Machine-in-the-Loop Rewriting for Creative Image Captioning

Vishakh Padmakumar
New York University
vishakh@nyu.edu

He He
New York University
hehe@cs.nyu.edu

Abstract

Machine-in-the-loop writing aims to enable humans to collaborate with models to effectively complete their writing tasks. Prior work in the creative domain has found that providing humans with a machine written draft or sentence level continuations has limited success as the generated text tends to deviate from the human's intentions. We train a rewriting model that, when prompted, modifies targeted spans of text within the user's original draft, enabling the human to retain control over the content while still taking advantage of the strengths of text generation models to introduce descriptive and figurative elements locally in the text. We evaluate the model on its ability to collaborate with humans on the task of creative image captioning through a user study on Amazon Mechanical Turk. Users report that the model is helpful and third-party evaluation shows that users write more descriptive and figurative captions on average in the collaborative setting compared to a baseline of the human completing the task alone.

1 Introduction

Creative writing tasks are challenging for humans because of their open-ended nature. Prior work shows that exposing authors to a collaborator that provides independent suggestions can spark new ideas (Garfield, 2008). This has motivated a line of work in machine-in-the-loop writing (Clark et al., 2018; Roemmele and Gordon, 2015; Samuel et al., 2016) where a human collaborates with a model to complete a writing task. However, recent work (Akoury et al., 2020; Clark et al., 2018) has shown that providing humans a draft generated by a machine is not very helpful because it may diverge from the direction envisioned by the author. As a result, very little machine generated text is ultimately retained. In this work, we aim to provide a form of interaction that gives human authors more control over the content while also assisting them to better express their own ideas (Roemmele and Gordon, 2015).

We focus on the setting where authors have a clear writing outline but would benefit from suggestions on wording or framing. To allow authors to control the content, we develop a machine-in-the-loop system called Creative Rewriting Assistant (CRA) which either rewrites a span of text or infills between two pieces of text when requested (Figure 1). Our CRA is a sequence-to-sequence model, building upon recent advances in controllable text generation (Shih et al., 2019; Ma et al., 2020; Kumar et al., 2020) and text infilling (Donahue et al., 2020; Fedus et al., 2018; Joshi et al., 2019; Shen et al., 2020). Specifically, the input is a sentence with text spans or blanks marked, and the output is a revised sentence where the marked span is replaced by a potentially more descriptive phrase. The CRA model is trained on a pseudo-parallel corpus of sentence pairs—a generic sentence and a more descriptive or figurative alternative created from existing datasets of creative text (Section 3.1). This process is detailed in Section 3.1 and we show that fine-tuning on the pseudo pairs results in a more helpful model in Section 4.2.

To evaluate our system, ideally we would use tasks like poem writing. However, it is challenging to control the content for fair comparison of different systems while allowing room for creativity. Therefore, we evaluate on a proxy task, creative image captioning (Chen et al., 2015), where the user is asked to write an expressive caption (a figurative or descriptive one as opposed to a literal one) for a given image. Importantly, we note that the purpose of the image is to ground the text so that different captions can be compared conditioned on similar content. The rewriting model does *not* take the image as an input, thus the content is largely controlled by the human author. We evaluate the system by hiring users on Amazon Mechanical Turk to perform the creative image captioning task

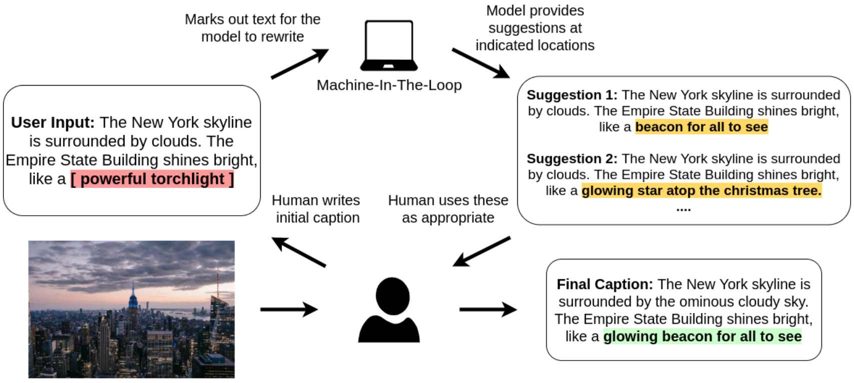

Marks out text for the model to rewrite
Machine-In-The-Loop
Model provides suggestions at indicated locations
User Input: The New York skyline is surrounded by clouds. The Empire State Building shines bright, like a [powerful torchlight]
Suggestion 1: The New York skyline is surrounded by clouds. The Empire State Building shines bright, like a beacon for all to see
Suggestion 2: The New York skyline is surrounded by clouds. The Empire State Building shines bright, like a glowing star atop the christmas tree.
....
Human writes initial caption
Human uses these as appropriate
Final Caption: The New York skyline is surrounded by the ominous cloudy sky. The Empire State Building shines bright, like a glowing beacon for all to see

Figure 1: Machine-in-the-loop rewriting for image captioning. The human is the central actor in the writing process and initiates interactions with the model by indicating what spans of text are to be rewritten. The model provides suggestions at these locations and the user chooses how to use them.

with and without model assistance. A third-party human evaluation (Section 4.3) shows that users writing in collaboration with CRA produce more creative captions than those writing alone, highlighting the end-to-end benefit of our machine-in-the-loop setup.

