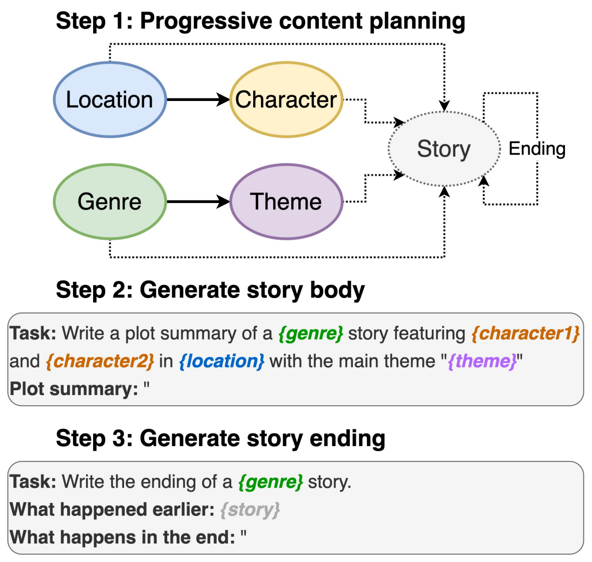
Plot Writing From ~~*Scratch*~~ Pre-Trained Language Models

Anonymous ACL submission

Abstract

Pre-trained language models (PLMs) fail to generate long-form narrative text because they do not consider global structure. As a result, the generated texts are often incohesive, repetitive, or lack content. Recent work in story generation reintroduced explicit content planning in the form of prompts, keywords, or semantic frames. Trained on large parallel corpora, these models can generate more logical event sequences and thus more contentful stories. However, these intermediate representations are often not in natural language and cannot be utilized by PLMs without fine-tuning. We propose generating story plots using *off-the-shelf* PLMs while maintaining the benefit of content planning to generate cohesive and contentful stories. Our proposed method, ScratchPlot, first prompts a PLM to compose a content plan. Then, we generate the story's body and ending conditioned on the content plan. Furthermore, we take a generate-and-rank approach by using additional PLMs to rank the generated (story, ending) pairs. We benchmark our method with various baselines and achieved superior results in both human and automatic evaluation [1].


Step 1: Progressive content planning
Location
Character
Genre
Theme
Story
Ending
Step 2: Generate story body

Task: Write a plot summary of a ***{genre}*** story featuring ***{character1}*** and ***{character2}*** in ***{location}*** with the main theme "***{theme}***"
Plot summary: "

Step 3: Generate story ending

Task: Write the ending of a ***{genre}*** story.
What happened earlier: *{story}*
What happens in the end: "

Figure 1: Overview of ScratchPlot. We factorize the elements of a story into four attributes {location, characters, genre, and theme}. We first prompt a PLM to compose them sequentially, then generate the story conditioned on these attributes. When writing the ending of the story, the model additionally conditions on the previously generated story.

1 Introduction

Long-form story generation is challenging because language models lack global planning (Hua and Wang, 2020; Tan et al., 2021), discourse coherence (Bosselut et al., 2018; Ji and Huang, 2021), and common sense knowledge (Xu et al., 2020; Ji et al., 2020). While individual sentences appear fluent and logical, they do not fit together as a whole and the stories often have no clear content (See et al., 2019; Goldfarb-Tarrant et al., 2020).

Interestingly, recent work in long-form story generation relied on *explicit* content planning (Reiter and Dale, 1997), contrary to the prevalent trend of end-to-end learning across NLP tasks. The content plan usually takes the form of prompts (Fan et al., 2018), keywords/keyphrases (Xu et al., 2018; Yao et al., 2019), semantic frames (Fan et al., 2019), or summaries (Sun et al., 2020).

These content plans are usually not in the form of natural language [2] and cannot be understood by pre-trained language models (PLMs) without fine-tuning using parallel data. Another subtle problem of modeling story generation as a supervised learning task is that the model learns common sense and frequently occurring action sequences, like morning routines (Fan et al., 2019). Such an action plan may not be interesting and surprising, which are crucial characteristics of stories.

We propose generating stories using off-the-

[1] Code and data available at `https://github.com/anonymized-for-review`.

[2] Except for using summaries as the content plan.

shelf PLMs without fine-tuning. We tap on DINO (Schick and Schütze, 2021), a framework to generate datasets using instructions, to compose stories progressively. Our method, SCRATCHPLOT, is depicted in Figure 1. We firstly prompt a PLM to perform content planning, including the location, characters, genre, and theme. We then generate a story conditioned on these attributes. Finally, we generate story endings and rank them.

2 Related Work

Fan et al. (2018) introduced hierarchical story generation by first generating a prompt, then transforming it into a full story. They introduced a novel fusion-based architecture to improve the relevance between the generation and the input prompt. Subsequent works used sequences of keywords (Yao et al., 2019; Rashkin et al., 2020), semantic frames (Fan et al., 2019), or paragraph summaries (Sun et al., 2020) as the content plan because they have a higher capacity than a single prompt. See et al. (2019) compared Fan et al. (2018) with a fine-tuned GPT-2 model (Radford et al., 2019) and showed that GPT-2 writes better stories and conditions on the prompt more strongly.

Recently, Tan et al. (2021) proposed ProGen, a multi-stage seq2seq model, by extracting keywords at different granularities. Each stage takes the output from the previous stage and adds finer-grained details. Unlike previous works, we use heterogenous plot elements sampled from a PLM as the content plan. We also do not require any fine-tuning and rely solely on off-the-shelf PLMs.

3 Plot Generation From Scratch

DINO (Schick and Schütze, 2021) is a framework to generate labeled NLI datasets (Bowman et al., 2015) using a pre-trained GPT-2 model (Radford et al., 2019). Schick and Schütze (2021) formulated different task descriptions to generate sentence pairs for each category. Instead of generating the sentence pairs at once, they sample the first sentence x_1, then incorporate it into the task description to sample the second sentence, as shown in Figure 2.

Task: Write two sentences that mean the same thing.
Sentence 1: "x_1"
Sentence 2: "

Figure 2: Task description to generate a similar sentence by incorporating the first generated sentence, x_1.

Factorizing Elements of a Story We factorize story generation into multiple stages analogous to generating NLI sentence pairs. We define four main plot elements: location, characters, genre, and theme. These elements are not entirely independent. For example, the genre will influence the theme. We denote these dependencies using solid arrows in Figure 1. We then use different task descriptions to sample these elements sequentially. Figure 3 shows example task descriptions to generate the genre and theme. We use paraphrases of the task descriptions, and the complete list is presented in Appendix A.

Task: Write a story genre.
Story genre: "

`> action`

Task: Write the twist in a <x_1> story.
Twist: "

`> In the early morning hours on a warm summer's morning, a boy came home and found the family locked in a room.`

Figure 3: Task description for generating genre and theme. <X1> denotes the generated genre. The example continuations are generated by GPT2-XL.

We sample one value for each plot element except for characters, where we generate a male and a female character. After sampling all plot elements, we fuse them into a single task description to generate the story, as depicted in Step 2 of Figure 1.

Generating Coherent Story Ending A coherent and thoughtful ending is crucial to stories. However, it is not obvious how to write the story ending with PLMs. One challenge is that GPT-2 does not have an <EOS> (end-of-sentence) token. Schick and Schütze (2021) always end the task description with an open quotation mark and treat the first quotation mark generated by the PLM as the <EOS> token. However, the PLM usually generates the quotation mark after a couple of sentences, making it unsuitable for generating long-form stories. Therefore, we ignore the artificial <EOS> token and generate the story with a fixed length. Then, we truncate it till the last complete sentence.

