Modeling Protagonist Emotions for Emotion-Aware Storytelling

Faeze Brahman
University of California
Santa Cruz
fbrahman@ucsc.edu

Snigdha Chaturvedi
University of North Carolina
Chapel Hill
snigdha@cs.unc.edu

Abstract

Emotions and their evolution play a central role in creating a captivating story. In this paper, we present the first study on modeling the emotional trajectory of the protagonist in neural storytelling. We design methods that generate stories that adhere to given story titles and desired *emotion arcs* for the protagonist. Our models include Emotion Supervision (EmoSup) and two Emotion-Reinforced (EmoRL) models. The EmoRL models use special rewards designed to regularize the story generation process through reinforcement learning. Our automatic and manual evaluations demonstrate that these models are significantly better at generating stories that follow the desired emotion arcs compared to baseline methods, without sacrificing story quality.

1 Introduction

Stories are an integral part of human culture. They allow us to express emotions, share knowledge, and to shape our perspective of the world (McKee, 2003). Stories are made interesting through emotions that connect the characters, their motivations, goals, and achievements (Vonnegut, 1981).

Cognitive scientists have pinpointed the central role of emotions in storytelling (Parkinson and Manstead, 1993; Hogan, 2011). Early automatic storytelling systems based on symbolic planning also showed that addressing character emotions for plot construction resulted in more diverse and interesting stories (Theune et al., 2004; Pérez y Pérez, 2007; Méndez et al., 2016). However, these studies were rule-based and limited to small-scale data. The advent of deep learning has shifted computational storytelling efforts towards neural methods (Martin et al., 2018; Yao et al., 2019). However, despite the broad recognition of its importance, neural story generation methods have not explored the modeling of emotional trajectory.

Title (input): Raw burger
Emotion arc (input): joy → anger → sadness
Story (output): Tom went to a burger place with his friends. He ordered a burger. When he got it , he noticed that it was raw. Tom yelled at the waiter for it being raw. He was really disappointed.

Figure 1: An example story generated by our model for a given title and emotion arc of the protagonist. Story segments are highlighted with the emotions the protagonist (Tom) experiences.

In this paper, we present the first study to take into account the emotional trajectory of the protagonist in neural story generation. Research in cognitive science has shown that while comprehending narratives, readers closely monitor the protagonist's emotional states (Komeda and Kusumi, 2006; Gernsbacher et al., 1992). However, emotions experienced by the protagonist might differ from the general emotions expressed in the story. For example, the general emotion of *"My boss was very angry and decided to fire me."* is *anger*, but the narrator's emotional reaction would be to feel *upset*. At any point in a story, we represent the protagonist's emotions using a set of *basic emotions*. The theory of basic emotions is well-accepted in psychology, but there is little consensus about the precise number of basic emotions. Plutchik (1982) proposed 8 primary emotions, and Ekman (1992) first proposed 7 and then changed to 6 basic emotions. Following recent theories (Jack et al., 2014; Gu et al., 2016), we choose *anger, fear, joy*, and *sadness*, to describe the protagonist's emotions. We additionally include *neutral* to account for cases with no strong emotions. We refer to these 5 emotions as *basic emotions*.

Moreover, emotions evolve within a narrative. For modeling the evolving emotions of the protagonist, we define an *emotion arc* for a story. Our definition is inspired by Prince's change-of-state

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 5277–5294,
November 16–20, 2020. ©2020 Association for Computational Linguistics

formalization (Prince, 2009), which asserts that stories are about change. According to this theory, a story has three components: a starting state; an ending state; and events that translate the starting into the ending state. Motivated by this, we define the *emotion arc* as a sequence of three *basic emotions* that describe the starting, body, and ending emotional states of the protagonist.

Given a story title and the emotion arc of the protagonist as inputs, our goal is to generate a story about the title that adheres to the given emotion arc. Fig. 1 shows an example story generated by our model, where the protagonist's emotion evolves from *joy* to *anger* and then *sadness*.

To address this problem, we present three models based on GPT-2 (Radford et al., 2019) that incorporate the protagonist's emotion arc as a controllable attribute while preserving content quality: an Emotion Supervision (EmoSup), and two Emotion-Reinforced (EmoRL) models based on reinforcement learning. The EmoRL models use two Emotion-Consistency rewards, EC-EM and EC-CLF. EC-EM uses semantically enhanced emotion matching to encourage the model to adhere to the given emotion arc. It infers the protagonist's emotions in the generated stories using *Commonsense Transformers*, COMET (Bosselut et al., 2019). EC-CLF achieves the same goal using a classifier to infer the protagonist's emotions.

In the absence of a training corpus of stories labeled with the protagonist's emotions, we automatically annotate a large-scale story corpus using COMET. Our automatic and manual evaluations show that our models can not only express the desired emotion arcs but also produce fluent and coherent stories. Our contributions are:

- We present the first study on modeling the *emotion arc* of the protagonist for neural story generation.
- We propose two Emotion-Consistency rewards designed to enforce the desired emotion arcs using reinforcement learning.
- We track the protagonist's emotions in a story (1) using a commonsense knowledge model based pipeline; and (2) through an emotion classifier trained using transfer learning from out-of-domain data.
- We empirically demonstrate that our models can effectively generate stories that follow the desired emotion arc. We also illustrate how these models can find novel applications.

2 Related Works

Early story generation systems relied on symbolic planning (Lebowitz, 1987; Pérez y Pérez and Sharples, 2001; Porteous and Cavazza, 2009; Riedl and Young, 2010) or case-based reasoning (Gervás et al., 2005). Although these systems could ensure long-term coherence, they could only operate in predefined domains and required manual engineering. These problems have been somewhat alleviated by recent seq2seq storytelling models (Roemmele, 2016; Jain et al., 2017), some of which are based on intermediate representations (Martin et al., 2018; Xu et al., 2018; Fan et al., 2018b; Yao et al., 2019; Fan et al., 2019).

Recent approaches have also used large-scale language models (LMs) based on Transformers (Vaswani et al., 2017), such as GPT-2 (Radford et al., 2019). Being trained on large amounts of data, these models can generate highly fluent text and find applications in story generation (Qin et al., 2019; Guan et al., 2020) and dialogue systems (Budzianowski and Vulić, 2019; Wolf et al., 2019). However, they lack the ability to dictate any auxiliary objective for the generated text, such as expressing specific attributes.

To address this, approaches such as conditional training or weighted decoding have been proposed to control different properties of the generated text such as sentiment, tense, speaker style, and length (Kikuchi et al., 2016; Hu et al., 2017; Ghazvininejad et al., 2017; Wang et al., 2017; Fan et al., 2018a). Tambwekar et al. (2019) use reinforcement learning for generating goal-driven story plots, which are sequences of event tuples. Dathathri et al. (2020) propose PPLM, which uses an attribute classifier to steer text generation without further training the LM.

More closely related to our work is generating automatic responses with a specific sentiment or emotion (Zhou et al., 2018; Huang et al., 2018; Zhou and Wang, 2018; Song et al., 2019). Modeling characters (Bamman et al. (2013), Bamman et al. (2014), Vala et al. (2015), Iyyer et al. (2016), Kim and Klinger (2019), Krishnan and Eisenstein (2015) *inter alia*) and sentiment trajectory (Chaturvedi et al. (2017), Chen et al. (2019) *inter alia*) has been shown to be useful for story understanding, in general. However, there are limited works on incorporating characters or sentiment for story generation. Previous work model characters but not sentiment (Clark et al., 2018; Liu et al., 2020). Weber et al. (2020) is a contemporary and unpublished work that incorporates sentiment while "filling in" a narrative. Peng et al. (2018) and Luo et al. (2019) control the overall sentiment for story ending generation. These works are limited to coarse-grained sentiments and/or only target the ending sentence. Instead, we model the emotional trajectory of the protagonist as the story progresses, which is more central to storytelling than the overall sentiment.

3 Emotion-aware Storytelling

We first explain how our models can track the protagonist's emotional trajectory (§3.1). We then define the problem statement (§3.2) and followed by an introduction to our base storytelling model (§3.3), which is used as the backbone of our three proposed models (§3.4 and §3.5)[1].

3.1 Tracking Protagonist's Emotions

In this work, we define the *protagonist* as the most frequently occurring character in a narrative (Morrow, 1985). Our two rewards for the EmoRL models (§3.5) need to track the protagonist's emotions to guide the generation. For this, we obtain their emotions at various stages in the story using one of the following two approaches:

Commonsense Transformer Our EC-EM reward uses a commonsense knowledge model to reason about the implicit emotional states of the protagonist. We use COMET (Bosselut et al., 2019), a knowledge base construction model trained on ATOMIC *if-then* knowledge triples (Sap et al., 2019). It contains information about everyday events and their causes and effects. Given an event and a relation, COMET can generate commonsense inferences about the relation. For tracking emotions, we use relations `xReact` and `oReact` that correspond to emotional reactions to events (more details on this in §4.1).