2 System Overview

Creative Image Captioning Task To allow for creativity while controlling the main content of the text for system comparison, we choose to situate the writing task visually in an image. Specifically, we adopt the creative image captioning task proposed by Chen et al. (2015). The goal for the user is to produce a figurative or descriptive caption for a given image. In our setup, the user is also given access to the model as they complete the task and we study the effect of this collaboration. Note that our model does not use the image for generation, which is analogous to real use cases where the model does not have access to the author's global writing plan, but instead provide improvements based on the local context.

Machine-in-the-loop system An overview of our system is illustrated in Figure 1. The user collaborates with the model to complete the writing task. We follow the user-initiative setup (Clark et al., 2018) where the model provides suggestions only when requested by the user. The system facilitates two types of editing: span rewriting and text infilling. Given a piece of text (written by the user), to request span rewriting, the user demarcates spans within the text that need to be rewritten. The model then edits the marked spans. For example, given *"The iPhone was a **[great piece of technology]** that changed the world"*, the model suggests the rewrite *"The iPhone was a **revolution in technology** that changed the world"*. To request text infilling, the user marks blanks to infill. For example, given *"The lion stalks the deer, a _____ in its element"*, the model infills *"The lion stalks the deer, a **predator** in its element"*. By limiting the edits to local spans, we alleviate the issue of deviating from the input content or generating incoherent text (Holtzman et al., 2019; Wang and Sennrich, 2020). For both rewriting and infilling, we sample multiple outputs from the model for users to consider. Then, they have the option to either accept a suggestion and continue writing, or reject them and retain their initial draft. This interactive process continues until the user is satisfied with the text and indicates the completion of the writing task.

3 Approach

3.1 Learning from Creative Text

The goal is to train a model capable of rewriting specific spans of an input sentence as indicated by a human user to assist them at the creative writing task. To this end, we need a dataset that contains sentence pairs where the target sentence is produced by replacing or inserting text spans in the source sentence to make it more descriptive or figurative. To our knowledge, there is no such dataset with paired revisions for creative writing; however, there are many datasets of text with annotated spans corresponding to literary devices (including metaphors, emotional cues, and figurative comparisons) in them. Therefore, we take the existing creative text as the target, and synthesize the source sentence by replacing the annotated spans

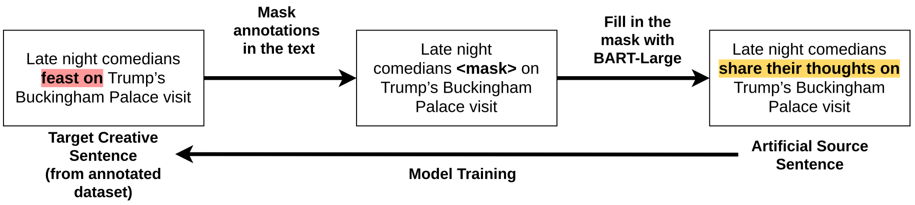

Late night comedians feast on Trump's Buckingham Palace visit
Mask annotations in the text
Late night comedians <mask> on Trump's Buckingham Palace visit
Fill in the mask with BART-Large
Late night comedians share their thoughts on Trump's Buckingham Palace visit
Target Creative Sentence (from annotated dataset)
Model Training
Artificial Source Sentence

Figure 2: Training data creation. The source sentence is created by masking out the annotated span and infilling it using BART-Large. The model is then trained to produce the creative sentence from the synthesized source sentence.

Source	**Domain**	**Annotation**	**Example**
Mohammad et al. (2016)	WordNet example sentences	Words that elicit emotion	I **attacked** the problem as soon as I was up.
Gordon et al. (2015)	Text collected by Mohler et al. (2015)	Metaphors in text	I will be out in the city today, feeling the vinous veinous thrust of blood, **the apple-red circulation of democracy**, its carnal knowledge without wisdom.
Bostan et al. (2020)	Headlines	Textual cues associated with emotion	Detention centers will **shock the conscience** of the nation.
Niculae and Danescu-Niculescu-Mizil (2014)	Product reviews	Figurative language	The stones appeared dull and almost opaque, **like black onyx**, with none of the sparkle you would expect from something called a diamond.
Steen et al. (2010)	News, fiction and academic text	Metaphors and personification	**Like a buzzard in the eyrie,** he would fly around.

Table 1: Sources of creative text and annotations used for creating training examples.

with infills from a generic language model, which presumably produces less creative text. The process of creating a paired corpus is shown in Figure 2. We start with a creative sentence from one of the datasets listed in Table 1, mask the annotated creative spans in it, and infill these using the pretrained BART model (Lewis et al., 2019) to generate the non-creative source sentence. For each pair from this pseudo-parallel corpus, we create one rewriting example by inserting the rewrite markers, `<replace>` and `</replace>`, at the beginning and the end of the rewritten span and one infilling example by replacing the span with a mask token, `<mask>`. We then train a sequence-to-sequence model (referred to as CRA) to generate the target creative sentence given the marked source sentence.