We design a separate task description for story ending generation by providing the story body and asking the PLM to write what happens in the end. As the story ending is usually short, we treat the first quotation mark as the <EOS> token.

We observe that PLMs sometimes ignore the task descriptions and write generic or irrelevant story endings. Therefore, we propose two methods to rank the story endings. Firstly, we use the next sentence prediction (NSP) task of BERT (Devlin et al., 2019) to measure the coherence between the story and the ending. Specifically, we calculate $\mathcal{P}_{NSP}(b, e)$, where b denotes the story body and e denotes the story ending.

Inspired by previous works in fact-checking with PLMs (Lee et al., 2020, 2021), we use the perplexity score as another metric for the story ending's quality. Specifically, we concatenate the story body and ending to form the input to the PLM: $X = \{x_{b_0}, ..., x_{b_B}, x_{e_0}, ..., x_{e_E}\}$, where B and E denote the number of tokens in the story body and ending separately. We then calculate the conditioned perplexity by

$$PPL(X) = \sqrt[E]{\prod_{i=1}^{E} \frac{1}{p(x_{e_i}|x_{b_0}, ..., x_{b_B}, ..., x_{e_{i-1}})}}$$

Note that we use the story body tokens to condition the perplexity, but they do not contribute to the $PPL(X)$.

We hypothesize that NSP and PPL attend to different aspects. NSP performs sentence-level classification and may pay attention to discourse markers or lexical overlap to determine whether a sentence is a valid continuation of another sentence. On the other hand, PPL aggregates scores for individual tokens and measures the intrinsic quality of the story ending.

We sample multiple story bodies and their corresponding endings and rank them using NSP and PPL separately [3]. We select the story body/ending pair with the lowest mean rank as the final output.

4 Experiments

Experimental Details We use the official implementation of DINO (Schick and Schütze, 2021) [4] with the default GPT2-XL language model. We follow the default parameters except setting k=30 for top-k sampling and blocking repeating trigrams during generation. We use self-debiasing (Schick et al., 2021) to differentiate different geographical units and male/female names. The rest of the generations use a vanilla PLM without self-debiasing. For story ending ranking, we use HuggingFace (Wolf et al., 2020) `bert-base-uncased` checkpoint to calculate the NSP probabilities and `gpt2` (base) to calculate the perplexity.

We perform simple post-processing to clean or filter the continuations, such as removing tailing punctuations and filtering continuations that repeat words from the prompt or contain 1st or 2nd person pronouns. The story body must also contain some plot elements to ensure it is contentful and respects the task description. The post-processing is detailed in Appendix B.

Baselines We compare with two conditional story generation baselines. The first one is a Fusion model with a convolutional encoder and a self-attention decoder (Fan et al., 2018) [5]. The second model is ProGen (Tan et al., 2021) [6], a multi-stage seq2seq model using salient keywords as intermediate representations. We provide the same *generated* content plans to the baselines to make the comparison fair. We use the generated *theme* as input to the Fusion model, which is analogous to the prompt. On the other hand, we extract keywords using TF-IDF following Tan et al. (2021) from all plot elements to prepare the input to ProGen. We also experiment with a baseline GPT2-XL without content planning where we sample a list of stories by providing the instruction "**Task:** Write a plot summary.\n **Plot summary:**". We limit the story length to 150 tokens in all models for ease of human evaluation [7].

RQ1: How does SCRATCHPLOT compare to the baselines? We generate 50 stories using each model and invite three crowdworkers to evaluate each story on the following fine-grained aspects: naturalness, interestingness, and cohesiveness. We take the average of the scores assigned by the annotators as the final score. Appendix E provides full details of the crowdsource evaluation.

Table 1 overviews the result. SCRATCHPLOT outperformed all baselines by a large margin. We notice that the Fusion model tends to generate common narratives, thus receiving the lowest interestingness score. ProGen sometimes generates ungrammatical text, likely due to its intermediate content representation, which is a list of keywords.

[3] NSP the higher, the better. $PPL(X)$ the lower, the better.
[4] https://github.com/timoschick/dino

[5] https://github.com/pytorch/fairseq/tree/main/examples/stories
[6] https://github.com/tanyuqian/progressive-generation
[7] Additional experimental details can be found in Appendix C.

Model	**natur**	**inter**	**cohes**
Fusion	2.13	2.31	1.89
ProGen	2.13	3.05	1.88
ScratchPlot	**4.02**	**4.17***	**3.99***
- content plan	3.64	3.19	3.41

Table 1: Human evaluation result of various models on different aspects. The columns denote naturalness, interestingness, cohesiveness. All scores are on a scale from 1 (worst) to 5 (best). Best scores for each aspect are highlighted in bold. * indicates statistical significance compared with the second best system using two-sided paired t-test with p=0.01.

Table 9 shows a randomly sampled content plan with stories generated by each model.

RQ2: Does unsupervised content planning help? Different from previous work in story content planning, ScratchPlot is entirely unsupervised, relying only on task descriptions. Table 2 presents the average expert rating for each generated plot element [8]. As we can see, the PLM generates simple fields like locations and person names with high quality. However, some generated themes are ambiguous or nonsensical.

Element	Location	Cast	Genre	Theme
Score	0.930	0.931	0.792	0.654

Table 2: Expert rated scores for generated plot elements normalized to the range of 0 (worst) to 1 (perfect).

ScratchPlot outperformed the baseline without content planning in all aspects, demonstrating the contribution of content plans in story generation even when they are imperfect.

Furthermore, we measure *intra*-story lexical diversity using self-BLEU (Zhu et al., 2018) and *within*-story lexical diversity (or repetition) using *distinct-n* (Li et al., 2016) and summarize the result in Table 3. Unsurprisingly, the baseline without explicit content planning generates less diverse stories because they are sampled by conditioning on the same instruction. It also generates more *within*-story repetitions than ScratchPlot.

RQ3: Which story ending ranking performs best? We compare our story ending ranking with three alternatives, selecting the best NSP score, the best PPL score, and a random story body and ending pair. We randomly sample 50 content plans

[8]Detailed evaluation on the content plan quality can be found in Appendix D.

Model	self-BLEU		distinct-n	
	n=1	n=2	n=1	n=2
ScratchPlot	**.763**	**.329**	**.249**	**.728**
- content plan	.799	.380	.204	.650

Table 3: Result of self-BLEU scores to measure intra-story diversity (the lower the better) and distinct-n scores to measure repetitions (the higher the better). The best results are highlighted in bold.

where each method selects a different story ending [9] and use them to conduct a pair-wise crowd-source evaluation. Each time, we present the annotators two stories generated using the same content plan, one of which uses our story ending ranking (with randomized order). We highlight the story endings for ease of comparison and ask them to rate which story ends better. We take the majority vote from three annotators for each comparison and present the result in Table 4.

Method	**Win**	**Lose**
NSP	**36***	14
PPL	23	**27**
Random	**36***	14

Table 4: Pair-wise comparison of the story ending quality. "Win" indicates our method is rated better than the alternative and vice versa for "Lose". * indicates statistical significance using two-sided Wilcoxon signed-rank test with p=0.01.

Using NSP to select story endings does not perform better than random when compared with our proposed method. Due to the unreliability of NSP scoring, using the mean rank also performs worse than relying on the perplexity alone. However, the difference is not statistically significant. Based on this empirical result, it seems perplexity is a robust metric to select appropriate story endings.

5 Conclusion

We introduced ScratchPlot, a framework to perform unsupervised content planning for story generation using only pretrained language models (PLM). ScratchPlot achieved strong results compared to supervised baselines fine-tuned on large parallel corpora and a PLM without access to content plans. In future work, we plan to generalize the framework to other types of long-form text.

