Choose Your Own Adventure: Paired Suggestions in Collaborative Writing for Evaluating Story Generation Models

Elizabeth Clark[1] and **Noah A. Smith**[1,2]
[1]Paul G. Allen School of Computer Science & Engineering, University of Washington
[2]Allen Institute for Artificial Intelligence
{eaclark7,nasmith}@cs.washington.edu

Abstract

Story generation is an open-ended and subjective task, which poses a challenge for evaluating story generation models. We present CHOOSE YOUR OWN ADVENTURE, a collaborative writing setup for pairwise model evaluation. Two models generate suggestions to people as they write a short story; we ask writers to choose one of the two suggestions, and we observe which model's suggestions they prefer. The setup also allows further analysis based on the revisions people make to the suggestions. We show that these measures, combined with automatic metrics, provide an informative picture of the models' performance, both in cases where the differences in generation methods are small (nucleus vs. top-k sampling) and large (GPT2 vs. Fusion models).

1 Introduction

Systems that automatically generate text suggestions to human authors have emerged as a new application of natural language generation models. Evaluating such models, however, is challenging. Typically, writers rate a single system's quality after some period of use, for example while authoring an entire story or poem (e.g., Clark et al., 2018; Ghazvininejad et al., 2017). A model's quality is measured using Likert scale scores, sometimes combined with additional analysis, like the type or quantity of writer edits (e.g., Roemmele and Gordon, 2015; Akoury et al., 2020).

In contrast, a *pairwise* system evaluation—where evaluators are given two suggestions at the same time and asked to choose between them—would allow researchers to compare generation models directly. Comparative evaluations have been shown to produce more reliable and consistent results than Likert-scale ratings (Callison-Burch et al., 2007; Kiritchenko and Mohammad, 2017), and they have been used to evaluate natural language generation systems for translation and dialogue (Otani et al., 2016; Sedoc et al., 2019).

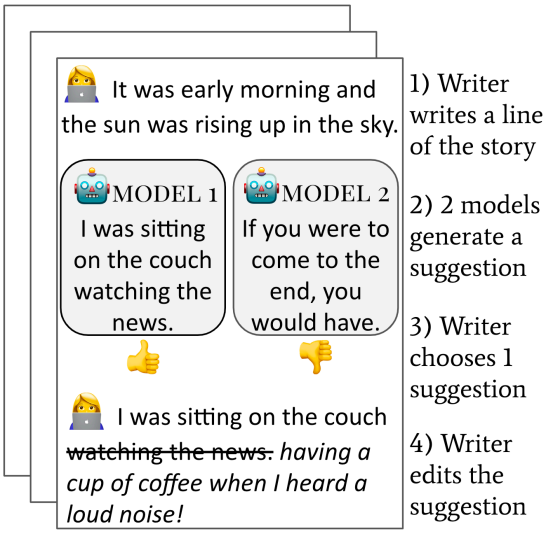

Figure 1: CYOA has a writer write a line of the story alone, and then two models generate suggestions for the next line. The writer chooses one (in this case, MODEL1), edits it, and then adds it to the story. They repeat this process 5 times. CYOA collects writers' preferences between the two models, along with the human-authored, machine-generated, and human-edited text, to evaluate the models.

We propose CHOOSE YOUR OWN ADVENTURE (CYOA), a protocol for pairwise evaluations of collaborative writing models, focusing on story generation. Instead of scoring a single model, we compare two models. At fixed points during the writing process, each generates a suggestion, and writers choose one to continue their story (see Fig. 1). The result is utterance-level feedback on which model's generated text writers prefer at that point in the story. Along with the writer's revisions to the generated suggestions and comparisons between the generated and human-authored portions of the story, this evidence can help a researcher answer the following questions about their model:

1. Is my model better at generating story suggestions than a baseline model?
2. How useful are my model's suggestions?

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3566–3575
June 6–11, 2021.

3. How does my model's generated text compare to human-authored text?

In this paper, we show how CYOA can answer these questions and provide insights into story model behavior, both in cases when the expected differences in text quality are large (e.g., the text is generated with two different models; §3) and when they are small (e.g., the text is generated with the same model but using two different sampling methods; §4).