3.2 Learning from Interactions

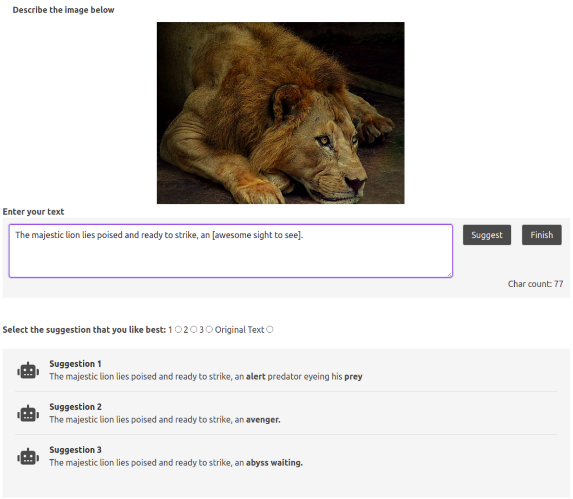
One important advantage of machine-in-the-loop systems is that they can be improved through usage given user feedback. Once users interact with CRA, we obtain their feedback on the suggestions, i.e. acceptance and rejection. We then use the feedback to update the model, so that it adapts to the observed user preference. Specifically, we create an example pair whenever the user indicates a preference for one sentence over another when presented with model suggestions. When the user accepts a suggestion, we take the accepted suggestion as the target (creative) sentence and the user's initial input as the source (non-creative) sentence. On the other hand, when the user rejects a suggestion, we take the rejected suggestion as the source and the user's initial input as the target. We then add these new pairs to a similar-sized subset of the original training examples (to prevent forgetting) and fine-tune the rewriting model on it.

4 Experiments

4.1 Setup

Crowdsourcing We hire users on Amazon Mechanical Turk to perform the creative image captioning task. A screenshot of our user interface is shown in Figure 3. Each user is presented with an image and asked to write a caption that is as figurative and/or descriptive as possible with at least 100 characters. The images were randomly sampled from the figurative subset of the Déjà Captions dataset (Chen et al., 2015), where the gold caption contains literary elements like metaphors and hyperbole. We ask users to request suggestions from the model at least twice while they are writing; however, they are free to ignore the suggestions. Users are instructed to use square brackets (as seen in Figure 1) to mark spans to be rewritten and underscore to indicate blanks to be infilled. They can edit the text with the model iteratively until they are satisfied with the caption. Once users submit the final caption, they are asked to complete a survey to rate the model. The survey questions are listed in Section 4.2 and the full task instructions are provided in Appendix A. The plan for the study was approved by the Institutional Review Board at NYU.

Model Details To train the Creative Rewriting Assistant (CRA) model, we first create the pseudo-parallel corpus as detailed in Section 3.1. Using creative sentences from all the sources from Table 1, we obtain a corpus containing 42,000 training pairs, 2,000 validation pairs, and 1,626 test pairs. The CRA model is trained by fine-tuning the `fairseq` (Ott et al., 2019) implementation of BART on this corpus. We train the model for 5 epochs with a learning rate of 3×10^{-5}. The learning rate was selected by perplexity on the validation set. We retain the recommended default values in `fairseq` for the hyperparameters of the Adam optimizer (Kingma and Ba, 2014), dropout rate, and learning rate scheduler.[1]


Describe the image below

Enter your text

The majestic lion lies poised and ready to strike, an [awesome sight to see].

Char count: 77

Select the suggestion that you like best: 1 ○ 2 ○ 3 ○ Original Text ○

Suggestion 1
The majestic lion lies poised and ready to strike, an **alert** predator eyeing his **prey**

Suggestion 2
The majestic lion lies poised and ready to strike, an **avenger.**

Suggestion 3
The majestic lion lies poised and ready to strike, an **abyss waiting.**

Figure 3: User interface. The user demarcates the span they want suggestions for in a text box and the model offers three suggestions for the user to pick from. This continues iteratively till the human is satisfied and submits the caption to finish the task.

4.2 Evaluating Suggestion Quality

To evaluate that fine-tuning on the pseudo-parallel corpus provides more helpful suggestions, we compare the performance of CRA against a pre-trained infilling language model, BART (Lewis et al., 2019). When BART is deployed in collaboration with a user, we mask the spans of text demarcated by them and infill the blanks with the model. For creative writing, we want a balance of diversity and fluency in model outputs. To choose the decoding scheme, we conducted a small internal pilot and observed a lack of diversity in beam search outputs. Thus, we use top-k sampling for both models, with k set to 10.

User Evaluation To evaluate the quality of the suggestions provided by CRA vs. the pre-trained BART baseline, we conduct A/B testing on 50 images randomly sampled from the Déjà Captions dataset. We ensure that each image has one caption from each model. Upon connecting to our server, each user is randomly assigned to work with one of the two models. So users working with both models are recruited from the same pool during the same time period, minimizing difference in performance due to individual users.

Once the task is completed, we ask the user to answer the following questions about the model on a Likert scale of 1 (worst) to 5 (best):

- How helpful were the model suggestions?
- How grammatically correct were the model suggestions?
- How satisfied were you with the final caption?

In addition, to analyze the effect of users' initial writing ability, we ask them to assess their writing skills:

- How would you rate your own writing ability on a scale of 1 to 5? 1—I don't have much experience with writing or am not too confident with the language, to 5—I have writing experience and/or have considerable proficiency with the English language.