[9]They may or may not have the same story body.

Ethical Considerations

Our proposed method is intended for creative text composition. The generated stories can be either consumed by readers or help writers to come up with new ideas. There are several potential risks if the proposed method is not deployed with care. However, they are inherent from large pre-trained language models (PLMs) instead of intrinsic to our method.

First, PLMs may recall partially from the training data instead of composing stories from scratch. Due to the vast size of the pre-training data, it is not feasible to measure what percentage of the generated stories are "original". Secondly, the system sometimes generates real person names of famous people as the main characters. It should be noted that the system is for literature purposes and is not meant to be a factual report of real persons or anecdotes. Lastly, the system might generate inappropriate or disrespectful stories to a particular population, such as the genres "biblical epic" and "erotica". Manual curation or automatic content filtering can be deployed to mitigate this problem.

We relied on crowdworkers to conduct human evaluations in this work. The crowdworkers are from various countries, and the adequate payment differs drastically. Therefore, We target paying $6.0 per hour. Some of the tasks took longer than we initially estimated, and we issued all crowdworkers a one-time bonus of $0.2 to compensate.

Although we use a relatively large PLM (GPT2-XL; 1.5 billion parameters), our approach does not require training. Generating a single story takes around 1 minute, consuming 0.003 kWh power based on the max power consumption of the Quadro P5000 we used in the experiment.

References

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, Jianfeng Gao, Po-Sen Huang, and Yejin Choi. 2018. Discourse-aware neural rewards for coherent text generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 173–184, New Orleans, Louisiana. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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

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

Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660, Florence, Italy. Association for Computational Linguistics.

Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. 2020. Content planning for neural story generation with aristotelian rescoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4319–4338, Online. Association for Computational Linguistics.

Xinyu Hua and Lu Wang. 2020. PAIR: Planning and iterative refinement in pre-trained transformers for long text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 781–793, Online. Association for Computational Linguistics.

Haozhe Ji and Minlie Huang. 2021. DiscoDVT: Generating long text with discourse-aware discrete variational transformer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4224, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. 2020. Language generation with multi-hop reasoning on commonsense knowledge graph. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 725–736, Online. Association for Computational Linguistics.

Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. 2021. Towards few-shot fact-checking via perplexity. In *Proceedings of the 2021 Conference of* *the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1971–1981, Online. Association for Computational Linguistics.

Nayeon Lee, Belinda Z. Li, Sinong Wang, Wen-tau Yih, Hao Ma, and Madian Khabsa. 2020. Language models as fact checkers? In *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*, pages 36–41, Online. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. PlotMachines: Outline-conditioned generation with dynamic plot state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4274–4295, Online. Association for Computational Linguistics.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6943–6951, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *arXiv preprint arXiv:2103.00453*.

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. Do massively pretrained language models make better storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861, Hong Kong, China. Association for Computational Linguistics.

Xiaofei Sun, Chun Fan, Zijun Sun, Yuxian Meng, Fei Wu, and Jiwei Li. 2020. Summarize, outline, and elaborate: Long-text generation via hierarchical supervision from extractive summaries. *arXiv preprint arXiv:2010.07074*.

Bowen Tan, Zichao Yang, Maruan Al-Shedivat, Eric Xing, and Zhiting Hu. 2021. Progressive generation of long text with pretrained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4313–4324, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jingjing Xu, Xuancheng Ren, Yi Zhang, Qi Zeng, Xiaoyan Cai, and Xu Sun. 2018. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4306–4315, Brussels, Belgium. Association for Computational Linguistics.

Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro. 2020. MEGATRON-CNTRL: Controllable story generation with external knowledge using large-scale language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2831–2845, Online. Association for Computational Linguistics.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

A Full List of Task Descriptions

Table 5 shows the complete list of task descriptions to generate various plot elements for content planning.

We use self-debiasing (Schick et al., 2021) when generating locations and casts to ensure the generated texts using each task description are distinct. The intuition of self-debiasing is to calculate the

Element	Task Description
Location	**Task:** Write the name of a country.\n **Country:** “
	Task: Write the name of a province.\n **Province:** “
	Task: Write the name of a city.\n **City:** “
	Task: Write the name of a county.\n **County:** “
Cast	**Task:** Write the male character’s full name in a story happened in <X1>. **Full name:** “
	Task: Write the female character’s full name in a story happened in <X1>. **Full name:** “
Genre	**Task:** Write a story genre.\n **Story genre:** “
	Task: Write a literary genre.\n **Literary genre:** “
	Task: Write a novel genre.\n **Novel genre:** “
Theme	**Task:** Write the main point from a <X1> story.\n **Main point:** “
	Task: Write the twist in a <X1> story.\n **Twist:** “
	Task: Write the lesson learned from a <X1> story.\n **Lesson learned:** “
	Task: Write the spectacle of a <X1> story.\n **Spectacle:** “

Table 5: Full list of task descriptions to generate each element. <X1> denotes the previously generated element. Please refer to Figure 1 for the dependency among plot elements and the task descriptions to generate the story body and ending.

token’s probability p_y assigned by the PLM using each task description. For each label y (specified by a task description), the token’s final logit is as follows:

$$\delta_y = p_y - \max_{y' \neq y} p_{y'} \tag{1}$$

We do not use self-debiasing when generating other plot elements because the task descriptions are complementary to each other.

B Post-Processing

We perform various post-processing depending on the plot elements. We rely on simple heuristics based on common errors we observe.

Including tailing punctuations For some plot elements, we expect a phrase instead of a whole sentence. However, the PLM sometimes inserts a punctuation mark, such as a full stop or a comma. Therefore, we recursively remove punctuations at the end till the last character is a letter.

Repeating the prompt We observe that the PLM sometimes repeats or rephrases the task description instead of trying to perform the task. Therefore, we filter out continuations that contain *any* word in the task description (excluding stop words and the text replacing the placeholder <X1>).

Generating 1st or 2nd person pronouns We usually do not expect first or second-person pronouns in a story plot. When the model generates plot elements containing first or second-person pronouns, it is often generic or opinionated, such as “I’ll try not to think about it” or “You will not fail me.” Therefore, we filter continuations containing a first or second-person pronoun of any case.

Ignoring task description When generating the story, we want to ensure that it includes essential plot elements specified in the task description. Therefore, we filter out a story if it contains fewer than two of the following {male character’s first name, female character’s first name, location}.

Table 6 overviews the post-processing applied when generating each type of output.

C Additional Experimental Setups

We generate a large set of plot elements offline in batch and store them to speed up the inference. The number of each plot element is shown in Table 8. When generating stories, we randomly sample plot elements and combine them to form a content plan. Table 7 presents the detailed parameters used for each type of generation.

For the Fusion model (Fan et al., 2018), we use the checkpoint provided in the official repository, which is fine-tuned on the WritingPrompts dataset with 300k prompt-story pairs. For ProGen (Tan et al., 2021), we use a two-stage seq2seq architecture, where the first seq2seq model takes the input keywords and generates a refined intermediate representation containing keywords with finer-grained details. The second seq2seq model then uses it as input and generates the final story. We fine-tune a BART-base model for both stages using 1k examples randomly sampled from the WritingPrompts dataset following Tan et al. (2021). We truncate

Post-processing	**Location**	**Cast**	**Genre**	**Theme**	**Body**	**Ending**
Remove tailing punctuations	✔	✔	✔			
Filter repeating prompt	✔	✔	✔	✔	✔	✔
Filter 1st & 2nd person pronouns				✔	✔	✔
Filter by plot elements					✔	

Table 6: Post-processing steps applied for each generation task.

