Verena Rieser. 2017. [Why we need new evaluation metrics for nlg](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.
- Mark O Riedl and Robert Michael Young. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:217–268.
- Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D Manning. 2019. Do massively pretrained language models make better storytellers? *arXiv preprint arXiv:1909.10705*.
- Lili Yao, Nanyun Peng, Weischedel Ralph, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the thirty-third AAAI Conference on Artificial Intelligence (AAAI)*.

A Appendices

A.1 Data Quality in Mechanical Turk Studies

We require that Turkers doing pairwise story comparison report their confidence in their decisions and are clear that this makes no difference in their remuneration, and then we use patterns of confidence scores to find areas where models are very distinguishable and areas where they are not.

We additionally create manual True-False questions for each prompt used in human evaluation to be used as an attention check. We experimentally found that extractive attention checks (*this keyword is in the title*) were ineffective, but manually created simple ones such as *The protagonist is a child* were very effective at filtering out poor quality responses. 15% of all responses failed attention checks, and we excluded that data and reran those human evaluations until we had a full dataset. We did additionally verify that none of our questions were overly difficult by manually reviewing all attention checks that $< 80\%$ of respondents passed.

A.2 Dataset Statistics

The Writing Prompts dataset contains 303358 pairs of prompts and stories. Stories are already tokenized in the available dataset (Fan et al., 2018). Like (Fan et al., 2018) do we truncate to a maximum length of 1000 words per story (with an average of 700 words) and replace words that appear less than 10 times with an UNK token.

We split the dataset into Train/Validation/Test splits of 65/10/10 for fine-tuning Language Models, with the remaining 15% split into 10 for fine-tuning Rescoring models, and 5 for training Mixture Weights and generally running metrics and tests.

A.3 Example Outputs For High and Low Confidence Scores

All examples in Tables 7, 8, and 9 are unedited save removal of special separation tokens and detokenization. Line breaks are added by the model.

A.4 Implementation Details

We fine-tune BART with same hyperparameters mentioned in ²¹ with the exception of MAX-TOKENS (size of each mini-batch, in terms of the

²¹<https://github.com/pytorch/fairseq/blob/master/examples/bart/README.summarization.md>

Prompt: Earth has gained a new season, unlike anything currently known

Aristotelian Plot System Generated Story:

The Earth is more beautiful than it thought! It's the most beautiful planet. The sun was shining bright, and the air was crisp, yet the air itself was cold. Everything was so bright. It was the perfect day for earth to be beautiful. Everyone on the planet Earth had lived a long and happy life, this planet has been home to the human race for thousands of years. The humans were a species that had been able to survive through Earth's harsh environment. the humans could survive in these harsh conditions. But one day Earth had a new life, and we were the first to notice it. In the middle of the night, we could see something, a small, dark cloud. A small, white cloud, it's size was almost invisible. Then that cloud grew. And it grew... The sun, shining bright.. There was no sun. All the light was black and it covered the sky. Every day the sky was a dark blue. No matter the time of day, it never seemed like the perfect time, it was a beautiful day. For a new season to begin. Yet no one could know why. Why did we not have a new cycle?"

Naive Plot System Generated Story:

"It's not a new day, John. "

"What? What is it? You ca n't see?"

The man was silent.

He said, "It'd be the first time we've all seen its as we know it, but it's the first we're seen in over a million years. It is a new world, one with no sun. "

A few seconds of silence. "Its not a New Day." the man said.

"It're a new season," said the woman. We've seen them all. I mean, we all know by now. You know, I've- seen, you've always seen- "

"We'll see us. But you'll be the First, you were the first one to see, we'll all be the Last, we will all be, we have all been the first of our species. There is no one more, no one is better than me." "The woman was right.

Prompt to Story Generated Story:

I was in my bed, in my own room, in the middle of a forest, when the world stopped.

It was the first time in my life that I had felt I had a sense of peace. My family and me, myself, were gone, we had all been taken from us. The only thing our family had left were the three children and my world, but my wife and I were the only survivors. There were we, the three of us. But ourselves, our children were all gone. And the only reason why me and our family survived was because ours had become ourselves. It was as if both of us were one with the world. As all the children were gone and the adults had left myself to fend for myself, you and your family. No longer were you and I to be each other. Now we were just two. All our memories of us, all ours. Nothingness. Only all of us, and my children, each of us...

Table 7: High Confidence: Overall Quality (score 4.67 out of 5)

Prompt: Evolution has gone a little differently. Humans have poison and some way to transmit it.

Aristotelian Plot System Generated Story:

So the poison is in your genome," said the scientist.

"I have n't even it's looked at you yet Mr. Doctor!"

"Yes sir!"

"And its in my DNA," said the man.

"Well, you must be the first human, to have developed this. I mean, you are." He said, "It's not a question of if but when You. It'd be a question if your DNA would even be the same."

The man looked him in the eye. "Well Mr. Doctor..."