Pre-trained BART (baseline) vs CRA The results from the survey are presented in Table 2. Each reported value is an average of scores given to the particular model by 50 users. We find that, on average, users find the CRA to be more helpful than BART. And this is despite the fact that in terms

[1] The beta values for the Adam optimizer are 0.9 and 0.999, the dropout rate is set to 0.1, and we use a polynomial decay learning rate scheduler with the weight decay parameter is set to 0.01. These were obtained from the released BART fine-tuning script.

of grammaticality, users report no significant difference between the two models. While BART is trained to perform coherent text infilling, by training the CRA on the pseudo-parallel creative corpus, we align the model suggestions better to the creative writing task resulting in a more helpful collaborator. Each reading from the survey is a single score given to the model by a user after the collaboration is complete. We also examine if the human evaluation tallies with automatic metrics we compute from the observed interactions. We compute the fraction of model suggestions accepted by the users in Table 3. Across 50 users, the CRA model has a higher acceptance rate than pre-trained BART, consistent with the helpfulness rating from users. In our setup, we also allow users to further edit model suggestions even after accepting them. So we want to measure if the text generated by the CRA is more useful comparted to the BART baseline in the case of an accepted suggestion. To quantify this positive model intervention, we calculated the Rouge-L recall scores of accepted model generations against the final caption submitted by the user. This value was 0.824 for the CRA model and 0.744 for the baseline pre-trained BART model so larger fractions of the CRA model suggestions was retained by users. Lastly, the total number of suggestions requested from BART is slightly higher, perhaps explained by its lower acceptance rate—users may persist with variants upon receiving unsatisfactory suggestions.

Question	BART	CRA
Helpfulness	2.23*	**3.06***
Grammaticality	2.96	**3.22**
Satisfaction	**3.69**	3.65

Table 2: User evaluation (50 user scores) of model performance for pre-trained BART baseline vs. CRA. Rows marked with an asterisk indicates statistically significant differences (p-value< 0.05 according to an independent samples t-test). Users find the CRA model to be more helpful by a statistically significant margin.

4.3 End-to-End System Evaluation

In the previous section, we observed that the CRA compared favourably to a pre-trained baseline model. We also want to evaluate the effectiveness of the collaboration in an end-to-end manner to see if the machine-in-the-loop setup helps users perform the task more effectively than users writing without model assistance (i.e. solo writing). To this end, we collect two captions each for a set of 100 images, one from the machine-in-the-loop setup (with CRA) and one from the solo writing setup. For solo writing, we recruit workers from the same pool as before (Amazon Mechanical Turk) and provide them the same instructions as in the machine-in-the-loop setup, except that all mentions of model assistance are removed. We then ask a 'third-party' human annotator (who did not participate in the writing task) to compare the two captions for each image. The annotator is asked to pick the more creative caption from the two: "Choose the better (more descriptive and/or figurative) caption for the image". The goal of this experiment is to identify if the machine-in-the-loop setup is more effective than solo writing so the wording of this criteria is kept consistent across both sets of writers as well as the third-party evaluators. For each pair of captions, we collect three annotations; the caption which obtains a majority vote is declared the winner.

	# request	# accepted	% accepted	Rouge-L
BART	**151**	37	24.5	0.744
CRA	141	**45**	**31.9**	**0.824**

Table 3: Interaction statistics (50 users) - How many suggestions were requested and accepted for the different models and the Rouge-L recall scores of accepted model generations against the final caption submitted by the user. Higher fractions of model suggestions are accepted when users collaborate with the CRA model. Also larger fractions of model generated text is retained in the final caption.

Does working with CRA improve the final caption? As shown in Table 4, the machine-in-the-loop setup (Human+CRA) won the majority vote 57 times out of 100. While prior work (Clark et al., 2018) in the creative domain, were unable to match the performance of the human only baseline using a less controllable assistant, here we show that CRA is able to collaborate well with human authors by allowing them to control the content and outperform the solo writing baseline. The improvement does not only come from direct edits of the text, some users also reported that considering different alternatives suggested by the model provided inspiration on how to improve the text (even though the suggestions are not accepted). We include representative positive and negative user feedback in Appendix B.

	Human+CRA	Human Only
# Majority Vote Wins	**57**	43

Table 4: Third-party evaluation of captions generated by our machine-in-the-loop setup (Human+CRA) vs. a human writing without assistance (Human Only). Wins were decided by a majority vote amongst 3 crowd workers. Users are able to write better captions with CRA.

4.4 Effect of Learning from User Interaction

The advantage with machine-in-the-loop systems is that once they are deployed, we can learn from user feedback to make them even more useful for new users. From our previous experiments, we have observed the user interactions with CRA (acceptance and rejection of the suggestions). As detailed in Section 3.2 we create a new set of paired examples that are used to further adapt the model to user preferences. The interactions from 50 users result in a dataset of 474 pairs of sentences. To ensure that the model does not suffer from forgetting, we also sampled 450 sentence pairs from the pseudo-parallel corpus. The initially trained CRA model is then further fine-tuned for 5 epochs on this dataset. We choose the learning rate of 3×10^{-6} using five-fold cross validation.[2]. We then evaluate this user-adapted CRA model against the initial CRA model on a fresh sample of 50 images, again following the A/B testing scheme from section 4.2.