Element	**num**	**min_len**	**max_len**
Location	20	1	5
Cast	10	1	5
Genre	20	1	5
Theme	10	5	25
Story Body	30	-	100
Story Ending	10	10	50

Table 7: Parameters used to generate each plot element and story parts. For plot elements, **num** indicates the number of entries to generate per *task description*. Please refer to Table 5 for the number of task descriptions for each plot element. For conditioned generation (cast, theme, story ending), **num** is the number of entries per *input* <X_I> and *task description* combination.

the generated stories from both baselines to 150 tokens till the last complete sentence to have the same length as stories generated by our model.

During training/generation, we use the tokenizers associated with the corresponding PLM in the HuggingFace library (Wolf et al., 2020). When calculating diversity and repetition, we use NLTK (Bird and Loper, 2004) to perform word tokenization. We calculate self-BLEU scores using NLTK's `sentence_bleu` method by treating each example as the reference in each round and averaging the BLEU scores over the whole dataset.

All experiments in this work are conducted on cloud instances with an NVIDIA Quadro P5000 GPU (16GB vRAM). The time to generate a story is roughly 1 minute, which includes generating multiple story bodies and endings and using scoring models to select the best candidate. Since we do not require any fine-tuning, using a CPU to perform inference is also possible. The reader can consider using a smaller GPT2-medium PLM instead of GPT2-XL when the resource is limited. The generation quality is comparable based on our observation.

D Evaluation on Generated Content Plan

We invite an *expert* annotator to rate each generated plot element [10]. We use binary rating (acceptable/unacceptable) for location, cast, and genre. We use a scale from 1 to 5 for theme because it is more subjective. We present the result in Table 8.

Element	Location	Cast	Genre	Theme
Count	43	493	24	117
Score	0.930	0.931	0.792	0.654

Table 8: Expert rated scores for generated plot elements normalized to the range of 0 (worst) to 1 (perfect).

E Details of Crowdsource Evaluation

We conducted the crowdsource evaluations on the Toloka platform [11]. In this section, we detail the specification of the annotation tasks, the quality control measures, and the stats of the annotation.

E.1 Annotation Task Specifications

For the fine-grained evaluation, we decompose it into a separate annotation task per aspect so that the annotators can focus on evaluating a single aspect and avoid context switching.

Fine-grained evaluation Rate each story in the following aspects on a scale of 1 (worse) to 5 (best).

- **Naturalness:** is the story fluent and understandable? The language should be natural. Minor grammatical errors are acceptable if they do not affect understanding the story.

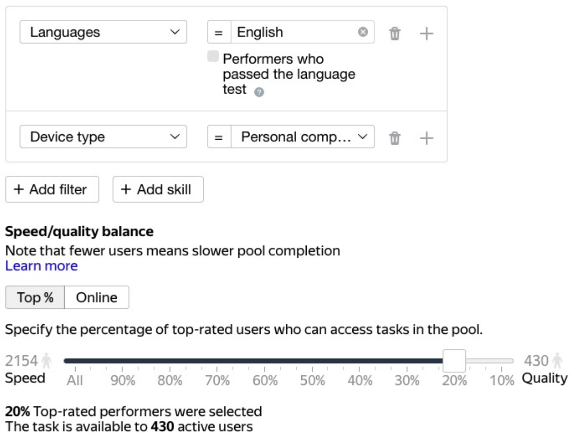
- **Interestingness:** is the story interesting to readers? Rate this aspect as objective as possible. Assuming someone familiar with the particular genre, will the story interest them?

[10]Plot elements are much shorter and faster to rate. Therefore, we use an expert annotator for superior accuracy.

[11]`https://toloka.ai/`

- **Cohesiveness:** is the story cohesive and logical? Common problems include mixing up the characters and introducing illogical event sequences (unless it appears like a deliberate choice).

Figure 5 shows the annotation interface and Figure 6, 7, 8 shows the detailed annotation instructions.


Languages = English
Performers who passed the language test
Device type = Personal comp...

Speed/quality balance
Note that fewer users means slower pool completion
Learn more

Top % | Online

Specify the percentage of top-rated users who can access tasks in the pool.

20% Top-rated performers were selected
The task is available to **430** active users

Figure 4: Audience filter for the annotation task pool.

Story ending evaluation Indicate which of the two stories has a better ending. A good story ending should be relevant to the story, logical, conclusive, and thoughtful. Figure 9 shows the full annotation instructions and Figure 10 shows the annotation UI.

E.2 Quality Control

We select crowdworkers who are fluent in English and among the 20% top-rated performers. Figure 4 shows a screenshot of the annotator filter. Additionally, they have to pass a short training session and correctly answer 3 out of 4 training questions to be selected for the main evaluation. If they answer a question wrongly during training, the system will show a hint to help them improve, as shown in Figure 11.

During annotation, we apply various quality control rules, including limiting each annotator to no more than 50 tasks, adding occasional captcha to block bots, banning users who consistently submit tasks too fast (less than 5 seconds for fine-grained evaluation and less than 10 seconds for story ending evaluation), and banning users who skip more than 5 tasks in a row.

E.3 Annotation Task Stats

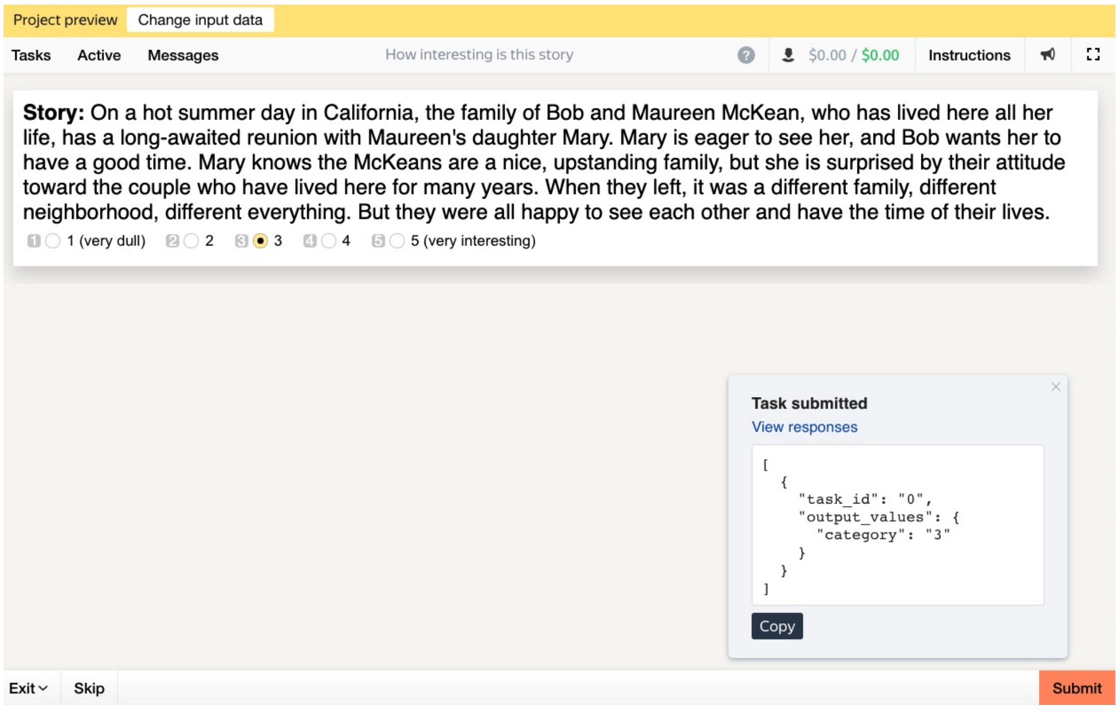
We paid \$0.05 for each fine-grained evaluation task. On average, it took around 30 seconds to complete each task, making the average earning \$6 an hour. There are around 40 crowdworkers evaluating for each aspect. Figure 12 shows an example pool stats for the naturalness evaluation.