"Yeah you are the first, I mean sir," he said. * I am not sure he was. But he is who i am. Do you see that? You and I are not the first species to have it, but this one has come from the beginning of this.". The human genome of the virus had the most advanced species known to have had It. And it was spread by a species that had the best genetic makeup in the world. A species which was a virus spread by the human race. Now, you and I were not the best of the best at it, because we did n't have the ability to develop a species who could spread the toxin. We were the best. This is why we live in this place now. Because we can live in a world with no poisons. You know we've killed off each other, but now we live our lives, you all know. Just like myself, Mr. President, I do n't know how long it took. They live by. Well, it was about a thousand years ago. In the same world we come from, the human genome spread by The human race, and spread by that same process.

Naive Plot System Generated Story:

"What? People say that they know what they're saying. I know, but the ones who kill? I know. The ones who do the killing!"

"Yeah I know! They're all dead! All of them, the ones you know about! That's what you say... but what? That we're gon na kill them? You see, the one who's dead, are the ones that you've killed. Not me. It's us." They are my brothers and sisters, after all.

The one's that will die, the people who we kill, the guys and girls that you guys and boys kill for.

Prompt to Story Generated Story:

We are the most dangerous species in history, yet, we do not know why. We have been hunted for thousands of years. It was the same for our ancestors, they were their prey, and the same fate for it.

Our species evolved to be more intelligent than the others, and so they were able to survive. But the human race was n't able to adapt. So they were forced to adapt to the world themselves, and the first few years were difficult. There were no predators to eat, no predators that could eat us, no prey that could be killed, no one to protect them. All the predators had been wiped out and their prey was no longer needed. The population was so small, so few that no predator could take on the entire world. And so they died. No one was left to fight. Just the humans. That is until one of them, an old man named the man who was the most intelligent and powerful of all of them... A few of them were his friends. Some were his wife, others his kids, and some even he was a friend. Then the humans came to their deaths, they were not as strong as them."

"And we were the last to go." "The man continued."

Table 8: High Confidence: Relevance (4.86 out of 5)

Prompt: Write a story where I can't tell whether the protagonist is a hero or a villain.

Aristotelian Plot System Generated Story:

"The men seemed a little more than usual, a bit. The women were n't the same. No.

They were all different. All of them, my wife. Each one of them were given the love of a woman, and none of them knew me as I did. If it took the people of the city to realize what they had done with the world, the world would n't have been so different...

But this man. He was my hero. The one who was the one who made the difference, the only thing what had kept the women from the other women. The one who had saved the day. And the woman was the most special. It was the woman, the one that he had loved. But who would take the fall for the save a single life, for a person who saved his wife, her life. Not a single person was going to be able to save from this. There was no one to save. Nobody. Even the police. That seemed like it to us. "No one cared.". In a world where there were no people who needed help, no people that could need help. A world where those who needed for others, were the ones who would be the ones to do the work.

Naive Plot System Generated Story:

So he's the hero. He's a good guy. Or is it?" "We did n't know him before. But we come in here for a meeting. What's his name?"

"We'll have a talk. Do n't go anywhere, my name is n't your name."

* His name is not mine, I'm not sure its a real name. * I'M not sure if I've ever been told I was a hero, or just a villain.

"No, no, you're not. I'm not."

"What does that even mean?" "The man ask me. "What is your name? Is yours name really mine?"...

The mayor of this small town is on the phone with his secretary. We've got a new guy." the woman says. A young man with a black suit and a dark suit. In the suit, he was told she was a heroine. No.

Prompt to Story Generated Story:

I was a hero.

I have never seen a hero before, and me being one is something you do n't expect in your first year of high school.

The first thing that happened to my mind was that i was a kid, and then its a little hard to explain. The first few weeks of school were a blur. A few months went by and then it was just myself and my best friend. Then the bullying started. And then she started to do her own shit. So then it was mine, and soon it was all she did. But he did n't know what to do either. It was n't until his first few grades that it was nt until he started to talk. When we were in middle school that was when the bully started to take things out on him. That was when it all started...

It took us about a year before our teacher finally got the courage to tell the class to call in sick to school.. To go to the hospital.

Table 9: Low Confidence: Relevance (2.8 out of 5)

number of tokens.) being 1024 for us. For decoding we generate plots and stories from our models using a top-k random sampling scheme (Fan et al., 2018). At each timestep, the model generates the probability of each word in the vocabulary being the likely next word. We randomly sample from the $k = 5$ most likely candidates from this distribution. We also use a softmax temperature of 0.7.

A.4.1 Fine-Tuning BART for Prompt to Plot

1. **No of Parameters:** For BART we use the BART large checkpoint (400M parameters) and use the implementation by FAIRSEQ (Ott et al., 2019) ²²
2. **No of Epochs:** We fine-tune pre-trained BART for 19 epochs for FTB model keeping all default hyper-parameters as mentioned in <https://github.com/pytorch/fairseq/blob/master/examples/bart/README.summarization.md> with the exception of $MAX_TOKENS = 1024$ and $UPDATE_FREQ = 16$.
3. **Training Time:** Our training time is 32 hours
4. **Hardware Configuration:** We use 2 RTX 2080 GPU (11GB)