Does user feedback improve the model? Our hypothesis is that adapting the model to user feedback should make it more helpful to new users. From Table 5, we see that the users do find the updated model to be slightly more helpful than the initial model on average; however, an independent samples t-test shows that this difference is not statistically significant (p-value $= 0.402$). A possible reason this happens is that the differing usage patterns of different users leads to the model getting noisy feedback and hence not significantly improving on the initial trained state. Thus a potential future direction is to explore adapting the model separately to each single user in a few-shot setting based on observing slightly longer interactions.

5 Analysis of Interactions

In Section 4, we verify that the CRA is helpful to users and the machine-in-the-loop system enables them to more effectively complete the task. We also want to better understand the cases when the model succeeds and fails at helping the users.

Question	Initial CRA	User-adapted CRA
Helpfulness	2.81	**3.05**
Grammaticality	2.87	**3.26**
Satisfaction	3.67	**3.78**

Table 5: User evaluation (50 user scores) of model performance for the initial model vs. the adapted model trained on user interactions. Users find the adapted model to be more helpful but the difference is not statistically significant.

5.1 When is CRA effective?

Which users find CRA more helpful? The motivation for a rewriting model was that human authors would benefit from a form of interaction where they retain more control over the written content (Roemmele and Gordon, 2015). But this relies on users having a coherent writing plan which might result in varying model effectiveness based on the skill level of the writer. To analyze the influence of users' inherent writing skill on model effectiveness, we put users into two groups based on their self-assessed writing ability (1 is the least skilled and 5 is the most skilled). A user is considered a *skilled writer* if they rate themselves higher than 3 and otherwise a *novice writer*. Out of the 50 users who interacted with CRA, 22 fall into the novice group and 28 fall into the skilled group. [3]

We show the ratings of helpfulness of CRA and the acceptance rate of model suggestions by user group in Table 6. We observe that skilled writers find the model more helpful, novice writers tend to request more suggestions and skilled writers accept a higher fraction of the provided suggestions. This is consistent with the idea that the skilled writers have a more clear plan thereby playing to the model's strengths. We would next like to understand the strengths of the CRA and if skilled users request a different profile of suggestions which informs the discrepancy in model effectiveness.

What kind of modifications is the CRA good at? We identify trends of edits the CRA is good at and provide illustrative examples for the same in table 7. We find that the usage patterns of skilled users align better with the model strengths.

[2] We again use the recommended hyperparameters for the Adam optimizer, dropout rate and learning rate scheduler

[3] As a sanity check, the self-reported skill level is consistent with the result from third party evaluation—72.72% of the captions written by skilled writers were judged as the winning caption by third-party annotators and this percentage drops to 46.42% among novice writers

	Novice	Skilled
Helpfulness	2.27*	**3.23***
# request	**3.04**	2.64
% accepted	29.8	**33.7**

Table 6: Model performance grouped by self-assessed writing skill: Average ratings of model helpfulness from the user survey, the average number of requests made to the model and the acceptance rate of received suggestions for both user groups. Rows marked with an asterisk indicates statistically significant differences (p-value < 0.05 on an independent samples t-test. Skilled writers find the model more helpful, request fewer suggestions but accept a higher percentage of them.

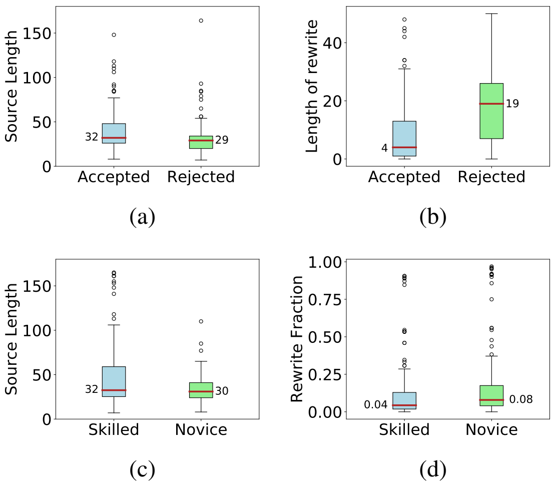


Figure 4: Analysis of interactions in terms of length of source sentences provided to the model (a, c) and rewritten spans in the generated text (b, d). We see that the model is more effective when given longer source context sentences (a) and generating shorter spans of text in the target sentences (b). Skilled writers find the model to be more effective (Table 6) because they play to the model's strengths by writing longer context sentences (c) and requesting shorter spans to be rewritten in them (d).

The model is more effective at editing longer sentences. A longer context allows the model to better infer the content and style of the requested suggestion so we expect that the model would be more effective at editing long sentences. In Figure 4a, we see that the accepted suggestions are more often generated from longer source sentences compared to rejected ones. From Figure 4c, we also see that skilled writers tend to write longer sentences (which CRA is good at); this partially explains why skilled users find the model to be more helpful. (Example 3 in Table 7)

Skilled writers request shorter rewrites which play to the model's strengths. One hypothesis to explain why the model is more helpful to the skilled writer group is that these users request suggestions at specific spans of text within longer sentences. Figure 4d shows us that though skilled writers tend to write longer sentences, they request smaller fractions of these sentences to be rewritten. (Examples 1 and 2 in Table 7)

Longer model rewrites get rejected more frequently. Our assumption is that users want to control the content of the caption. When the model rewrites a longer span and adds more new text to the draft, it is likely to diverge from the original content given by the user. We compare the length of text introduced into the draft (by rewriting or infilling) by the model among the accepted and rejected suggestions. From Figure 4b, we see that longer revisions are more likely to be rejected.