CYOA allows human and automatic evaluations to be collected simultaneously; we run standard automatic evaluations of text quality on the collaboratively-generated text and get results consistent with previous analyses of "statically"-generated text. CYOA is useful to both NLG researchers and story writers; writers report being happy with the stories they write with the system and that the paired suggestions help them come up with new ideas. We release a template website for CYOA and the evaluation script[1] to support future story and collaborative writing evaluation work.

2 CHOOSE YOUR OWN ADVENTURE

CYOA evaluates a pair of story generation models by having people select and interact with text generated by each of the models as they write a story. Both models generate suggestions for the writer at the same point in the story, and the writer must choose between the two suggestions, forcing a pairwise comparison of the two models. By having multiple people write stories with the two models, we can aggregate their preferences and interactions with the suggestions and analyze them to provide feedback on the two models.

2.1 Writing Setup

To allow the writers control over the story while still encouraging them to use the suggestions, CYOA uses a turn-taking writing process, with writers alternating between writing by themselves and then receiving suggestions to continue the story (Swanson and Gordon, 2012; Clark et al., 2018).

The writer begins the story by writing the first sentence alone; an image (Fig. 2 in App. A) is provided as an optional prompt to help them get started. Once the writer submits the writing from their turn, two models each generate a suggestion to continue the story, which are presented to the writer in random order. As shown in Fig. 1, the writer then chooses which of the suggestions they prefer and edits it as they wish before adding it to the story. It is then the writer's turn to write alone again. This process repeats 5 times, at which point the story is finished and submitted.

Each "turn" in the story has to be between 20 and 260 characters for it to be submitted to the story. Other than length, there is no restriction on how writers can edit the suggestions; they can delete the suggestion entirely or submit it as-is. When editing a computer-generated suggestion, the writer can change their mind and select the other model's suggestion instead, but once a writer submits a turn, they cannot go back to edit it later.

After the finished story is submitted, participants are asked Likert-scale and open-ended questions about the system and the suggestions they received. We asked participants to indicate on a 5-point Likert scale (ranging from "Strongly Disagree" to "Strongly Agree") how much they agreed with the following statements:

- I'm happy with my final story.
- I felt the system and I were working collaboratively to write the story.
- I thought having the suggestions was useful while writing the story.
- The suggestions connected to what had happened in the story so far.
- The suggestions helped me come up with new ideas.

We then provided textboxes for them to write their responses to the following questions:

- What made you choose one suggestion over another?
- What were you looking for in the suggestions?

We chose these questions for this project to capture people's reactions to the overall writing setup and a general sense of areas for improving story generation models. However, these questions could be eliminated or adjusted to fit the evaluation goals of the researcher.

A demo of CYOA is at homes.cs.washington.edu/~eaclark7/multi-model-demo.

2.2 Evaluation Setup

From the writing setup, we collect the generated suggestions from each model, the writers' preferences between the two models, and the revisions they make to the generated text. We analyze these sources of information to answer three questions

[1] github.com/eaclark07/cyoa

NLP practitioners have when evaluating their models. There are many analyses researchers could run with the data gathered from CYOA beyond those listed here; we include some examples.

(Q1) Is my model better at generating story suggestions than a baseline model? CYOA reports how many of the model's suggestions people chose to work with vs. the baseline's suggestions. We further break this down by the suggestion round (1–5) to see if the writers' preferences change over the course of the story.

Another option would be to break down the writers' preferences by writer attributes, e.g., to analyze the effect of the author on the stories or desired suggestions (August et al., 2020).

(Q2) How useful are the models' suggestions? We analyze the revisions writers make to the suggestions to see how much of the generated text they find useful for continuing their story. We use three metrics to see how much of the original text is preserved after a writer's revisions. Levenshtein edit distance measures the number of character insertions, deletions, and substitutions the writers made, and Jaccard similarity measures the proportion of tokens that are shared between the original and the edited text. User Story Edit Ratings (USER; Akoury et al., 2020)[2] measures similarity by recursively counting the longest contiguous substrings between the edited and the original text.