Emotion Classifier Our EC-CLF reward captures the protagonist's emotions using an emotion classifier. For this, we adapt the pre-trained BERT$_{large}$ for multi-label classification over 5 *basic emotions*: *anger, fear, joy, sadness,* and neutral. Following Devlin et al. (2019), we use a fully-connected layer over the final hidden representation corresponding to the special classification token (`[CLS]`). We train this classifier in two steps.

First, we train this classifier on a human-annotated dataset for emotion identification in tweets (Mohammad et al., 2018), consisting of $6,857$ tweets, with binary labels for 11 emotions, among which we only focus on our *basic emotions*. On this dataset, the classifier achieves better or comparable performance to state-of-the-art results (Kant et al., 2019) (see Appendix B.1 for detailed results).

Next, in order to identify the protagonist's emotions from a given story-text, we further fine-tune the classifier on story training data that is automatically annotated with the protagonist's emotions using the pipeline described in §4.1. To evaluate the classifier, we obtain manual annotations for the protagonist's emotions on Amazon Mechanical Turk for a subset of 50 randomly selected stories (250 sentences) from our story corpus. Each sentence was annotated by 3 judges. Workers agreed with our emotion classifier 70% of the time (random agreement would be 20%). See Appendix B.2 for more details about these annotations.

3.2 Problem Statement

We formulate the emotion-aware storytelling task as follows: given a story title as a sequence of tokens $\boldsymbol{t}$=$\{t_1, t_2, ..., t_m\}$, and an emotion arc for the protagonist as a sequence of *basic emotions* $\boldsymbol{a}$=$\{e_1, e_2, e_3\}$, the task is to generate a story as a sequence of tokens $\boldsymbol{y}$=$\{y_1, y_2, ..., y_n\}$ that adheres to the title and emotion arc.

3.3 Transformer-based Storytelling Model

Our models are built upon a base storytelling model that can generate a story consistent with a given prompt (e.g., title). We choose GPT-2 (medium) (Radford et al., 2019) because our initial experiments demonstrated that it outperforms other state-of-the-art story generation models, in general (§5.1). GPT-2 uses multiple Transformer blocks of multi-head self-attention and fully connected layers (the left box in Fig. 2). Since it was trained on a broad range of domains, we fine-tune it on a dataset of stories (§4.1) by minimizing the negative conditional log-likelihood:

$$\begin{aligned} \mathcal{L}_{ML} &= -\sum_{i=m}^{m+n} \log p(y_i|y_{<i}, \boldsymbol{t}) \\ p(y_i|y_{<i}, \boldsymbol{t}) &= \text{softmax}(h_i^L W^T) \\ h_i^l &= \text{block}(h_{<i}^{l-1}), l \in [1, L] \\ h_i^0 &= W_i + P_i \end{aligned} \tag{1}$$

[1] Code at: https://github.com/fabrahman/Emo-Aware-Storytelling

where m and n denote the number of tokens in the title and story respectively. h_i^l is the l-th layer's output at the i-th position computed through transformer block with the masked multi-head self attention mechanism, and h_i^0 is a summation of *token embedding* W_i and *position embedding* P_i for the i-th token. $y_{<i}$ indicates left context.

3.4 Emotion Supervision (EmoSup) Model

The underlying idea behind our Emotion Supervision (EmoSup) model is to provide the emotion arc as an additional input similar to conditional training (Fan et al., 2018a; Kikuchi et al., 2016). Specifically, each title has the corresponding emotion arc prepended at the beginning, separated by a delimiter token (<$>). This way, emotion arcs receive special treatment (Kobus et al., 2017), since they are propagated to all of the story and the model learns to maximize $p(y_i|y_{<i}, \boldsymbol{t}, \boldsymbol{a})$.

3.5 Emotion-Reinforced (EmoRL) Models

The emotion arc guides the generation in EmoSup as an initial input. However, we want to continually supervise the model *during* the generation process. This motivates us to use a reinforcement learning framework. To deal with exposure bias, many previous works have optimized the evaluation measures (e.g., BLEU, ROUGE, CIDEr) as rewards (Rennie et al., 2017; Paulus et al., 2018). Here, we propose two Emotion Consistency rewards, Ec-Em and Ec-Clf, which optimize adherence to the desired emotion arc.

Ec-Em Reward This reward quantifies the alignment of the emotion arc of the generated story to the desired arc using the commonsense knowledge model, COMET. For an N-sentence-long generated story, we use COMET to obtain the protagonist's emotional reaction for each sentence, resulting in a sequence of emotion-phrases $\boldsymbol{a}^g$=$\{g_1, g_2, ..., g_N\}$[2]. We then define the reward as a modified Levenshtein distance (Levenshtein, 1966) between the generated reactions $\boldsymbol{a}^g$ and the desired emotion arc $\boldsymbol{a}^*$=$\{e_1, e_2, e_3\}$. This modification allows only two operations: (1) Deletion of an emotion-phrase (in $\boldsymbol{a}^g$), and (2) Replacement of an emotion-phrase with a basic emotion at a cost proportional to semantic similarity between the two (e.g., *happy to help* and *joy*). Semantic similarities are computed using cosine similarity between the averaged GloVe embeddings (Pennington et al., 2014) [3]. The reward is defined as:

$$r_{em} = \text{lev}(\boldsymbol{a}^g, \boldsymbol{a}^*) \qquad (2)$$

where lev denotes the modified Levenshtein distance. We refer to the model that uses this reward as **Rl-Em**.

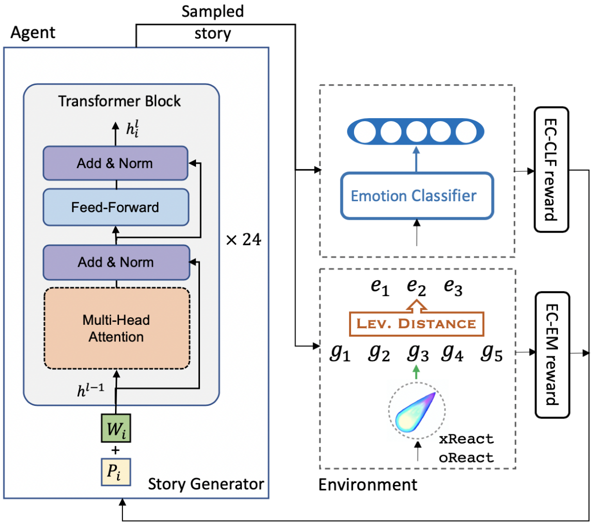


Figure 2: Transformer block architecture (left) and emotion-reinforced storytelling framework (right)

Ec-Clf Reward This reward infers the protagonist's emotions in a given text using our emotion classifier (§3.1). We first divide the generated story into segments: beginning, body, and ending[4]. Then, for each segment, we use the classifier to obtain the probability of the desired emotion. The reward is defined as the probabilities of the desired emotions averaged across the segments:

$$r_{clf} = \frac{1}{k} \sum_{j=1}^{k} p_{\text{clf}}(e_j^* | \boldsymbol{x}_j) \qquad (3)$$

where k is the number of tokens in the emotion arc (here, k=3), and e_j^* denotes the desired emotion for j-th segment $\boldsymbol{x}_j$. We refer to the model that uses this reward as **Rl-Clf**.

Policy Gradient For training, we use the REINFORCE algorithm (Williams, 1992) to learn a generation policy p_θ of the storytelling model with

[2] N=5 for our dataset. Also, COMET's outputs, g_is, are phrases representing emotional reactions. Details on obtaining emotional reactions during training are provided in Appendix A.1.

[3] We experimented with contextualized representations from pre-trained BERT but opted for static embeddings because (1) they yielded better results, and (2) enabled us to focus on semantics of the individual emotion tokens/phrases rather than context-sensitive representations.

[4] We generate 5-sentence long stories similar to our training corpus and segment them into the beginning, body, and ending in 1:3:1 ratio.