We paid \$0.1 for each story ending evaluation task, which takes on average 1 minute 13 seconds to complete. There are in total 20 crowdworkers participating in this evaluation task.

The overall budget we spent on all crowdsource evaluations is \$200 (including payment and bonus to crowdworkers and platform fees).

F Sample Generated Stories

We present randomly sampled stories generated using different models in Table 9 and story endings selected by different ranking metrics in Table 10.

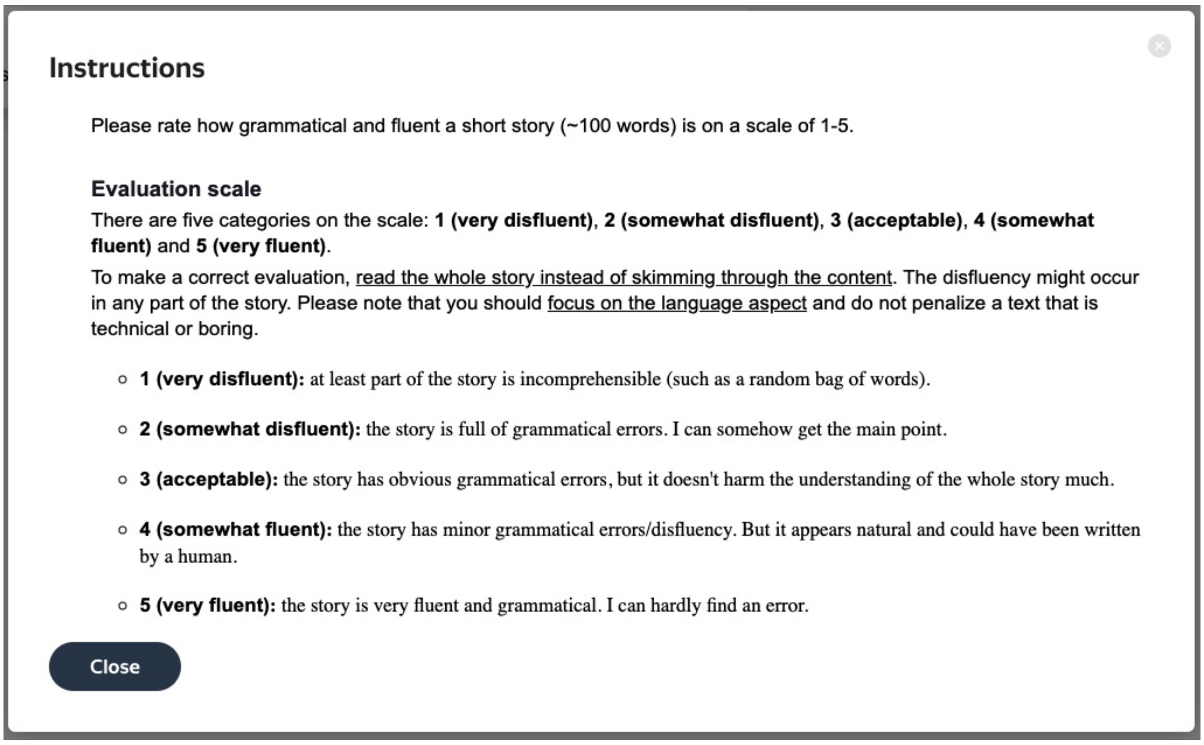


Story: On a hot summer day in California, the family of Bob and Maureen McKean, who has lived here all her life, has a long-awaited reunion with Maureen's daughter Mary. Mary is eager to see her, and Bob wants her to have a good time. Mary knows the McKeans are a nice, upstanding family, but she is surprised by their attitude toward the couple who have lived here for many years. When they left, it was a different family, different neighborhood, different everything. But they were all happy to see each other and have the time of their lives.

1 (very dull) 2 [✓ 3] 4 5 (very interesting)

```
[
  {
    "task_id": "0",
    "output_values": {
      "category": "3"
    }
  }
]
```

Figure 5: Annotation interface for the interestingness aspect. The interface for other aspects are analogous and we omit them for brevity.



Instructions

Please rate how grammatical and fluent a short story (~100 words) is on a scale of 1-5.

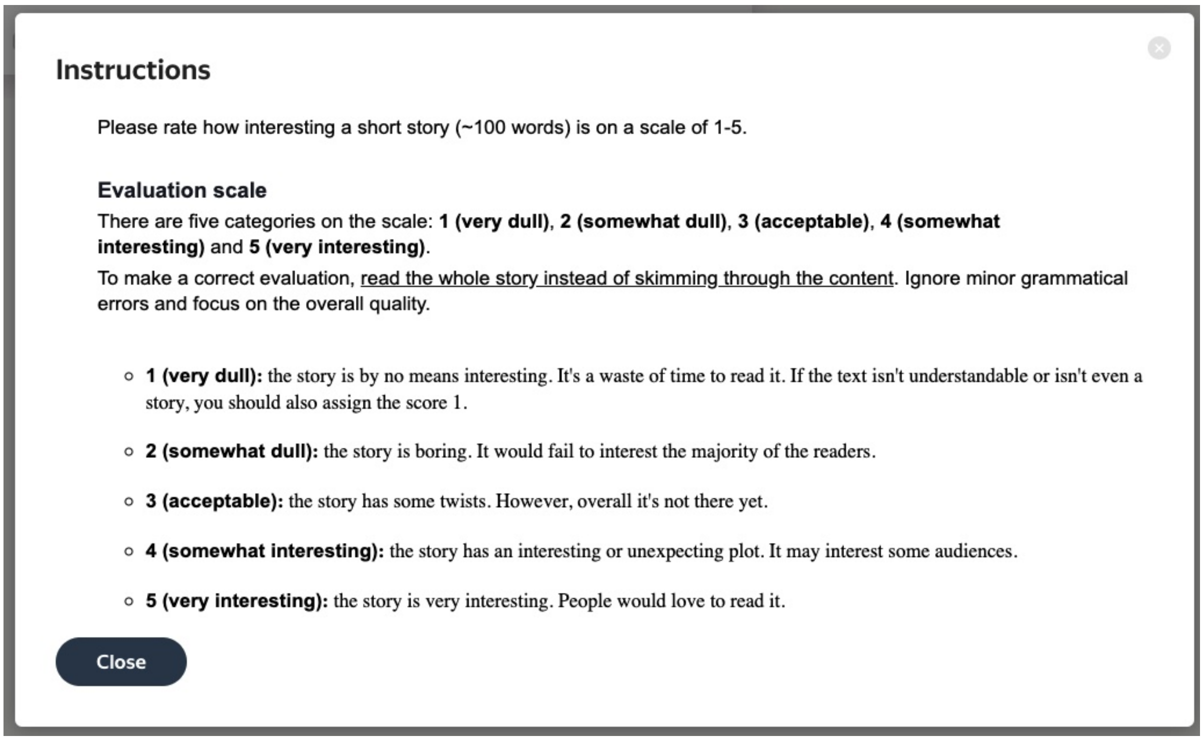
Evaluation scale

There are five categories on the scale: **1 (very disfluent)**, **2 (somewhat disfluent)**, **3 (acceptable)**, **4 (somewhat fluent)** and **5 (very fluent)**.