These edit-based metrics capture exact matches between the texts, measuring how much of the generated content makes it to the final story in the strictest sense. However, other metrics could be used if the researcher is interested in capturing broader notions of similarity, e.g., embedding-based measures like cosine similarity or BERTScore (Zhang et al., 2020).

(Q3) How do the models' generated texts compare to human-authored text? Pairwise comparison gives us the models' relative quality; comparing them to human-authored text gives an idea of their absolute quality. To do this, we take the parts of the story the writer wrote alone (i.e., the turns without generated suggestions) and compare it to the generated text. We look at average sentence length (a common proxy for text complexity in stories; See et al., 2019; Roemmele et al., 2017) and distinct-n, a measure of repetition (Li et al., 2016). As in See et al. (2019), we also look the concreteness of the text's nouns and verbs, using the concreteness ratings from Brysbaert et al. (2014).[3]

If the system is being used to evaluate a model that focuses on a specific aspect of stories, e.g., events or characters, this analysis could be extended to compare how these specific elements are introduced and referenced in the machine-generated vs. human-authored text.

3 Experiment #1: FUSION vs. GPT2

We first test CYOA with two popular story generation models: (1) FUSION, the fusion model from Fan et al. (2018), which uses a fusion mechanism to combine two convolutional sequence-to-sequence models; and (2) GPT2, the small GPT2 model (Radford et al., 2019) finetuned on story data and using top-k sampling (Fan et al., 2018).

We compare FUSION and GPT2 to see how CYOA can evaluate two models with different underlying architectures; they are also both common story generation baselines (See et al., 2019; Xu et al., 2020; Rashkin et al., 2020).

To train the models, we use the WritingPrompts dataset (Fan et al., 2018), a collection of writing prompts from Reddit paired with stories. During the CYOA evaluation, both models generate their suggestions conditioned on the whole story written so far. (Data and model details in App. B and C.)

We run CYOA on Amazon Mechanical Turk with 105 Turkers to compare the two models. Each Turker can only complete the task once. Turkers are required to have over 1,000 tasks approved, have an 95% approval rate, and be from the United States, and they are paid $2.50 for participating in the study. The study was approved by our instiution's Institutional Review Board.

We break down our results and discussion by the research questions listed in §2.2.

(Q1) Table 1 shows that, of the 525 suggestion pairs, Turkers significantly[4] preferred the GPT2 suggestions over FUSION, choosing them 65.7% of the time. Breaking it down by suggestion round 1–5, the writers' preference for the GPT2 was largest at the beginning of the story and decreased over the course of the story. To understand why, we look at how writers edited the suggestions and how the generated text compared to human-authored text.

[2] github.com/dojoteef/storium-frontend

[3] Sentence length, concreteness, and distinct-n: github.com/abisee/story-generation-eval

[4] Binomial test: $p < 0.01$.

	Total	#1	#2	#3	#4	#5
% GPT2	66	76	70	63	63	57

Table 1: % of chosen suggestions that are from GPT2.

	ED (↓)	JS (↑)	USER (↑)
FUSION	37.61	51.13	60.69
GPT2	**29.49**	**61.35**	**71.77**

Table 2: Edit distance, Jaccard similarity, and USER scores between the edited and the original suggestions.

(Q2) In Table 2, all three edit metrics show that writers used significantly[5] more of the accepted GPT2 suggestion text in their story than the accepted FUSION suggestion text. When we break down the scores by round, we see that this is true regardless of where in writer is in the story (see Table 7 in App. D.1). Taken with the pairwise results, this points to GPT2 as the better collaborative story generation model. FUSION, perhaps due to its hierarchical structure, did not generate as many useful suggestions as GPT2 in the interactive setting.

(Q3) Finally, we look at how the generated text compares to the story text the writers wrote alone. From Table 3, we see that GPT2 generates shorter, more concrete, and more repetitive suggestions than FUSION.

Both models generate shorter sentences than people, and GPT2 generates more concrete nouns and verbs than FUSION, corroborating the analysis of See et al. (2019). GPT2 generated the most repetitive text, which may explain why it is chosen less frequently as the story goes on. FUSION's subhuman level of repetition indicates it often fails to refer back to the story context, as illustrated by the low Likert-scale scores for *The suggestions connected to what had happened in the story so far.* (Fig. 3 in App. D.2).