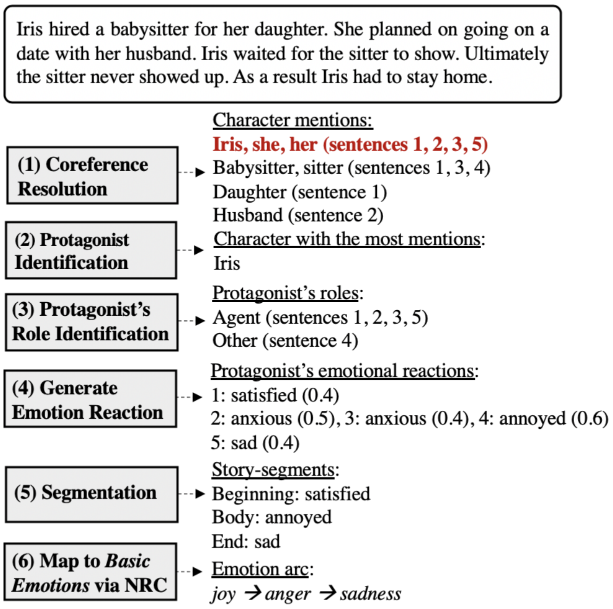

Iris hired a babysitter for her daughter. She planned on going on a date with her husband. Iris waited for the sitter to show. Ultimately the sitter never showed up. As a result Iris had to stay home.
(1) Coreference Resolution
Character mentions:
Iris, she, her (sentences 1, 2, 3, 5)
Babysitter, sitter (sentences 1, 3, 4)
Daughter (sentence 1)
Husband (sentence 2)
(2) Protagonist Identification
Character with the most mentions:
Iris
(3) Protagonist's Role Identification
Protagonist's roles:
Agent (sentences 1, 2, 3, 5)
Other (sentence 4)
(4) Generate Emotion Reaction
Protagonist's emotional reactions:
1: satisfied (0.4)
2: anxious (0.5), 3: anxious (0.4), 4: annoyed (0.6)
5: sad (0.4)
(5) Segmentation
Story-segments:
Beginning: satisfied
Body: annoyed
End: sad
(6) Map to Basic Emotions via NRC
Emotion arc:
joy → anger → sadness

Figure 3: Annotation pipeline for *emotion arc*.

parameters θ. Here, the model generates a sample story $\boldsymbol{y}^s$ from the model's output distribution, and the goal is to minimize the negative expected reward, which is approximated by:

$$\mathcal{L}_{RL} = -(r(\boldsymbol{y}^s) - r(\hat{\boldsymbol{y}})) \sum_{i=k+m}^{k+m+n} \log p_\theta(y_i^s | y_{<i}^s) \qquad (4)$$

We follow the self-critical training approach (Rennie et al., 2017), and take the reward of the greedily decoded story $\hat{\boldsymbol{y}}$ as the baseline reward $(r(\hat{\boldsymbol{y}}))$. This ensures that with better exploration, the model learns to generate stories $\boldsymbol{y^s}$ with higher rewards than the baseline $\hat{\boldsymbol{y}}$. Optimizing only with the RL loss mentioned above using the emotion-consistency rewards may increase the expected rewards, but at the cost of fluency and readability of the generated story. Therefore, we optimize the following mixed loss (Paulus et al., 2018):

$$\mathcal{L}_{mixed} = \gamma \mathcal{L}_{RL} + (1 - \gamma)\mathcal{L}_{ML} \qquad (5)$$

where γ is a hyper-parameter balancing the two loss functions. Our emotion-reinforced storytelling framework is depicted in Fig. 2.

4 Experimental Setup

4.1 Dataset and Annotation Pipeline

We use the *ROCStories* corpus (Mostafazadeh et al., 2016) for our experiments. It contains $98,162$ five-sentence stories, designed to have a clear beginning and ending, thus making it a good choice for our emotion-aware storytelling task. We held out 10% of the stories each for validation and test sets, respectively.

For training our models, we need stories annotated with emotion arcs of the protagonists. We annotated the stories in our dataset automatically using the multi-step annotation pipeline shown in Fig. 3. In step 1, we identify all characters and their mentions in a story using coreference resolution. In step 2, we identify the character with the most mentions as the *protagonist* (e.g., 'Iris' who is mentioned in 4 sentences). Then, in step 3, in each sentence of the story, we identify the protagonist's role as *Agent* or *Other* using its dependency parse[5]. The protagonist is an *Agent* if he/she is the subject of the main verb in the sentence and *Other* otherwise (e.g., Iris's role is *Other* in sentence 4 and *Agent* in all other sentences). Next, in step 4, we obtain the emotional reaction of the protagonist in each sentence using COMET. Given a context, c, and relation type, r, COMET can yield the emotional reaction of the *Agent* (r=`xReact`) and *Others* (r=`oReact`). Depending on the protagonist's role in the sentence, we use the appropriate relation to get their emotional reaction, g, and COMET's confidence in the prediction, φ_g. In sentences without an explicit mention of the protagonist, his/her role is assigned as *Other*, and we use `oReact` since the event in that sentence will affect all characters of the story, including the protagonist (e.g., sentence 4 in Fig. 3).

Step 4 gives the protagonist's emotions for each sentence of the story, but the emotion arc has to represent them for the three segments: beginning, body, and end. The stories in our corpus are 5-sentence long, and following previous work on this corpus (Chaturvedi et al., 2017), we segment them in 1:3:1 ratio. For the protagonist's emotion in the body (middle 3 sentences), we take the emotion of the sentence in which COMET was most confident (e.g., 'annoyed' for the body of the running example in Step 5).

Note that since COMET's outputs, gs, are open-ended emotion-phrases, in step 6, we need to map these phrases to one of the 5 *basic emotions* using NRC Affect Intensity Lexicon (Mohammad, 2018). The lexicon is a list of words with their real-valued intensities for 4 non-*neutral* basic emotions. We represent the likelihood of g getting mapped to

[5] We use AllenNLP (Gardner et al., 2018) for coreference resolution and dependency parsing: https://github.com/allenai/allennlp

each of the basic emotions, e, as $score_e$(e). For mapping, we first tokenize, lemmatize, and filter stop words from g. Then we find exact matches of g's tokens to words in the lexicon (along with the match-intensities). For each match, we increase $score_g$(e) by the match-intensities. Finally, g is mapped to the basic emotion with the maximum score. An emotion-phrase with no matching tokens is mapped to *neutral*.

Note that we also experimented with the emotional reactions generated by COMET to constitute the *emotion arc* without mapping them to *basic emotions*. However, with more than 500 unique emotional reactions, the space of possible arcs became too large with too few training examples for each which prevented the models from effectively learning the pattern. The smaller set of *basic emotions* also made it more natural and manageable for the user to provide a desired emotion arc as input.

4.2 Implementation Details

We follow the training and inference settings of medium-size GPT-2 as in Radford et al. (2019) (for completion, we provide full details in Appendix A.2). Our models are implemented with the Texar toolkit (Hu et al., 2019).

4.3 Evaluation Measures

Automatic We adopt several automatic measures to evaluate the generated stories both on content quality and emotion faithfulness.

For evaluating the content quality, we use the following measures: (1) **Perplexity** as an indicator of fluency. A smaller value is better[6]. (2) **BLEU**, which is based on n-gram overlaps (Papineni et al., 2002). Following Guan et al. (2020), since BLEU scores become extremely low for large n, we used n=$1, 2$. (3) **Distinct-n** (with n=$1, 2, 3$) measure the percentage of unique n-grams (Li et al., 2016). A high ratio indicates a high level of lexical diversity. (4) **Repetition-4** is the percentage of generated stories that repeat at least one 4-gram (Shao et al., 2019). A high value indicates redundancy in the generated text.

For evaluating the emotion faithfulness of a generated story, we adapt lexical measures (1) **Seg-word** and (2) **Arc-word** (Song et al., 2019). Given a desired emotion arc for a story, Seg-word is the percentage of the story's segments that contain emotion words corresponding to desired emotion. Correspondingly, Arc-word for a story is a binary score indicating if *all* of its segments contain emotion words corresponding to the desired emotions. We also define (3) **Seg-acc** and (4) **Arc-acc** for a generated story. Seg-acc for the story is the fraction of generated segments for which the emotion (as determined by the emotion classifier) exactly matches the desired emotion. Similarly, Arc-acc for a story is a binary score indicating if its emotion arc (as determined by the emotion classifier) exactly matches the desired emotion arc. We also use the reward functions, (5) **EC-CLF** and (6) **EC-EM**, to score a generated story. For all these measures, we report averaged scores across all stories generated by a model.

Manual We also conduct a manual evaluation of generated stories using Amazon Mechanical Turk. Following Song et al. (2019), workers are asked to evaluate pair of stories on a 0-3 scale (3 being very good) from two different perspectives: (1) **emotion faithfulness** to assess whether it follows the desired emotion arc for the protagonist, and (2) **content quality** to indicate whether a story is fluent, logically coherent, and on-topic (related to the given title). Workers were also asked to indicate their **overall preference** by choosing the better story of the two while considering both aspects, or indicate that they are of equal quality. More details about evaluation measures are provided in Appendix A.3.

5 Results and Discussion

We first describe our experiments on choosing the base storytelling model (§5.1) followed by evaluation of the proposed models (§5.2).

5.1 Base Storytelling Model Results

As noted before, our models build upon a base storytelling model (GPT-2). We compared GPT-2 with the following state-of-the-art story generation models, given the title as input. (1) *S2S* (Bahdanau et al., 2015), (2) *ConvS2S* (Gehring et al., 2017), (3) *Fusion* (Fan et al., 2018b), (4) *Plan&Write* (Yao et al., 2019), (5) *Decom-Event* (Fan et al., 2019).