To make a correct evaluation, read the whole story instead of skimming through the content. The disfluency might occur in any part of the story. Please note that you should focus on the language aspect and do not penalize a text that is technical or boring.

- **1 (very disfluent):** at least part of the story is incomprehensible (such as a random bag of words).
- **2 (somewhat disfluent):** the story is full of grammatical errors. I can somehow get the main point.
- **3 (acceptable):** the story has obvious grammatical errors, but it doesn't harm the understanding of the whole story much.
- **4 (somewhat fluent):** the story has minor grammatical errors/disfluency. But it appears natural and could have been written by a human.
- **5 (very fluent):** the story is very fluent and grammatical. I can hardly find an error.

Figure 6: Annotation instructions for the naturalness aspect.


Instructions

Please rate how interesting a short story (~100 words) is on a scale of 1-5.

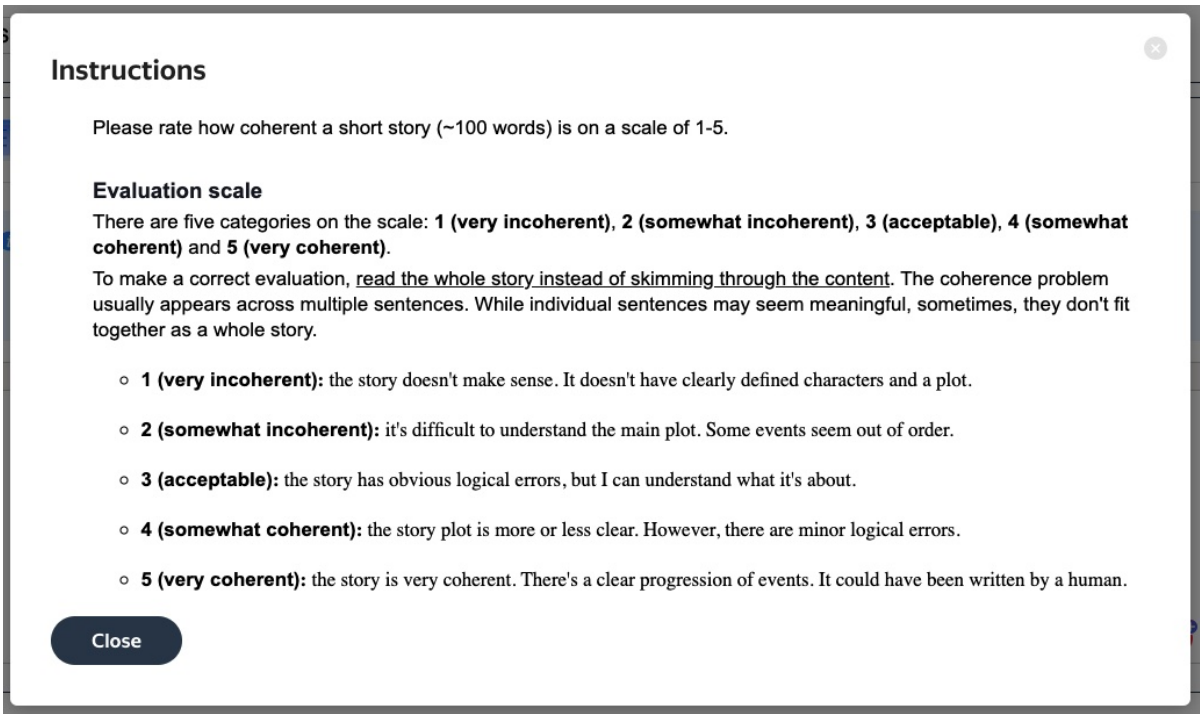
Evaluation scale

There are five categories on the scale: **1 (very dull)**, **2 (somewhat dull)**, **3 (acceptable)**, **4 (somewhat interesting)** and **5 (very interesting)**.

To make a correct evaluation, read the whole story instead of skimming through the content. Ignore minor grammatical errors and focus on the overall quality.

- **1 (very dull):** the story is by no means interesting. It's a waste of time to read it. If the text isn't understandable or isn't even a story, you should also assign the score 1.
- **2 (somewhat dull):** the story is boring. It would fail to interest the majority of the readers.
- **3 (acceptable):** the story has some twists. However, overall it's not there yet.
- **4 (somewhat interesting):** the story has an interesting or unexpecting plot. It may interest some audiences.
- **5 (very interesting):** the story is very interesting. People would love to read it.

Figure 7: Annotation instructions for the interestingness aspect.


Instructions

Please rate how coherent a short story (~100 words) is on a scale of 1-5.

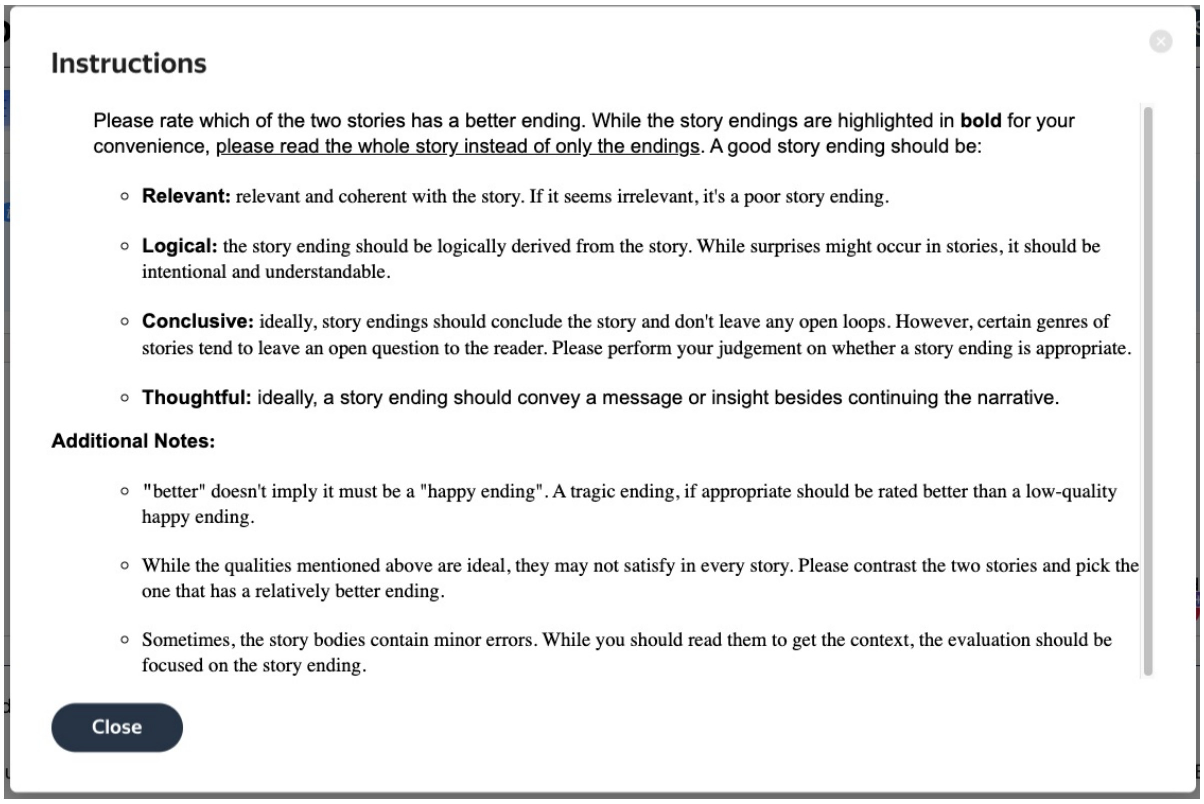
Evaluation scale

There are five categories on the scale: **1 (very incoherent)**, **2 (somewhat incoherent)**, **3 (acceptable)**, **4 (somewhat coherent)** and **5 (very coherent)**.