5.2 Error Analysis

To provide the full picture of our model we manually labelled 50 rejected suggestions to identify common error modes. Some illustrative examples from these are listed in Table 8. The most common failure case (21 out of the 50) is content drift: when the model is asked to replace key content words, sometimes the rewritten text changes the meaning of the user draft. This is seen in example 2 in Table 8, where the model changes "wave" to "motorized scooter"; while the suggestion is coherent, it changes the original meaning of the sentence. This is likely an artifact of how we create the pseudo-parallel corpus of training data: When BART performs infilling, the text introduced is not guaranteed to preserve the original content. The second common error type (14 out of the 50) is to copy the source text verbatim (example 3 in Table 8), especially when a long text span (e.g., a full sentence) is rewritten (Figure 4b), which is rare in our training data. Lastly, there is a small fraction of cases (9 out of the 50) when the model makes suggestions outside the desired demarcated region—this is often seen when the demarcated text spans two sentences and contains incoherent phrases (example 4 in Table 8). From Section 5.1, skilled writers tend to avoid these error modes by using the model to its strengths and requesting smaller fractions of longer sentences to be rewritten.

ID	**Demarcated Source Sentence**	**Accepted Suggestion**	**Edit**
1	A solemn woman place her mother's diary on a stepping stone her late father laid in the garden. The **[surrounding pale grass gently sway in the cold breeze]** while the woman ponders times of the past. Reminiscence now taking over and winter's beginning, the woman braces herself for dreary time to come.	A solemn woman place her mother's diary on a stepping stone her late father laid in the garden. The **pale grass gently danced and teased in the wind** while the woman pondered times of the past. Reminiscence now taking over and winter's beginning peaks, the woman braces herself for dreary time to come.	Figurative language
2	A man walks along the seashore with the horizon looming in the background. The dark clouds ____ as the sun sets for the day.	A man walks along the seashore with the horizon looming in the background. The dark clouds **slowly disperse** as the sun sets for the day.	Precise wording
3	The image represents wisdom and profound intelligence. It is the face of a man who lead the nation with grace and honor. It is a man who gave it all for the people and their well being. It is a statue that reflects the **[moral aspect of American people today]**.	The image represents wisdom and profound intelligence. It is the face of a man who lead the nation with grace and honor. It is a man who gave it all for the people and their well being. It is a statue that reflects **the moral aspect of a great man who lived and breathed the ideals of freedom and democracy**	Embellishment

Table 7: Examples where the model was successful in providing assistance. Bold spans in the source and target sentences are what marked by users and rewritten by the model, respectively.

ID	**Demarcated Source Sentence**	**Poor Suggestion**	**Error Type**
1	In front of a wall, a girl with blonde hair is on her hands who seems to be **[coming out of a magical door]**	In front of a wall, a girl with blonde hair is on her hands who seems to be **laughing out loud**.	Content drift
2	A child stands tall in a **[wave]** on the beach.	A child stands tall in a **motorized scooter** on the beach.	Content drift
3	I am witnessing a field of golden grain. Within that field a tall flower is blooming. That flower is not yet fully grown, yet its shades of purple are there and plainly visible. **[Overall, the image is nice. I do believe, however, that the quality of the image could be sharpened a bit.]**	I am witnessing a field of golden grain. Within that field a tall flower is blooming. That flower is not yet fully grown, yet its shades of purple are there and plainly visible. **Overall, the image is nice. I do believe, however, that the quality of the image could be sharpened a bit.**	Repeated the source
4	A beautiful **[sunset.A]** beautiful sunset in the ocean lighting up the sky in exotic colors.	A beautiful **sunset in the ocean lighting up the sky in exotic colors. A breathtaking view of nature at its best.**	Excessive editing

Table 8: Examples where the model was not successful in providing assistance. Bold spans in the source and target sentences are what marked by users and rewritten by the model, respectively.

6 Related Work

Collaborative writing. *Creative Help* (Roemmele and Gordon, 2015) looked at providing suggestions to writers by retrieving sentences from a corpus of stories. A follow-up study (Roemmele and Gordon, 2018) found that grammaticality and the presence of noun phrases in the text were indicative of helpful suggestions. Clark et al. (2018) evaluated a machine-in-the-loop setting on the tasks of story writing and slogan writing—providing sentence level suggestions for story writing and generating sentences from keywords for slogan writing. Akoury et al. (2020) developed models for machine-in-the-loop story writing and gave human writers access to a machine generated draft as a starting point. The finding that most machine text was removed or edited in Akoury et al. (2020) and the recommendation to allow for more human control in the process (Clark et al., 2018; Clark and Smith, 2021) motivated our approach to examine a rewrite based system for collaborative writing. Ito et al. (2020) demonstrated that a collaborative rewriting system could help non-native English speakers in revising fixed drafts of research papers. We extend this a step further by providing model access as users write from scratch allowing model interventions to guide the progress of the draft. Hence we evaluate the machine-in-the-loop collaboration in an end-to-end manner. A contemporary work (Coenen et al., 2021) frames collaborative writing as a conversation between a human and a dialog system. Rather than training the model to perform edits they select templated examples to provide as few shot context for each kind of edit.