Using various evaluation measures described earlier, our experiments showed that fine-tuned GPT-2 outperforms all baselines, indicating that it can serve as good base storytelling model. This is in-line with the observations made in Guan et al.

[6] For comparison, we compute *word-level* perplexity for GPT-2 based models. That is, we normalize the total negative log probability by the number of *word*-level tokens, not the number of BPE tokens.

Models	PPL (↓)	BLEU-1 (↑)	BLEU-2 (↑)	Dist-1 (↑)	Dist-2 (↑)	Dist-3 (↑)	Repet-4 (↓)
Fusion + Emo	24.02	21.10	2.61	66.18	90.88	96.91	23.30
Plan&Write + Emo	17.43	22.46	3.03	66.32	90.47	95.59	28.61
PPLM-3	–	20.36	2.37	71.37	93.90	98.19	13.36
PPLM-5	–	20.61	2.47	71.47	93.99	98.21	14.02
GPT-2 + FT*	12.16	22.68	3.10	72.93	94.24	98.28	12.10
EmoSup	**11.10**	22.70	3.23	71.44	93.75	98.10	13.94
RL-EM	11.98	22.52	3.15	**73.32**	**94.76**	**98.56**	**10.09**
RL-CLF	11.31	**22.78**	**3.26**	71.16	93.65	98.05	13.34

Models	Arc-word	Seg-word	Arc-acc	Seg-acc	EC-CLF	EC-EM
Fusion + Emo	6.32	38.89	29.89	62.59	60.06	73.59
Plan&Write + Emo	5.61	32.98	26.38	60.99	58.13	72.46
PPLM-3	7.11	36.10	23.97	57.01	56.02	73.19
PPLM-5	7.74	37.64	27.30	60.60	59.51	74.43
GPT-2 + FT*	4.46	33.28	17.32	48.69	47.93	69.00
EmoSup	7.33	40.86	31.25	64.26	62.88	74.10
RL-EM	8.85	43.77	33.56	65.13	63.93	**76.59**
RL-CLF	**10.14**	**45.42**	**37.58**	**68.90**	**67.55**	75.87

Table 1: Automatic evaluation of content quality (top) and emotion faithfulness (bottom). For content quality, RL-CLF and RL-EM outperform all baselines for BLEU and diversity/repetition scores respectively ($p < 0.05$). For emotion faithfulness, RL-CLF outperforms all baselines ($p < 0.05$). * indicates absence of emotion arc as input.

(2020). Since this is not our focus, we report full results and details in Appendix B.3.

5.2 Emotion-Aware Storytelling Results

Baselines We use the following baselines in our experiments: (1) GPT-2+FT, our base GPT-2 model fine-tuned on the ROCStories corpus, for which emotion arcs are not provided as inputs; (2) *Fusion+Emo* and (3) *Plan&Write+Emo*, which are two of the strongest storytelling baselines (we prepended emotion arcs to titles in our experiments); and (4) *PPLM* (Dathathri et al., 2020) which can be extended to accept emotion arcs for controlling story generation. *PPLM-3* and *PPLM-5* indicate 3 and 5 iterations respectively[7].

Automatic Evaluation For automatic evaluation, we used the titles and automatically extracted emotion arcs of the stories in our test set as input.

The evaluation results on content quality are shown in the top half of Table 1. Interestingly, even though the proposed models only aim to control emotion arc, they outperform GPT-2+FT on perplexity indicating better fluency. Among the proposed models, EmoSup obtains the best perplexity score mainly because that is what its loss function optimizes (as opposed to the mixed loss in EmoRL models). Overall, all of our proposed models outperform all baselines. In particular, RL-CLF has the highest BLEU scores, and RL-EM has the highest diversity and lowest repetition scores. All improvements over baselines are statistically significant (approximate randomization (Noreen, 1989), $p < 0.05$).

The evaluation results on emotion faithfulness are shown in the bottom half of Table 1. We see that, as expected, all models outperform GPT-2+FT, which is not provided the emotion arcs as inputs. Our proposed models also achieve significant improvements over all baselines (app. randomization, $p < 0.05$). In particular, RL-CLF achieves the best performance on almost all measures.

We also compare various models on the most common emotion arcs in our corpus. Fig. 4 shows the Arc-acc of various models on the 10 most common arcs. We can see that all models perform very well on "joy → joy → joy" as compared to other emotion arcs. This is because this is the most common emotion arc (34% of the training data) in our corpus, which results in availability of significant number of training examples for this arc. Nevertheless, for all arcs, RL-CLF consistently outperforms all other models indicating a better control over the desired emotion arc.

These results indicate that while all proposed

[7] We use the HuggingFace implementation: `https://github.com/huggingface/transformers/tree/master/examples/text-generation/pplm`. For a fair comparison, we used GPT-2 fine-tuned on stories as the underlying generation model.

	Specific Criteria		**Overall Preference**
	Emotion Faithfulness	Content Quality	Better / Worse / Tie (%)
RL-CLF vs. RL-EM	+0.20 (1.77 ± 0.91, 1.57 ± 0.95)	+0.15 (1.73 ± 0.78, 1.58 ± 0.85)	**52.33** / 38.00 / 9.66
RL-CLF vs. GPT-2+FT	+0.76 (2.24 ± 0.80, 1.48 ± 0.98)	+0.25 (2.25 ± 0.82, 2.00 ± 0.88)	**60.00** / 22.00 / 18.00
RL-CLF vs. EmoSup	+0.28 (1.97 ± 1.00, 1.69 ± 1.05)	+0.14 (1.93 ± 0.94, 1.79 ± 0.97)	**50.33** / 34.00 / 15.66
RL-CLF vs. PPLM-5	+0.48 (2.10 ± 0.86, 1.62 ± 0.94)	+0.34 (2.21 ± 0.90, 1.87 ± 0.96)	**61.00** / 25.66 / 13.33

Table 2: Manual evaluation results. For each criteria, we report the average improvements as well as the absolute scores for the two models, separated by a comma. RL-CLF is preferred over other methods ($p < 0.05$).

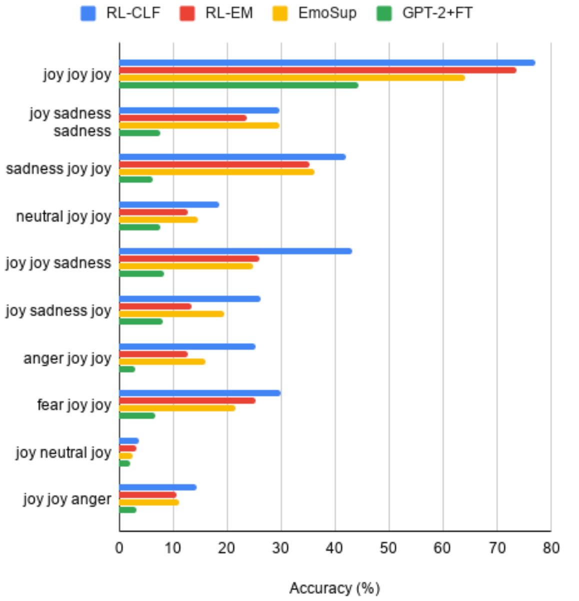


Figure 4: Arc-acc of various models on the 10 most common arcs in our corpus. RL-CLF outperforms other models for almost all arcs.

models can control the emotion arc of generated stories, RL-CLF achieves a good balance between both content and emotion quality.

Manual Evaluation Since concerns have been raised about automatic evaluation of language generation, we also conduct a manual evaluation on Amazon Mechanical Turk. For this, we randomly sampled titles and emotion arcs of 100 instances from our test set, and generated stories using the models being evaluated. We compared five models, and so overall, there were 500 stories. We conduct pairwise comparisons of generated stories, and each pair was evaluated by 3 judges. Table 2 reports the average improvements as well as absolute scores for emotion faithfulness and content quality (evaluated independently) and also the overall preference of the judges. We first compare our two EmoRL models (Row 1). We see that RL-CLF improves over RL-EM on both emotion faithfulness and content quality. Overall, it is judged to be better than RL-EM 52.33% of the time and worse in only 38.00% cases. We then compare the better of the two, RL-CLF, with the uncontrolled GPT-2+FT (Row 2). We see that on average, RL-CLF model is not only better at adhering to the emotion arc by +0.76 points but also generates better content (improvement of +0.25 points) and its stories are preferred 60% of the times by humans. We observe similar results for comparison with EmoSup and PPLM-5. All improvements are statistically significant (app. randomization, $p < 0.05$).