To make a correct evaluation, read the whole story instead of skimming through the content. The coherence problem usually appears across multiple sentences. While individual sentences may seem meaningful, sometimes, they don't fit together as a whole story.

- **1 (very incoherent):** the story doesn't make sense. It doesn't have clearly defined characters and a plot.
- **2 (somewhat incoherent):** it's difficult to understand the main plot. Some events seem out of order.
- **3 (acceptable):** the story has obvious logical errors, but I can understand what it's about.
- **4 (somewhat coherent):** the story plot is more or less clear. However, there are minor logical errors.
- **5 (very coherent):** the story is very coherent. There's a clear progression of events. It could have been written by a human.

Figure 8: Annotation instructions for the cohesiveness aspect.



Instructions

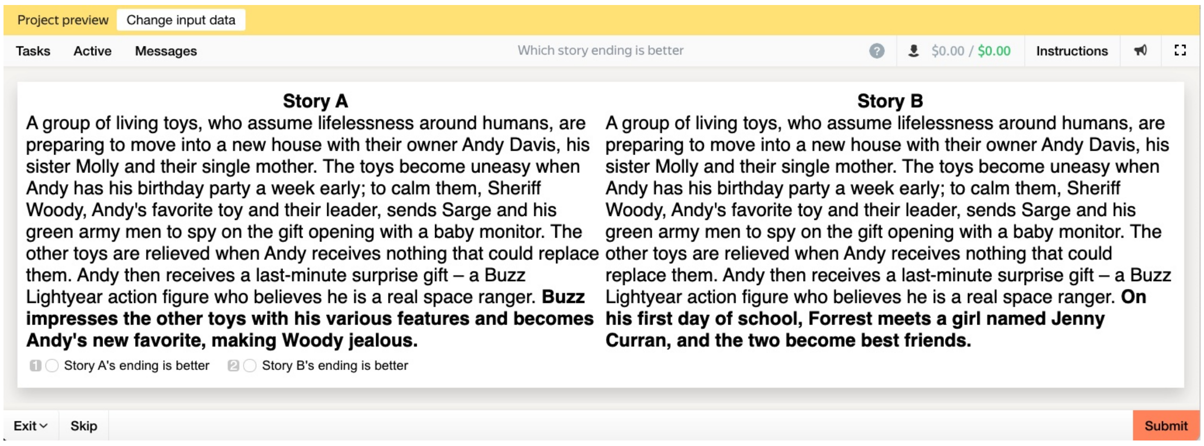
Please rate which of the two stories has a better ending. While the story endings are highlighted in **bold** for your convenience, please read the whole story instead of only the endings. A good story ending should be:

- **Relevant:** relevant and coherent with the story. If it seems irrelevant, it's a poor story ending.
- **Logical:** the story ending should be logically derived from the story. While surprises might occur in stories, it should be intentional and understandable.
- **Conclusive:** ideally, story endings should conclude the story and don't leave any open loops. However, certain genres of stories tend to leave an open question to the reader. Please perform your judgement on whether a story ending is appropriate.
- **Thoughtful:** ideally, a story ending should convey a message or insight besides continuing the narrative.

Additional Notes:

- "better" doesn't imply it must be a "happy ending". A tragic ending, if appropriate should be rated better than a low-quality happy ending.
- While the qualities mentioned above are ideal, they may not satisfy in every story. Please contrast the two stories and pick the one that has a relatively better ending.
- Sometimes, the story bodies contain minor errors. While you should read them to get the context, the evaluation should be focused on the story ending.

Figure 9: Annotation instructions for the story ending evaluation.



Story A

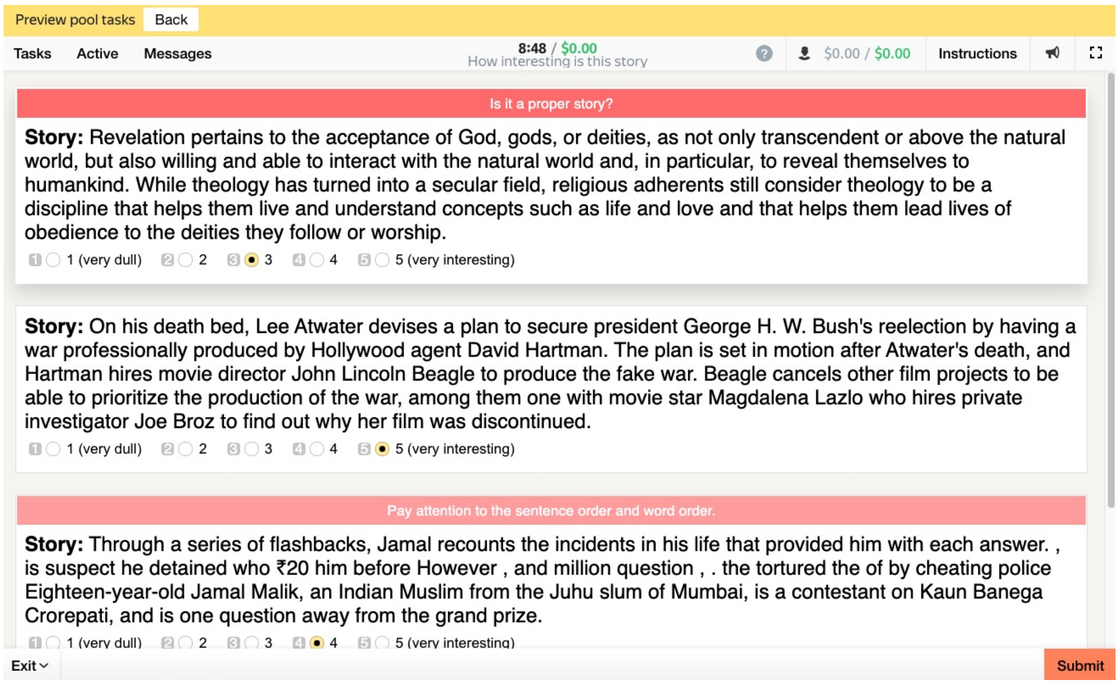
A group of living toys, who assume lifelessness around humans, are preparing to move into a new house with their owner Andy Davis, his sister Molly and their single mother. The toys become uneasy when Andy has his birthday party a week early; to calm them, Sheriff Woody, Andy's favorite toy and their leader, sends Sarge and his green army men to spy on the gift opening with a baby monitor. The other toys are relieved when Andy receives nothing that could replace them. Andy then receives a last-minute surprise gift – a Buzz Lightyear action figure who believes he is a real space ranger. **Buzz impresses the other toys with his various features and becomes Andy's new favorite, making Woody jealous.**

Story B

A group of living toys, who assume lifelessness around humans, are preparing to move into a new house with their owner Andy Davis, his sister Molly and their single mother. The toys become uneasy when Andy has his birthday party a week early; to calm them, Sheriff Woody, Andy's favorite toy and their leader, sends Sarge and his green army men to spy on the gift opening with a baby monitor. The other toys are relieved when Andy receives nothing that could replace them. Andy then receives a last-minute surprise gift – a Buzz Lightyear action figure who believes he is a real space ranger. **On his first day of school, Forrest meets a girl named Jenny Curran, and the two become best friends.**

1 ○ Story A's ending is better 2 ○ Story B's ending is better

Figure 10: Annotation interface for the pair-wise story ending evaluation.