Editing models. Transformer models have shown to be good at editing text in order to change the style (Shih et al., 2019; Krishna et al., 2020), debias text (Ma et al., 2020), post-edit translations (Grangier and Auli, 2018; Wang et al., 2020) and simplify text (Kumar et al., 2020). We differ from these by allowing humans to interactively choose where the rewrite is to be made. Additionally infilling literature (Donahue et al., 2020; Fedus et al., 2018; Joshi et al., 2019; Shen et al., 2020) has shown that we can train models to fill in blanks. We utilize this because it allows humans to direct the model to fill in parts of the text, differ in allowing any number of words in the blanks and extend the control to also allow for rewriting.

7 Conclusions and Future Work

Through this work, we train a Creative Rewriting Assistant (CRA) model that is able to effectively assist users to complete the task of creative image captioning. Our machine-in-the-loop rewriting setup allows for human users to control the content in text while taking advantage of the strengths of fine-tuned text generation models. But some of the limitations of our work point to directions of future research. The model is found to be more useful for skilled users so it remains to be explored how to better assist novice writers, perhaps a combination with autoregressive models or generating text from keywords. Additionally the main cause of failure is when the model suggestions alter the meaning of the user draft so another line of work is to balance the qualities of faithfulness and creativity in text generation assistance models.

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, Online. Association for Computational Linguistics.

Laura Ana Maria Bostan, Evgeny Kim, and Roman Klinger. 2020. GoodNewsEveryone: A corpus of news headlines annotated with emotions, semantic roles, and reader perception. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1554–1566, Marseille, France. European Language Resources Association.

Jianfu Chen, Polina Kuznetsova, David Warren, and Yejin Choi. 2015. Déjà image-captions: A corpus of expressive descriptions in repetition. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 504–514, Denver, Colorado. Association for Computational Linguistics.

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 *23rd International Conference on Intelligent User Interfaces*, pages 329–340.

Elizabeth Clark and Noah A. Smith. 2021. Choose your own adventure: Paired suggestions in collaborative writing for evaluating story generation models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3566–3575, Online. Association for Computational Linguistics.

Andy Coenen, Luke Davis, Daphne Ippolito, Emily Reif, and Ann Yuan. 2021. Wordcraft: a human-ai collaborative editor for story writing. *CoRR*, abs/2107.07430.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Association for Computational Linguistics (ACL)*.

William Fedus, Ian Goodfellow, and Andrew M. Dai. 2018. Maskgan: Better text generation via filling in the. In *International Conference on Learning Representations (ICLR)*.

Monica J Garfield. 2008. Creativity support systems. In *Handbook on Decision Support Systems 2*, pages 745–758. Springer.

Jonathan Gordon, Jerry Hobbs, Jonathan May, Michael Mohler, Fabrizio Morbini, Bryan Rink, Marc Tomlinson, and Suzanne Wertheim. 2015. A corpus of rich metaphor annotation. In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 56–66, Denver, Colorado. Association for Computational Linguistics.

David Grangier and Michael Auli. 2018. QuickEdit: Editing text & translations by crossing words out. 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 272–282, New Orleans, Louisiana. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Takumi Ito, Tatsuki Kuribayashi, Masatoshi Hidaka, Jun Suzuki, and Kentaro Inui. 2020. Langsmith: An interactive academic text revision system. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 216–226, Online. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. SpanBERT: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kalpesh Krishna, Josh Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Empirical Methods in Natural Language Processing*.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7918–7928, Online. Association for Computational Linguistics.

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 preprint arXiv:1910.13461*.

Xinyao Ma, Maarten Sap, Hannah Rashkin, and Yejin Choi. 2020. Powertransformer: Unsupervised controllable revision for biased language correction. In *EMNLP*.

Saif Mohammad, Ekaterina Shutova, and Peter Turney. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 23–33, Berlin, Germany. Association for Computational Linguistics.

Michael Mohler, Marc T Tomlinson, and Bryan Rink. 2015. Cross-lingual semantic generalization for the detection of metaphor. *Computational Linguistics and Intelligent Text Processing*.

Vlad Niculae and Cristian Danescu-Niculescu-Mizil. 2014. Brighter than gold: Figurative language in user generated comparisons. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2008–2018, Doha, Qatar. 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. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Melissa Roemmele and Andrew Gordon. 2018. Linguistic features of helpfulness in automated support for creative writing. In *Proceedings of the First Workshop on Storytelling*, pages 14–19, New Orleans, Louisiana. Association for Computational Linguistics.

Melissa Roemmele and Andrew S Gordon. 2015. Creative help: A story writing assistant. In *International Conference on Interactive Digital Storytelling*, pages 81–92. Springer.

Ben Samuel, Michael Mateas, and Noah Wardrip-Fruin. 2016. The design of writing buddy: a mixed-initiative approach towards computational story collaboration. In *International Conference on Interactive Digital Storytelling*, pages 388–396. Springer.

Tianxiao Shen, Victor Quach, Regina Barzilay, and Tommi Jaakkola. 2020. Blank language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5186–5198.

Yong-Siang Shih, Wei-Cheng Chang, and Yiming Yang. 2019. XL-Editor: Post-editing sentences with xlnet. *arXiv preprint arXiv:1910.10479*.

G.J. Steen, A.G. Dorst, J.B. Herrmann, A.A. Kaal, T. Krennmayr, and T. Pasma. 2010. *A method for linguistic metaphor identification. From MIP to MIPVU.* Number 14 in Converging Evidence in Language and Communication Research. John Benjamins.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online. Association for Computational Linguistics.