4 Experiment #2: NUCLEUS vs. TOP-K

Our second experiment compares text generated from GPT2 but now using different sampling strategies: TOP-K (as in §3) and NUCLEUS sampling (Holtzman et al., 2020). (Model details in App. C.) Here we expect to see narrower differences in the generated text than we did in §3. Comparing TOP-K vs. NUCLEUS focuses on CYOA's ability to compare models with fine-grained differences. 103 Turkers[6] write a story with the help of suggestions from this pair of models.

[5]Mann-Whitney U test: $p < 0.01$.

	FUSION	GPT2	HUMAN
avg. sent. len.	13.70	10.31	18.86
concrete N	4.04	4.35	4.17
concrete V	2.90	3.10	3.12
distinct-1	0.75	0.53	0.72
distinct-2	0.97	0.70	0.95
distinct-3	1.00	0.76	0.99

Table 3: Generated text results for the FUSION and GPT2-generated text, compared to the HUMAN-written portions of the story.

	Total	#1	#2	#3	#4	#5
% TOP-K	53	58	53	53	53	49

Table 4: % of chosen suggestions that are from TOP-K.

(Q1) Table 4 shows Turkers preferred the TOP-K suggestions over the NUCLEUS suggestions for 53.4% of the 515 suggestion pairs writers received; as expected, a smaller difference than in §3 and not significant.[7] Again, the writers' preference for TOP-K decreased over the course of the story, with NUCLEUS slightly more popular by the end.

(Q2) In Table 5, all three metrics show that writers used more of the NUCLEUS-sampled text than the TOP-K-sampled text, though the difference is not significant.[8] Despite writers' slight preference for TOP-K-sampled suggestions, when they choose NUCLEUS-sampled suggestions, they preserve more of the generated text. Table 8 (App. D.1) shows that difference is largest at the beginning and end of the story. This suggests TOP-K's safer suggestions may be less useful, especially when starting or finishing the task.

[6]These are a separate set of Turkers from §3, but subject to the same requirements.

[7]Binomial test: $p = 0.07$.

[8]Mann-Whitney U test: $p = 0.19$ (ED), $p = 0.23$ (JS), and $p = 0.27$ (USER).

	ED (↓)	JS (↑)	USER (↑)
NUCLEUS	**34.65**	**53.64**	**63.64**
TOP-K	36.69	50.96	62.18

Table 5: Edit distance, Jaccard similarity, and USER scores between the edited and the original suggestions.

	NUCLEUS	TOP-K	HUMAN
avg. sent. len.	12.76	10.53	19.28
concrete N	4.15	4.34	4.23
concrete V	3.08	3.08	3.11
distinct-1	0.77	0.60	0.72
distinct-2	0.96	0.78	0.96
distinct-3	0.99	0.84	0.99

Table 6: Generated text results for the TOP-K and NUCLEUS-generated text, compared to the HUMAN-written portions of the story.

(Q3) Table 6 shows that TOP-K-generated text is shorter, more concrete, and more repetitive than NUCLEUS-generated text. NUCLEUS's text comes closer to human-levels of repetition, consistent with the findings of Holtzman et al. (2020) and Akoury et al. (2020).

5 Writer Feedback

CYOA benefits writers as well as researchers. The results of the writer feedback across both experiments indicate that writers enjoy the paired-suggestion writing experience, regardless of which models they wrote with. The Likert-scale responses were particularly positive for *I'm happy with my final story.* (FUSION vs. GPT2: mean = 3.83, NUCLEUS vs. TOP-K: mean = 3.84) and *The suggestions helped me come up with new ideas.* (FUSION vs. GPT2: mean = 3.80, NUCLEUS vs. TOP-K: mean = 3.79). This compares favorably to single-suggestion collaborative story writing systems that use a similar writing process; Clark et al. (2018) report writers gave a mean score of 3.28[9] for happiness with the story they wrote with their collaborative writing system. Full Likert-scale results are in App. D.2.