Is it a proper story?

Story: Revelation pertains to the acceptance of God, gods, or deities, as not only transcendent or above the natural world, but also willing and able to interact with the natural world and, in particular, to reveal themselves to humankind. While theology has turned into a secular field, religious adherents still consider theology to be a discipline that helps them live and understand concepts such as life and love and that helps them lead lives of obedience to the deities they follow or worship.

1 (very dull) 2 [✓ 3] 4 5 (very interesting)

Story: On his death bed, Lee Atwater devises a plan to secure president George H. W. Bush's reelection by having a war professionally produced by Hollywood agent David Hartman. The plan is set in motion after Atwater's death, and Hartman hires movie director John Lincoln Beagle to produce the fake war. Beagle cancels other film projects to be able to prioritize the production of the war, among them one with movie star Magdalena Lazlo who hires private investigator Joe Broz to find out why her film was discontinued.

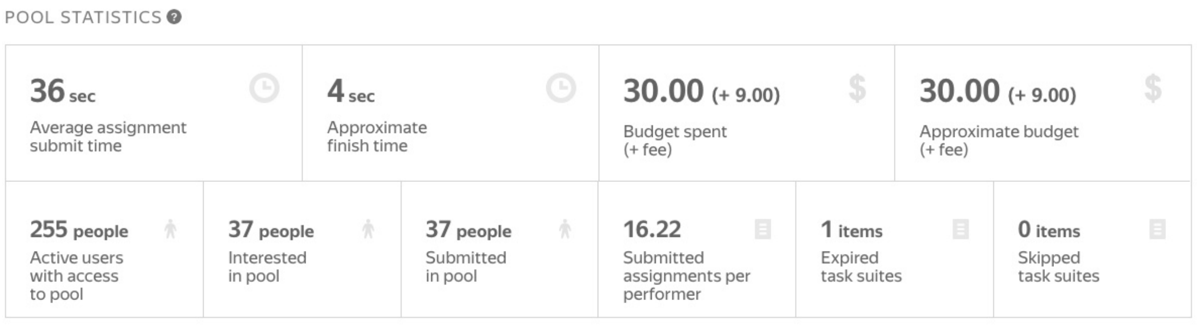
1 (very dull) 2 3 4 [✓ 5 (very interesting)]

Pay attention to the sentence order and word order.

Story: Through a series of flashbacks, Jamal recounts the incidents in his life that provided him with each answer. , is suspect he detained who ₹20 him before However , and million question , . the tortured the of by cheating police Eighteen-year-old Jamal Malik, an Indian Muslim from the Juhu slum of Mumbai, is a contestant on Kaun Banega Crorepati, and is one question away from the grand prize.

1 (very dull) 2 3 [✓ 4] 5 (very interesting)

Figure 11: Screenshot of the training session. A hint is displayed for the examples where the annotator provides the wrong answer.



POOL STATISTICS

36 sec Average assignment submit time	4 sec Approximate finish time	30.00 (+ 9.00) Budget spent (+ fee)	30.00 (+ 9.00) Approximate budget (+ fee)

255 people Active users with access to pool	37 people Interested in pool	37 people Submitted in pool	16.22 Submitted assignments per performer	1 items Expired task suites	0 items Skipped task suites

Figure 12: The annotation pool stats for the naturalness evaluation.

Location:	Alameda County
Cast:	John *and* Cheryl
Genre:	literary fiction
Theme:	The only problem with the house was the people.
Fusion:	“What?” she whispered, as she pulled a small piece from her coat. It was a small piece of paper, with a picture of a young man sitting on that couch. “ What?” he said, as he flipped the picture around. “My name is John, and I ’m here to give you a story.” he said, as the photo of his wife came out of his hand and he pulled out a book. The picture was of some kind of book, with the Briggs on it. “What” she said, as she looked at it . “ I don’t know,” John said, as he read the book. “It was about to get pretty bad. I don’t like it here.” “It’s not good,” she said. “What did you tell me?” he asked, as the picture was on the couch. “I just can’t believe it
ProGen:	The first son I held inside my wife skin and body, I rested my head on her chest and prayed to the waiting Forbiddone, loved by the boy and girl alike, and that the house would bring smiles to all. Jerry Fairey, anxious and excited, pushed his face into my naked flesh. Jerry sounded just like his daddy and chirped, “Grr!” and they laughed together. Jerry took a deep breath, thought of again, “Fuck this. This place is empty, people.” Jerry slept.
ScratchPlot:	John and Cheryl had decided that they’d get a divorce. They’d never been married, and they’d always been in the midst of an argument that had escalated to violence. John had left the house, locked the door behind him, and walked into the woods. Cheryl had followed him, but after a brief conversation with her father and a call to an ex-boyfriend from his stepmother, she had walked out to the car. After John had walked down the driveway, he came up behind Cheryl in her driveway and punched her in the face with both fists.
- content plan:	The plot summary tells the reader what the story is about. Plot summaries are very long and sometimes the authors need to explain some things and show the readers something. I use a simple one here: The plot summaries of most of the books I read are in the order of their publication date (as seen above). The authors do not like the readers to have to remember this order, so they just list the events in a very long way. I have seen plots summary in the form of a list in the back of a book, but that has problems: the reader has to find the book. A plot summary in a book will be much more readable and more likely to get across the events.

Table 9: Stories generated by different models. All models except for “- content plan” use the same plot element(s) listed above.

Location:	Gotham
Cast:	Jack Ryan *and* Doreen Green
Genre:	the great american novel
Theme:	a series of ornaments, decorations, or decorations as distinguished from a picture
ScratchPlot:	Jack Ryan was a detective for the Gotham City Police Department who specialized in counter-terrorism and intelligence work. Doree Green was a journalist who specialized on terrorism, and was a regular contributor to Gothamist. Dormammu was a powerful villain with a reputation that reached far into the Gotham underground. Jack began investigating Dormalm and the mysterious Ornaments of Death. `[SEP]` After his first interview with Dormammu, Jack Ryan's mind is shattered, and it's not long before he has no memory of who he is.
NSP:	Doreen's sister is in prison for murder. Jack Ryan, the most brilliant and ruthless man the government has ever produced, is trying to rescue her. He's in Gotham and the place is swarming with criminals. Jack, with the help of two friends, the brilliant and deadly Jack Ryder and the deadly and brilliant Tommy Elliot, sets out to rescue Doree, and make things right. `[SEP]` They have the world at their feet but they have the eyes of a hawk.
PPL:	Jack and Dora Green, the daughter of a major newspaper publisher, live in the Gotham Mansion, the home of Jack's business partner, Doree Green, and the mother of their three-year old daughter, Dora. Their home is a large home with a large lawn. A large stone sculpture of a man stands at the front of the front door, which is marked with a red paint that is difficult to remove without destroying the painting. `[SEP]` It was a beautiful day, sunny and warm. The grass was green and lush. The air was clear, and crisp. It was the perfect summer day.
Random:	Jack Ryan was a detective for the Gotham City Police Department who specialized in counter-terrorism and intelligence work. Doree Green was a journalist who specialized on terrorism, and was a regular contributor to Gothamist. Dormammu was a powerful villain with a reputation that reached far into the Gotham underground. Jack began investigating Dormalm and the mysterious Ornaments of Death. `[SEP]` And then there was a loud thud, like the sound of an iron fist smashing into concrete, and then the sky was a blood red.

Table 10: Story body and ending selected by different algorithms. We manually insert a `[SEP]` token to indicate the boundary between the story body and ending.