A.4.2 Fine-Tuning BART for Plot to Story

1. **No of Parameters:** For BART we use the BART large checkpoint (400M parameters) and use the implementation by FAIRSEQ (Ott et al., 2019) ²³
2. **No of Epochs:** We fine-tune pre-trained BART for 17 epochs for FTB model keeping all default hyper-parameters as mentioned in <https://github.com/pytorch/fairseq/blob/master/examples/bart/README.summarization.md> with the exception of $MAX_TOKENS = 1024$ and $UPDATE_FREQ = 16$.
3. **Training Time:** Our training time is 6 days due to limited GPU RAM and since stories and plots are large sequence of tokens.
4. **Hardware Configuration:** We use 1 RTX 2080 GPU (11GB)

²²<https://github.com/pytorch/fairseq/tree/master/examples/bart>

²³<https://github.com/pytorch/fairseq/tree/master/examples/bart>

A.4.3 Fine-Tuning roBERTa-large for Discriminators

1. **No of Parameters:** For BART we use the roBERTa large checkpoint (355 parameters) and use the implementation by FAIRSEQ (Ott et al., 2019) ²⁴
2. **No of Epochs:** We fine-tune pre-trained roBERTa 7 epochs for relevance discriminator model, 5 epochs for entity discriminator models, 4 epochs for inter-event, 2 epochs for intra-event and 10 epochs for intraV-event discriminator keeping all default hyper-parameters as mentioned in <https://github.com/pytorch/fairseq/blob/master/examples/roberta/README.glue.md> with the exception of $max_tokens = 1024$ and $lr_scheduler = fixed$.
3. **Training Time:** Our training time is 10 hours for relevance discriminator, 4 hours for entity discriminator, 4 hours for inter-event, for 2 hours for intra-event 12 hours for intraV-event discriminator
4. **Hardware Configuration:** We use 4 RTX 2080 GPU (11GB)

A.5 Hyper-Parameters for Mixture Weight Tuning

Mixture weights are tuned with a held out validation set of 10,000 samples. The models train for 3 epochs, but all converge in 1 epoch, which takes 24 hours on 1 RTX 2080 GPU (11GB). We use SGD at each step, with learning rates set to 0.001.

A.6 SRL-Event Representations

We compress event-representations by removing all verb arguments beyond args 0,1,2.

We exclude the following verbs when extracting plot events: "is", "was", "were", "are", "be", "ś", "ře", "íl", "can", "could", "must", "may", "have to", "has to", "had to", "will", "would", "has", "have", "had", "do", "does", "did"

B Output from Baseline systems for the first example prompt

Output from the remaining models for the example prompt in Table 1 of the paper: *Write the most*

²⁴<https://github.com/pytorch/fairseq/blob/master/examples/roberta/>

Prompt: Write the most beautiful end to the universe that you can imagine

Prompt to Story Generated Story:

We were n't sure how long it would last. It was only a matter of time before something would happen and the universe would be gone.

It had been a long time since the end had come. A long time ago they had been trying to make it happen for us but we were nt sure how its going to be , or how they would do that. But now its over. Its time. The end was here and I am the last one standing. And now you will know your end. This is my last post

Guan et al. (2019) Generated Story:

The wind blotted it's path through the clouds, the wind whipping through the trees, the sky. It was a beautiful day. - The wind cut through his hair and made him stop, a tear rolling down he'd always dreamed about. "That day. ". ity. a dark shadow stretched out and reached down into the man's pocket. h He took a deep breath, and looked at the wind. a small flash. i" Hey! '. 'hanging up her coat she said, walking up the street. d her legs made a sound as they crossed a busy sidewalk. b At this point, the wind took out its last breath. u'! hashing the door! 'y. y it made a loud noise and shouted at Katerina. y The noise stopped when it started to move again. z "Oi.'... 'I'm sorry, I was just walking up. "She stood up, and took out a small pack of cigarettes. m It kept you company. o Katerinas started to walk towards Keiran. t, Mina's voice was coming down. th! y! t Th As Tchaikovsky started to play the piano, the wind stopped, its face fell to the ground. p * * "What in the hell are you doing.' *. The wind stopped again.. * Keirans eyes followed. 1.. y. Mina was sitting on the bench, Keirany had a smirk on her face.

Mao et al. (2019) Generated Story:

The Mysterious Crawl -----
----- Part 1: Part 2: Part 3: Part 4: It was an exercise in Mars shoubu!!!; where one record holder just spins out, another record holder flies in and out of a hole in a water is a caterpillar. It's like the practice. Pre-secession: On one side, watches "forever"end in its own universe, one question is raised: The term" end of the universe," or the likely future, or the child of the universe, may be brought up and translated as "no matter what is said, it is true. "

Table 10: Output from Baseline systems for the first example prompt

beautiful end to the universe that you can imagine.

All examples are unedited save removal of special separation tokens and detokenization.

C Mechanical Turk Study Materials & Compensation Details

Human Evaluation consisted of 110 titles evaluated by 3 judges each, which results in 330 x 5 long stories for likert comparisons and 1320 x 2 long stories for pairwise comparisons (as there are 4 sets of pairwise comparisons). Turkers received \$12/hr, so the entire budget was \$2k.