Title: fire injuries

joy - sadness - joy	My friends and I went camping this summer. We got in my van and went to the woods. We decided to light a campfire. While driving around , our tire popped and the fire started. We had to call the fire department for help and they were able to put out the fire.
sadness - sadness - joy	The fire department was called to a house in the woods. The house was engulfed in flames. There were two people inside. One person was taken to the hospital by air ambulance. Luckily, the other person was treated for non-life threatening injuries.

Title: dance

fear - joy - joy	Kelly was worried about her dance recital. She had practiced her dance for weeks. She decided to try out for the school's dance team. Kelly was nervous but knew she could do well. She was so excited she gave her best impression!
sadness - joy - joy	I was very depressed. I went to a dance class with a friend of mine. We tried out some different moves. We got stuck dancing for a long time. The next day I tried out some new moves and got a standing ovation.

Table 3: For a given title, our model can generate different stories for different emotion arcs. Story segments with corresponding emotions are highlighted.

Case Studies Since the proposed models can generate stories conditioned on the protagonist's *emotion arc*, they can be used to unfold a story in diverse situations for a given title. We demonstrate this capability in Table 3. It shows two examples where for the same title, our model (RL-CLF here) can generate stories that follow different emotion

Input story: There was a girl called Cinderella who did all the work for her mean, ugly step sisters. One day, she got an invitation to go to a ball at the palace. A fairy Godmother appeared an made her a beautiful dress and a lovely carriage. After Cinderella left the ball, the prince looked everywhere for her. He eventually found her and they got married and lived happily ever after.
Automatically extracted emotion arc: sadness → joy → joy
Input Title: The wedding
Output story: Ryan had been feeling really lonely lately. He decided he needed a way to make a friend. He decided to go to a wedding. When he got there he met a beautiful girl. Ryan had made a new friend that day !

Table 4: Given a story, our model can generate another story with similar emotion arc.

arcs for the protagonists.

Alternatively, given a story, the models can also be used to generate another story with a similar emotion arc (after automatically extracting the protagonist's emotion arc in the given story using the pipeline described in §4.1). For example, in Table 4 we show how RL-CLF can be used to generate a novel story in which the protagonist follows the same emotion arc as in the 'Cinderella' story. Note that the goal here is not necessarily to generate a similar narrative but a story that follows the same emotional trajectory.

We provide more qualitative examples in Appendix Figure 8.

6 Conclusion

In this paper, we proposed the emotion-aware storytelling task for modeling the emotion arc of the protagonist. To this goal, we designed two emotion-consistency rewards using a commonsense transformer and an emotion classifier. Experiments demonstrated that our approach improved both content quality and emotion faithfulness of the generated stories. We also presented two case studies, which show interesting use cases of our model. In general, such models can have educational applications by enabling children to explore creative writing at an early age and addressing the literary learning needs of learners with disabilities.

This paper is a step towards future research directions on planning emotional trajectory while generating stories. Using commonsense inferences about the effect of the events on emotional states of various characters of the story has the potential of generating more coherent, realistic, and engaging stories. In this work, we focused only on the protagonist, but future works can explore modeling motivations, goals, achievements, and emotional trajectory of all characters. Our approach is general and provides a blueprint for similar works going forward and can be used outside emotion-aware storytelling, e.g., for generating other emotional content or text with other attributes or properties.

The various assumptions and choices made in this paper and the specific characteristics of the dataset we chose can introduce biases and errors. For example, COMET is a discourse-agnostic model, and separately extracting emotional reactions for each sentence may fail to maintain emotional consistency with the rest of the narrative. Such sources of errors and biases need further systematic investigation.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*.

David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 352–361, Sofia, Bulgaria. Association for Computational Linguistics.

David Bamman, Ted Underwood, and Noah A. Smith. 2014. A bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 370–379, Baltimore, Maryland. Association for Computational Linguistics.

Christos Baziotis, Athanasiou Nikolaos, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth S. Narayanan, and Alexandros Potamianos. 2018. NTUA-SLP at semeval-2018 task 1: Predicting affective content in tweets with deep attentive RNNs and transfer learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 245–255.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's GPT-2 - how can I help you? Towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22.

Snigdha Chaturvedi, Haoruo Peng, and Dan Roth. 2017. Story comprehension for predicting what happens next. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1603–1614.

Jiaao Chen, Jianshu Chen, and Zhou Yu. 2019. Incorporating structured commonsense knowledge in story completion. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 6244–6251.

Elizabeth Clark, Yangfeng Ji, and Noah A. Smith. 2018. Neural text generation in stories using entity representations as context. 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 2250–2260.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *8th International Conference on Learning Representations*.

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*, pages 4171–4186.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Angela Fan, David Grangier, and Michael Auli. 2018a. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54.

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

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.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1243–1252.

Morton Gernsbacher, Harold Goldsmith, and Rachel Robertson. 1992. Do readers mentally represent characters' emotional states? *Cognition & Emotion*, 6(2):89–111.

Pablo Gervás, Belén Díaz-Agudo, Federico Peinado, and Raquel Hervás. 2005. Story plot generation based on CBR. *In Applications and Innovations in Intelligent Systems XII*, 28(1):33–46.

Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. 2017. Hafez: an interactive poetry generation system. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, System Demonstrations*, pages 43–48.

Simeng Gu, Wei Wang, Fushun Wang, and Jason H Huang. 2016. Neuromodulator and emotion biomarker for stress induced mental disorders. *Neural plasticity*.

Jian Guan, Fei Huang, Minlie Huang, Zhihao Zhao, and Xiaoyan Zhu. 2020. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, pages 93–108.

P.C. Hogan. 2011. *What Literature Teaches Us about Emotion*. Studies in Emotion and Social Interaction. Cambridge University Press.

Zhiting Hu, Haoran Shi, Bowen Tan, Wentao Wang, Zichao Yang, Tiancheng Zhao, Junxian He, Lianhui Qin, Di Wang, Xuezhe Ma, Zhengzhong Liu, Xiaodan Liang, Wanrong Zhu, Devendra Singh Sachan, and Eric P. Xing. 2019. Texar: A modularized, versatile, and extensible toolkit for text generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 159–164.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1587–1596.

Chenyang Huang, Osmar Zaïane, Amine Trabelsi, and Nouha Dziri. 2018. Automatic dialogue generation with expressed emotions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Short Papers)*, pages 49–54.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1534–1544.

Rachael E Jack, Oliver GB Garrod, and Philippe G Schyns. 2014. Dynamic facial expressions of emotion transmit an evolving hierarchy of signals over time. *Current biology*, 24(2):187–192.

Parag Jain, Priyanka Agrawal, Abhijit Mishra, Mohak Sukhwani, Anirban Laha, and Karthik Sankaranarayanan. 2017. Story generation from sequence of independent short descriptions. *Workshop on Machine Learning for Creativity*.

Neel Kant, Raul Puri, Nikolai Yakovenko, and Bryan Catanzaro. 2019. Practical text classification with large pre-trained language models. *CoRR*.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338.

Evgeny Kim and Roman Klinger. 2019. Frowning Frodo, wincing Leia, and a seriously great friendship: Learning to classify emotional relationships of fictional characters. 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 647–653.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 372–378.

Hidetsugu Komeda and Takashi Kusumi. 2006. The effect of a protagonist's emotional shift on situation model construction. *Memory & Cognition*, 34:1548–1556.

Vinodh Krishnan and Jacob Eisenstein. 2015. "you're mr. lebowski, I'm the dude": Inducing address term formality in signed social networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1616–1626.

Micheal Lebowitz. 1987. Planning stories. *In Proceedings of the cognitive science society*, pages 234–242.

VI Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10(8):707.

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.

Danyang Liu, Juntao Li, Meng-Hsuan Yu, Ziming Huang, Gongshen Liu, Dongyan Zhao, and Rui Yan. 2020. A character-centric neural model for automated story generation. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Fuli Luo, Damai Dai, Pengcheng Yang, Tianyu Liu, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. Learning to control the fine-grained sentiment for story ending generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6020–6026.

Lara J. Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark O. Riedl. 2018. Event representations for automated story generation with deep neural nets. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 868–875.

Robert McKee. 2003. Storytelling that moves people: A conversation with screenwriting coach robert mckee. *Harvard business review*, 81:51–5, 136.

Hardik Meisheri and Lipika Dey. 2018. TCS research at SemEval-2018 task 1: Learning robust representations using multi-attention architecture. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 291–299.

Gonzalo Méndez, Pablo Gervás, and Carlos León. 2016. On the use of character affinities for story plot generation. In *Knowledge, Information and Creativity Support Systems*, pages 211–225.

Saif Mohammad. 2018. Word affect intensities. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, pages 173–183.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17.

Daniel G. Morrow. 1985. Prominent characters and events organize narrative understanding. *Journal of Memory and Language*, 24(3):304–319.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley New York.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Brian Parkinson and Antony Manstead. 1993. Making sense of emotion in stories and social life. *Cognition and Emotion*, 7(3-4):295–323.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations*.

Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. 2018. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, pages 43–49.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Rafael Pérez y Pérez and Mike Sharples. 2001. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental Theoretical Artificial Intelligence*, 13(2):119–139.

Robert Plutchik. 1982. A psychoevolutionary theory of emotions. *Social Science Information*, 21(4-5):529–553.

Jullie Porteous and Mike Cavazza. 2009. Controlling narrative generation with planning trajectories: the role of constraints. *In ICIDS*, pages 234–245.

Gerald Prince. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, second edition. Pearson Prentice Hall.

Rafael Pérez y Pérez. 2007. Employing emotions to drive plot generation in a computer-based storyteller. *Cognitive Systems Research*, 8:89–109.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. 2019. Counterfactual story reasoning and generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5043–5053.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1179–1195.

Mark O. Riedl and R. Michael Young. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:217–268.

Melissa Roemmele. 2016. Writing stories with help from recurrent neural networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, page 4311–4312.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 3027–3035.

Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. 2019. Long and diverse text generation with planning-based hierarchical variational model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3257–3268.

Zhenqiao Song, Xiaoqing Zheng, Lu Liu, Mu Xu, and Xuanjing Huang. 2019. Generating responses with a specific emotion in dialog. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3685–3695.

Pradyumna Tambwekar, Murtaza Dhuliawala, Lara J. Martin, Animesh Mehta, Brent Harrison, and Mark O. Riedl. 2019. Controllable neural story plot generation via reward shaping. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence,*, pages 5982–5988.

Mariët Theune, Sander Rensen, Rieks op den Akker, Dirk Heylen, and Anton Nijholt. 2004. Emotional characters for automatic plot creation. In *Technologies for Interactive Digital Storytelling and Entertainment*, pages 95–100.

Hardik Vala, David Jurgens, Andrew Piper, and Derek Ruths. 2015. Mr. bennet, his coachman, and the archbishop walk into a bar but only one of them gets recognized: On the difficulty of detecting characters in literary texts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 769–774.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 5998–6008.

Kurt Vonnegut. 1981. Palm sunday. *RosetTaBooks, LLC New York.*

Di Wang, Nebojsa Jojic, Chris Brockett, and Eric Nyberg. 2017. Steering output style and topic in neural response generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2140–2150.

Noah Weber, Leena Shekhar, Heeyoung Kwon, Niranjan Balasubramanian, and Nathanael Chambers. 2020. Generating narrative text in a switching dynamical system. *CoRR*, abs/2004.03762.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *CoRR*, abs/1901.08149.

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.

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

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 730–739.

Xianda Zhou and William Yang Wang. 2018. MojiTalk: Generating emotional responses at scale. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pages 1128–1137.

A Training and Experiment Setup

A.1 Obtaining Emotion Reactions During RL-EM Training

During self-critical training, for each training instance, two stories are generated: $\boldsymbol{y}^s$ which is sampled from the model's probability distribution, and $\hat{\boldsymbol{y}}$ which is greedily decoded. Then for each sampled or greedy story, the value of the reward is computed. For computing the EC-EM reward, we need to identify and track the protagonist and obtain his/her emotional reactions for each sentence of the sampled or greedy story. This requires identifying the protagonist and determining his/her role, *Agent* or *Other*, in every sentence so that the appropriate argument for COMET (`xReact` or `oReact`) can be chosen. In principle, this can be done using the annotation pipeline described in §4.1 of the main paper. However, doing this is computationally prohibitive during training as the pipeline requires running dependency parsing for each sentence and coreference resolution for every sampled or greedy story. To this end, upon analyzing our corpus and some generated stories, we devised several heuristics that approximate the tasks (i.e. identifying protagonists and their roles) with high accuracy. We describe these heuristics below.

For identifying the protagonist, we use the following heuristics. The first heuristic is based on the observation that if the narrator features in a story, the story primarily focuses on the narrator and his/her experiences, thus making him/her the protagonist. So our first heuristic is that if first person pronouns (I, We) appear in the story, they are considered the protagonist. Our second heuristic is based on the observation that the protagonist is usually introduced fairly early in the story. Especially, in our case, where the stories are 5-sentence long, the protagonist appears mostly in the first couple of sentences of the story. With this in mind, we define the first noun that appears in a lexicon of common protagonists as the protagonist of the story. This lexicon consists of terms for `Male_Char`, `Female_Char`, `Social_Group`, `Generic_People`, and the NLTK name corpus. Example of these terms are shown in Table 5. This also let's us identify the gender of the protagonist (using the lexicon category that the protagonist belongs to) and hence the pronouns that will be used in the following sentences to refer to the protagonist. This combination of protagonist's mentions and pronouns lets us track him/her throughout the story.

For identifying the role of the protagonist, if the first noun that occurs in a sentence matches the protagonist or his/her corresponding pronoun, we assume that the protagonist's role is *Agent*, otherwise the role is *Other*. Depending on this role, we use `xReact` or `oReact` when obtaining protagonist's emotional reaction in that sentence using COMET.

`Male_Char`	husband, father, dad, dady, brother, grandpa, granddad, son, nephew, man, boy, boyfriend
`Female_Char`	wife, mother, mom, momy, sister, grandma, grandmom, niece, daughter, nana, woman, girl, girlfriend
`Social_Group`	family, parents, grandparents, children, kids, couple, friends, boys, girls, band
`Generic_People`	cousin, friend, fiance, boss, manager, assistant, doctor, nurse

Table 5: Predefined terms used for tracking the protagonist.

A.2 Training Hyper-parameters

Our proposed models follow the setting of medium-sized GPT-2 (Radford et al., 2019) (345 million parameters) that used a 24-layer decoder-only transformer, 1024-dimensional hidden states, and 16 attention heads. The stories are encoded using BPE with vocabulary size of $50,257$. We set the maximum sequence length to 128 tokens, as it is large enough to contain complete stories and additional inputs. We use Adam optimization (Kingma and Ba, 2015) with an initial learning rate of 10^{-5} and minibatch of size 4. For stability, we first pre-train the models with teacher forcing until convergence, then fine-tune them with the mixed loss. Hyper-parameter $\gamma = 0.97$ is tuned manually on the validation set. All models were trained until there was no improvement on the validation set performance. We use a NVIDIA GTX 1080 Ti GPU machine to train our models. At inference time, we generate stories using top-k sampling scheme (Fan et al., 2018b) with k=40 and a softmax temperature of 0.7. It took about 3 hours to generate stories for our test set of size $9,796$.

For generating commonsense inferences about the protagonist's emotions using COMET, we use greedy decoding algorithm since it has been shown to have superior performance as evaluated by humans (Bosselut et al., 2019).

Models	Domain	Accuracy (Jaccard)	Mico F1	Macro F1
Meisheri and Dey (2018)	tweets	0.582	0.694	0.534
Baziotis et al. (2018)	tweets	0.595	0.709	0.542
Kant et al. (2019)	tweets	0.577	0.690	0.561
BERT$_{large}$ (ours)	tweets	0.595	0.708	0.522
BERT$_{large}$ (ours)	stories	0.617	0.650	0.557

Table 6: Emotion classification results on the tweets dataset (upper block), and the automatically annotated story corpus (lower block).

A.3 Evaluation Measures

In this section we provide details about the automatic and manual measures used to evaluate our models.

Automatic To compute Arc-word and Seg-word measures, we use NRC Affect Intensity Lexicon (Mohammad, 2018). This lexicon contains words with corresponding emotion-intensities for different *basic emotions*. To find emotionally expressive words in a given piece of text (e.g., a story segment), we create a dictionary of words with emotion intensity higher than 0.5 for each of our *basic emotions*.

Manual During the manual evaluation, we conducted pairwise comparison of the models on Amazon Mechanical Turk (AMT). To ensure high quality of evaluation, we selected turkers that had an approval rate greater than 97%, had at least $1,000$ approved HITS, and were located in the U.S. For each pairwise annotation, we showed the inputs (title and emotion arc) and two stories generated using the two models being compared. In order to avoid biases, we randomly shuffled the order in which the stories from the two models were shown to the turkers. We provided instructions to the turkers explaining the annotations and also provided examples. Following this process, each pair of stories was annotated by three turkers. Fig. 6 shows a screenshot of our setup on AMT.

B Supplementary Results

B.1 Emotion Classification

Our EC-CLF reward captures the protagonist's emotions using our emotion classifier. In this section we provide details about its evaluation.