The positive reactions from participants indicate this format could work well on alternative crowdsourcing platforms, like LabintheWild,[10] or launched as an independent writing game, similar to Akoury et al. (2020).

6 Related Work

Collaborative writing systems have been developed in domains like poetry (Ghazvininejad et al., 2017), slogans (Clark et al., 2018), and stories (Roemmele and Gordon, 2015; Goldfarb-Tarrant et al., 2019; Akoury et al., 2020). Like Storium (Akoury et al., 2020), we focus on the potential to use these systems as evaluation platforms. However, we suggest using paired suggestions in collaborative writing systems to directly compare generation models.

ChatEval (Sedoc et al., 2019) collects human evaluations for paired chatbot utterances and Otani et al. (2016) for paired translations, but the generated text is static. By having writers interact with dynamically generated suggestions, collaborative writing systems reward *helpful* and *robust* generation models, underemphasized attributes in current evaluations (Zellers et al., 2021; Ethayarajh and Jurafsky, 2020).

7 Conclusion

CYOA allows researchers to collect human and automatic evaluations for story generation models in a single collaborative writing task. The paired suggestions allow direct comparisons between two models, and automatic-metric comparisons among generated text, its revisions, and the human-authored portions provide additional insight. We expect CYOA evaluations to accelerate progress on applications for collaborative writing between humans and machines.

Acknowledgments

This research was supported in part by a NSF graduate research fellowship and the DARPA CwC program through ARO (W911NF-15-1-0543). The authors would also like to thank the ARK group, Ari Holtzman, Nader Akoury, and Yejin Choi for their help and feedback, the reviewers for their helpful comments, and the participants who took part in our study.

References

Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. STORIUM: A Dataset and Evaluation Platform for Machine-in-the-Loop Story Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6470–6484.

Tal August, Maarten Sap, Elizabeth Clark, Katharina Reinecke, and Noah A. Smith. 2020. Exploring the effect of author and reader identity in online story writing: the STORIESINTHEWILD corpus. In *Proceedings of the First Joint Workshop on Narrative*

[9] Scoring adjusted to a 5-point scale.

[10] www.labinthewild.org

Understanding, Storylines, and Events, pages 46–54.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods*, 46:904–911.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158.

Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. 2018. Creative writing with a machine in the loop: Case studies on slogans and stories. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces*, pages 329–340.

Kawin Ethayarajh and Dan Jurafsky. 2020. Utility is in the eye of the user: A critique of NLP leaderboards. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4846–4853.

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.

Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. 2017. Hafez: an interactive poetry generation system. In *Proceedings of ACL 2017, System Demonstrations*, pages 43–48.

Seraphina Goldfarb-Tarrant, Haining Feng, and Nanyun Peng. 2019. Plan, write, and revise: an interactive system for open-domain story generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 89–97.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *Proceedings of the 2020 International Conference on Learning Representations*.

Svetlana Kiritchenko and Saif Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 465–470.

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.

Naoki Otani, Toshiaki Nakazawa, Daisuke Kawahara, and Sadao Kurohashi. 2016. IRT-based aggregation model of crowdsourced pairwise comparison for evaluating machine translations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 511–520.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In *OpenAI blog*.

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.

Melissa Roemmele and Andrew S. Gordon. 2015. Creative help: A story writing assistant. In *Proceedings of the International Conference on Interactive Digital Storytelling*.

Melissa Roemmele, Andrew S. Gordon, and Reid Swanson. 2017. Evaluating story generation systems using automated linguistic analyses. In *Proceedings of the Workshop on Machine Learning for Creativity*.

João Sedoc, Daphne Ippolito, Arun Kirubarajan, Jai Thirani, Lyle Ungar, and Chris Callison-Burch. 2019. ChatEval: A tool for chatbot evaluation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 60–65.

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.

Reid Swanson and Andrew S. Gordon. 2012. Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling. *ACM Transactions on Interactive Intelligent Systems*, 2:16:1–16:35.

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.

Rowan Zellers, Ari Holtzman, Elizabeth Clark, Lianhui Qin, Ali Farhadi, and Yejin Choi. 2021. Evaluating machines by their real-world language use. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

A Writing Interface

A screenshot of the interface is shown in Fig. 2.