Qian Wang, Jiajun Zhang, Lemao Liu, Guoping Huang, and Chengqing Zong. 2020. Touch editing: A flexible one-time interaction approach for translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 1–11, Suzhou, China. Association for Computational Linguistics.

A HIT Instructions and Details

A.1 Instructions for crowdworkers completing the writing task

- Along with the first question in the survey is a link to the image captioning task. Navigate there. You will see a panel on the top left that shows you an image that you need to describe.
- You're free to interpret the image as you please—be as descriptive/figurative as possible.
- To help you with the same, we have a feature where you can highlight a part of your text with square brackets ('[', ']') and request targeted suggestions in that area. Please look at the accompanying examples on how to use it effectively.
- While writing we find that we are often able to provide content but to make the text more interesting is difficult, hopefully the assistant helps there. You will always have the option to reject the suggestions of the assistant and switch back to your original text. Bear in mind that the assistant isn't really great at guessing content words.
- To complete the task, continue editing until you are happy with the description. We require that you at least request suggestions from the assistant for a minimum of two times, even if you choose to reject the suggestions.

A.2 Instructions for crowdworkers evaluating the captions

- Choose the appropriate caption that best suits the image for the questions.
- A better caption is your subjective judgement, the rubrics to make the choice are that the caption is descriptive and/or figurative in its interpretation of the image (Refer the examples for further clarification).
- The explanation asked is supposed to be very brief. A single word of if you like it for being descriptive or interpretive will do.
- Relevance of the caption to the image is your subjective choice whether the caption appropriately represents what is in the image and is not just a catchy piece of text unrelated to the image.
- A caption that you deem irrelevant should never be the better caption, unless both are irrelevant.

B User Feedback from Mechanical Turk

We present some user feedback obtained from the task—these cover some of the positive and negative comments we received. The negative comments are representative of some of the issues we highlight in section 5.2

Positive

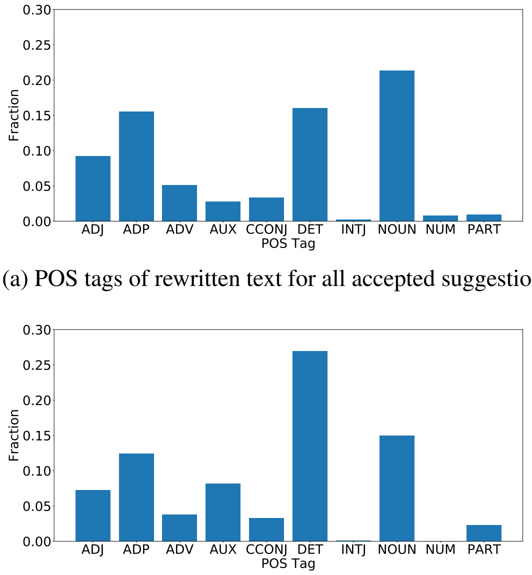
- I was impressed by how well this worked. I feel like my writing did improve by using the suggestions. At the very least it gave me good ideas.
- I got great suggestions that offered me words that I hadn't considered and fit even better than my own writing so I was pleased with the suggestions.
- I think everything was clear and straightforward and I enjoyed the interface.

Negative

- The suggestions were sometimes too far from the meaning of the original text so that they no longer made sense or were not grammatically correct.
- The instructions were fine, but the suggestions sure leave a lot to be desired. It replaced 'bright yellow' with red a couple of times.

C Profile of POS Tags in Accepted Suggestions

Accepted suggestions have more adjectives, adverbs and nouns. We analyze linguistic characteristics of accepted suggestions. Figure 5 shows the fraction of different POS tags in the revised span of accepted suggestions. Accepted suggestions tend to have a larger fraction of adverbs, adjectives and nouns whereas rejected suggestions have a large fraction of determiners. Prior work (Roemmele and Gordon, 2018) also observed that the presence of noun phrases in suggestions has a positive correlation with helpfulness.



(a) POS tags of rewritten text for all accepted suggestions.

(b) POS tags of rewritten text for all rejected suggestions.

Figure 5: Accepted suggestions tend to have more adjectives, adverbs and nouns and rejected suggestions tend to have higher fraction of determiners. The 10 most common POS tags were chosen to display in this figure.

D Ethical Considerations

Disproportionate assistance. One of the findings of our work was that the collaboration model discussed is more effective at assisting users who are already skilled at writing tasks. We noted in the paper that an important direction of future work is to develop systems that cater to the novice user group as well. An ethical consideration is that if such a system in its current state were deployed, it could lead to an increase in the disparity in performance between the two user groups. We believe that recording this observation is important as human-centered machine learning systems become more prevalent.

Appropriate remuneration for crowd workers. To complete the HIT on AMT, workers need to interact with the model a minimum of 2 times before submitting the caption—it is explicitly mentioned that they are free to reject the suggestions and accepting/rejecting suggestions has no bearing on the payment. From a small internal pilot (also confirmed with Mechanical Turk experiments) we estimate an average completion time to be 10 minutes with an additional 2 minutes to read the instructions, so the payment is set to $3 for the HIT (prorated to an hourly wage of $15). The estimated completion time for third-party evaluation was 1 minute so the payment was set to $0.25 per annotation (prorated to an hourly wage of $15).