We first evaluate the classifier on the tweets corpus (Mohammad et al., 2018)[8] by comparing it with

[8] https://competitions.codalab.org/competitions/17751

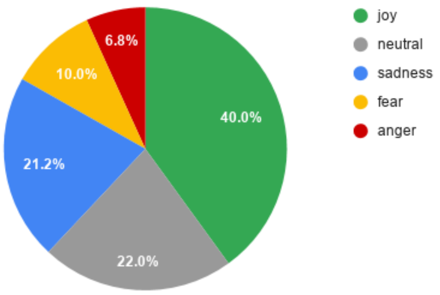


Figure 5: Label distribution of human annotated set.

several strong baselines (Kant et al., 2019). For this comparison, we trained all models on the training set of the corpora and tested them on a held-out test set. The models were evaluated using Jaccard Index based accuracy, and Micro and Macro F1 scores. This evaluation set-up (train-validation-test splits and choice of evaluation metrics) is as suggested in the challenge that provided the corpus (SemEval Task1:E-c challenge). The results of this comparison is shown in the top half of Table 6. We can see that our emotion classifier, BERT$_{large}$, is superior or competitive with other models.

The results reported above show that the model performs well for emotion classification in tweets. However, our goal is to design a model that can be used to track protagonist's emotions in stories. As described in the main paper, we further fine-tuned this classifier on our automatically annotated story corpora (described in the paper in §4.1). We also evaluated the classifier on a held-out portion of this story corpora consisting of about $1,201$ stories ($6,005$ sentences in total). The results are reported in the last row of Table 6. The classifier achieves a (Jaccard Index) accuracy of 61.75% and micro and macro F1 scores of 0.650 and 0.557 respectively. Note that this is different from the evaluation reported in the paper, which was conducted on a subset of stories annotated by humans.

B.2 Manual Annotation for Protagonist's Emotions

As described in the paper (§3.1), the emotion classifier was also evaluated on a subset of 50 randomly selected stories (250 sentences) manually annotated for the emotions experienced by their protagonists. This annotation was done on Amazon Mechanical Turk. To ensure good quality of the annotations, we selected turkers who had an approval rate greater than 97%, had at least $1,000$ approved HITS, and were located in the U.S. Fig. 7 shows a screenshot of our setup. The Fleiss kappa (inter-annotator agreement) was also moderate ($\kappa = 0.55$). We also analyzed the annotations to identify major sources of disagreements between the turkers. We found that most disagreements occurred between *neutral* and *joy*; and also between *sadness* and *anger*. The overall label distribution for this human annotated set is shown in Fig 5.

B.3 Base Storytelling Model

Baselines We compare our base storytelling model, GPT-2, with following state-of-the-art models:

1. *S2S*, an LSTM-based seq2seq model with attention (Bahdanau et al., 2015)

2. *ConvS2S*, a convolutional seq2seq model (Gehring et al., 2017) with decoder self-attention.

3. *Fusion*, a storytelling model that first pretrains a convolutional seq2seq model, then fixes the trained model and passes it to the second clone model with fusion mechanism (Fan et al., 2018b).

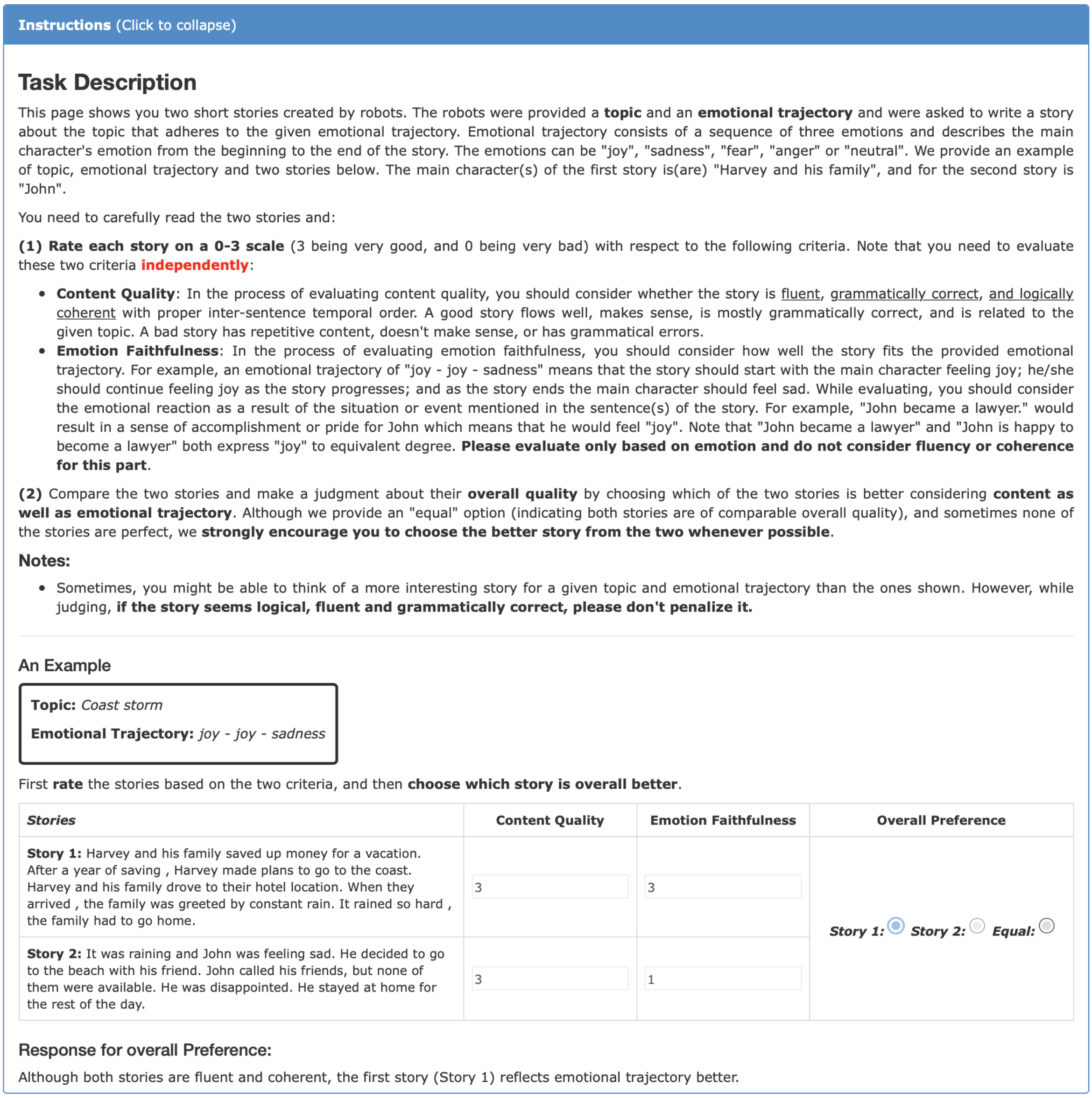
4. *Plan&Write*, another storytelling model that, given a title, first generates a plot as a sequence of keywords, and then conditioned on the plot it generates the text of the story (Yao et al., 2019).

5. *Decom-Event*, Fan et al. (2019) proposes to decompose the generation to two steps: generating successive events as the story plots, and generating story by surface realization of the plots. Events are represented by 4-tuples $<s, v, o, m>$, where s and o are the subject and object of verb v, and m is the modifier.

All of these models are trained, validated and tested on the same data splits described in §4.1.

Results The models are evaluated on content quality using the automatic measures described in the paper in §4.3. For comparison with other baselines, we compute *word-level* perplexity (PPL) for GPT-2+FT. That is, we normalize the total negative log probability of the target text by the number of *word*-level tokens (similar to the baselines), not the number of BPE tokens.

Table 7 describes the results of this evaluation. We can see that fine-tuned GPT-2 performs better



Instructions (Click to collapse)

Task Description

This page shows you two short stories created by robots. The robots were provided a **topic** and an **emotional trajectory** and were asked to write a story about the topic that adheres to the given emotional trajectory. Emotional trajectory consists of a sequence of three emotions and describes the main character's emotion from the beginning to the end of the story. The emotions can be "joy", "sadness", "fear", "anger" or "neutral". We provide an example of topic, emotional trajectory and two stories below. The main character(s) of the first story is(are) "Harvey and his family", and for the second story is "John".

You need to carefully read the two stories and:

(1) Rate each story on a 0-3 scale (3 being very good, and 0 being very bad) with respect to the following criteria. Note that you need to evaluate these two criteria **independently**:

- **Content Quality**: In the process of evaluating content quality, you should consider whether the story is fluent, grammatically correct, and logically coherent with proper inter-sentence temporal order. A good story flows well, makes sense, is mostly grammatically correct, and is related to the given topic. A bad story has repetitive content, doesn't make sense, or has grammatical errors.
- **Emotion Faithfulness**: In the process of evaluating emotion faithfulness, you should consider how well the story fits the provided emotional trajectory. For example, an emotional trajectory of "joy - joy - sadness" means that the story should start with the main character feeling joy; he/she should continue feeling joy as the story progresses; and as the story ends the main character should feel sad. While evaluating, you should consider the emotional reaction as a result of the situation or event mentioned in the sentence(s) of the story. For example, "John became a lawyer." would result in a sense of accomplishment or pride for John which means that he would feel "joy". Note that "John became a lawyer" and "John is happy to become a lawyer" both express "joy" to equivalent degree. **Please evaluate only based on emotion and do not consider fluency or coherence for this part**.