B Data Details

We filter the WritingPrompts dataset to contain the first 500 words of all the stories; we do not use the prompts. After filtering, the dataset has 56,582 types and 55,785,118 tokens. 1.3% of the data is replaced with UNK. The original dataset can be found at https://github.com/pytorch/fairseq/tree/master/examples/stories.

Because the fusion model was originally trained to map from "prompt" to "story," we reconfigure the data and retrain the model to map from "story beginning" to "story end." To do this, we randomly split the stories at a newline and make the first portion of the story the "source" and the second portion the "target." In cases where there are no newlines within the text, we instead split on a space.

C Model Details

C.1 Fusion model

We train the fusion model with the data split in "source" and "target" as described in App. B, using the settings described at https://github.com/pytorch/fairseq/tree/master/examples/stories. We pretrain the model for 9 epochs before adding the fusion model and training for 14 epochs.

To generate, we assign an UNK penalty $= 10$ to suppress UNKs and use top-k sampling with $k = 40$.

C.2 GPT2 model

We finetune small GPT2 model on the WritingPrompts data using the code and settings at https://github.com/huggingface/transformers. We finetune the model for 3 epochs.

To generate, we use either top-k sampling with $k = 40$ (for GPT2 and TOP-K) or nucleus sampling with $p = 0.9$ (for NUCLEUS).

D Results

D.1 Edit Results by Suggestion

The full results, broken down by suggestion #, for edit distance, Jaccard similarity, and USER are in Table 7 (for FUSION vs. GPT2) and Table 8 (for NUCLEUS vs. TOP-K).

D.2 Likert-Scale Results

As shown in Fig. 3, the majority of participant responses are positive about their experience of writing with the CYOA, regardless of which model pair they were working with. The median score for almost all questions is 4 ("Agree").

The one exception is the median response for *The suggestions connected to what had happened in the story so far.* for FUSION vs. GPT2 is slightly lower–3 ("Neutral"). As hypothesized in §3, this is likely due to the higher degree of randomness in the FUSION-generated text.

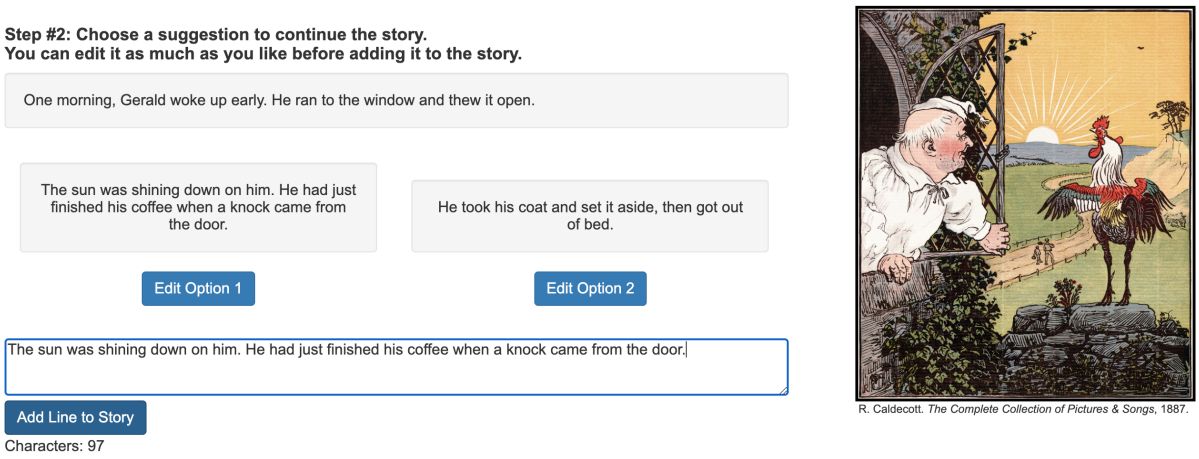

Figure 2: The story writing interface. The first box was the first turn of writing (author writing alone). In this case, Option 1 was generated with NUCLEUS sampling and Option 2 with TOP-K sampling. The writer has chosen Option 1, which shows up in the text box below and can now be edited before adding it to the story.

	OVERALL			FUSION			GPT2		
	ED (↓)	JS (↑)	USER (↑)	ED (↓)	JS (↑)	USER (↑)	ED (↓)	JS (↑)	USER (↑)
Total	32.27	57.85	67.97	37.61	51.13	60.69	29.49	61.35	71.77
Sugg. #1	24.10	65.11	73.77	25.76	62.09	70.17	23.57	66.05	74.90
Sugg. #2	27.26	62.15	71.71	34.19	48.09	57.79	24.22	68.31	77.81
Sugg. #3	34.40	55.96	67.09	36.51	52.20	63.74	33.15	58.17	69.06
Sugg. #4	31.52	59.74	70.89	36.95	56.40	65.55	28.32	61.71	74.05
Sugg. #5	44.08	46.29	56.39	48.13	41.71	50.61	41.03	49.73	60.72

Table 7: Edit distance (ED), Jaccard similarity (JS), and USER scores between the edited and the original generated suggestions overall, from FUSION, and from GPT2.

	OVERALL			NUCLEUS			TOP-K		
	ED (↓)	JS (↑)	USER (↑)	ED (↓)	JS (↑)	USER (↑)	ED (↓)	JS (↑)	USER (↑)
Total	35.74	52.21	62.86	34.65	53.64	63.64	36.69	50.96	62.18
Sugg. #1	32.40	54.12	65.56	26.98	61.37	71.31	36.28	48.93	61.44
Sugg. #2	37.13	49.24	60.22	37.71	47.86	58.25	36.62	50.44	61.95
Sugg. #3	32.93	55.56	65.04	34.79	51.43	60.58	31.31	59.16	68.92
Sugg. #4	33.35	55.37	66.60	34.38	57.31	66.77	32.45	53.68	66.45
Sugg. #5	42.89	46.75	56.90	38.23	51.28	62.24	47.84	41.94	51.24

Table 8: Edit distance (ED), Jaccard similarity (JS), and USER scores between the edited and the original generated suggestions overall, from NUCLEUS, and from TOP-K.

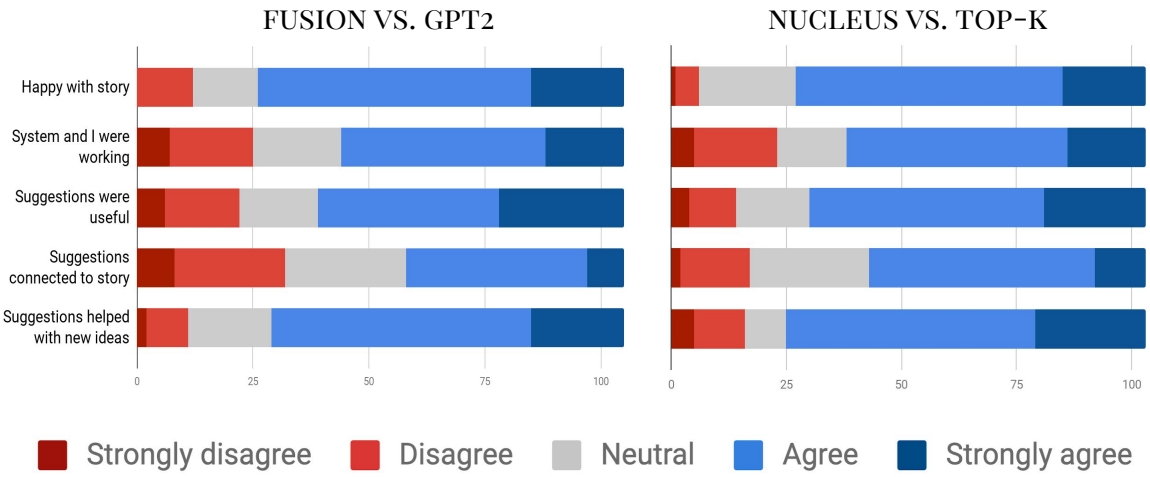

Figure 3: The Likert-scale results for FUSION vs. GPT2 and NUCLEUS vs. TOP-K.