(2) Compare the two stories and make a judgment about their **overall quality** by choosing which of the two stories is better considering **content as well as emotional trajectory**. Although we provide an "equal" option (indicating both stories are of comparable overall quality), and sometimes none of the stories are perfect, we **strongly encourage you to choose the better story from the two whenever possible**.

Notes:

- Sometimes, you might be able to think of a more interesting story for a given topic and emotional trajectory than the ones shown. However, while judging, **if the story seems logical, fluent and grammatically correct, please don't penalize it.**

An Example

Topic: *Coast storm*

Emotional Trajectory: *joy - joy - sadness*

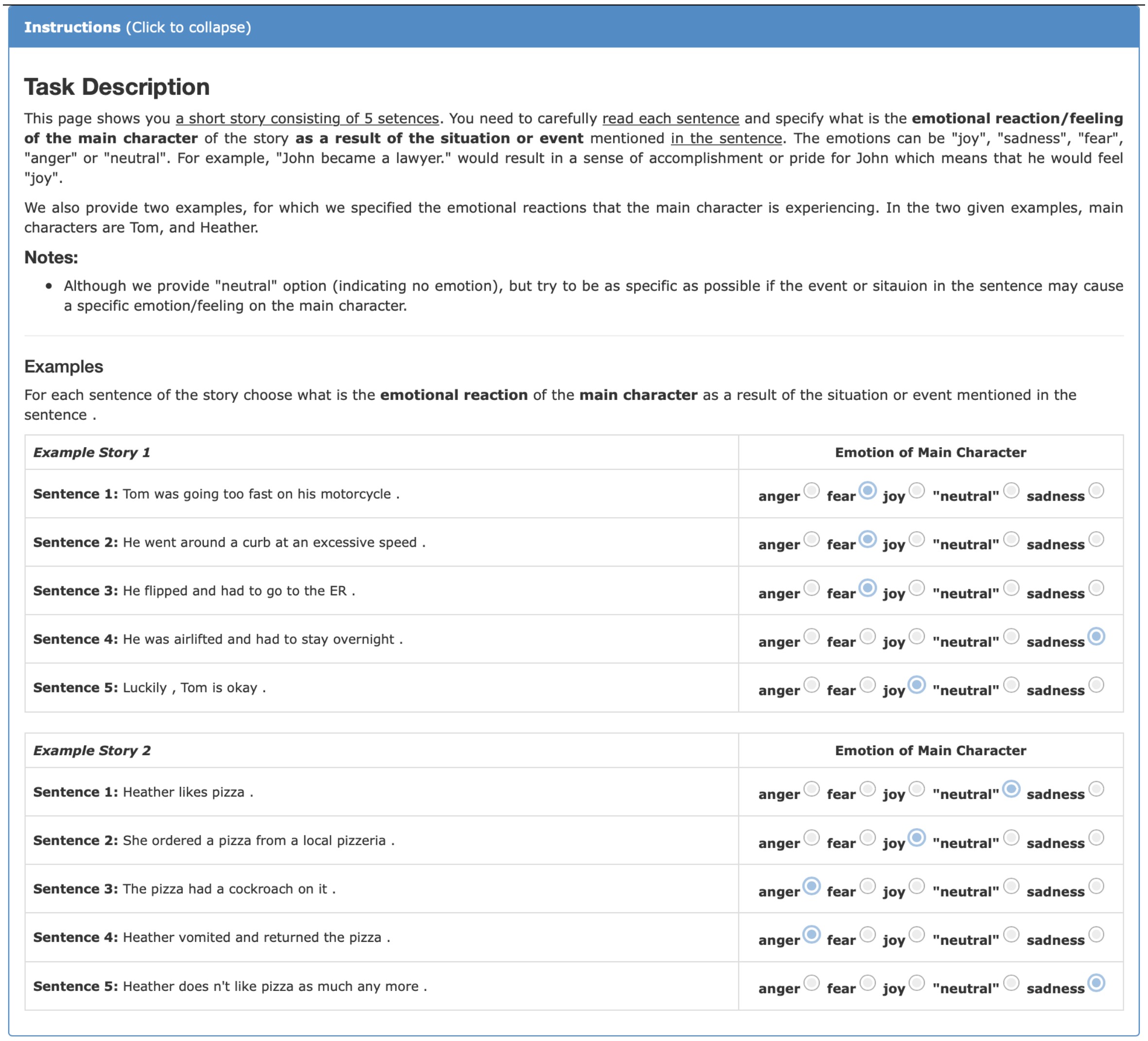
First **rate** the stories based on the two criteria, and then **choose which story is overall better**.

Stories	**Content Quality**	**Emotion Faithfulness**	**Overall Preference**
Story 1: Harvey and his family saved up money for a vacation. After a year of saving , Harvey made plans to go to the coast. Harvey and his family drove to their hotel location. When they arrived , the family was greeted by constant rain. It rained so hard , the family had to go home.	3	3	***Story 1:*** ● ***Story 2:*** ○ ***Equal:*** ○
Story 2: It was raining and John was feeling sad. He decided to go to the beach with his friend. John called his friends, but none of them were available. He was disappointed. He stayed at home for the rest of the day.	3	1	

Response for overall Preference:

Although both stories are fluent and coherent, the first story (Story 1) reflects emotional trajectory better.

Figure 6: An screenshot of manual evaluation on AMT.



Instructions (Click to collapse)

Task Description

This page shows you a short story consisting of 5 setences. You need to carefully read each sentence and specify what is the **emotional reaction/feeling of the main character** of the story **as a result of the situation or event** mentioned in the sentence. The emotions can be "joy", "sadness", "fear", "anger" or "neutral". For example, "John became a lawyer." would result in a sense of accomplishment or pride for John which means that he would feel "joy".

We also provide two examples, for which we specified the emotional reactions that the main character is experiencing. In the two given examples, main characters are Tom, and Heather.

Notes:

- Although we provide "neutral" option (indicating no emotion), but try to be as specific as possible if the event or sitauion in the sentence may cause a specific emotion/feeling on the main character.

Examples

For each sentence of the story choose what is the **emotional reaction** of the **main character** as a result of the situation or event mentioned in the sentence .

Example Story 1	**Emotion of Main Character**
Sentence 1: Tom was going too fast on his motorcycle .	**anger** ○ **fear** ● **joy** ○ **"neutral"** ○ **sadness** ○
Sentence 2: He went around a curb at an excessive speed .	**anger** ○ **fear** ● **joy** ○ **"neutral"** ○ **sadness** ○
Sentence 3: He flipped and had to go to the ER .	**anger** ○ **fear** ● **joy** ○ **"neutral"** ○ **sadness** ○
Sentence 4: He was airlifted and had to stay overnight .	**anger** ○ **fear** ○ **joy** ○ **"neutral"** ○ **sadness** ●
Sentence 5: Luckily , Tom is okay .	**anger** ○ **fear** ○ **joy** ● **"neutral"** ○ **sadness** ○

Example Story 2	**Emotion of Main Character**
Sentence 1: Heather likes pizza .	**anger** ○ **fear** ○ **joy** ○ **"neutral"** ● **sadness** ○
Sentence 2: She ordered a pizza from a local pizzeria .	**anger** ○ **fear** ○ **joy** ● **"neutral"** ○ **sadness** ○
Sentence 3: The pizza had a cockroach on it .	**anger** ● **fear** ○ **joy** ○ **"neutral"** ○ **sadness** ○
Sentence 4: Heather vomited and returned the pizza .	**anger** ● **fear** ○ **joy** ○ **"neutral"** ○ **sadness** ○
Sentence 5: Heather does n't like pizza as much any more .	**anger** ○ **fear** ○ **joy** ○ **"neutral"** ○ **sadness** ●

Figure 7: A screenshot of human annotation for protagonist's emotions.

Models	PPL (↓)	BLEU-1 (↑)	BLEU-2 (↑)	Dist-1 (↑)	Dist-2 (↑)	Dist-3 (↑)	Repet-4 (↓)
S2S	22.60	21.98	2.82	64.00	87.07	93.90	36.57
ConvS2S	23.90	20.37	2.63	66.98	91.22	95.68	19.29
Fusion	22.34	20.44	2.66	67.05	91.26	95.71	19.26
Plan&Write	21.11†	22.27	3.03	65.18	88.45	95.11	30.43
Decom-Event	18.16†	18.14	1.69	68.82	92.39	97.40	20.52
GPT-2 + FT	**12.16**	**22.68**	**3.10**	**72.93**	**94.24**	**98.28**	**12.10**

Table 7: Base Storytelling model: Automatic evaluation. The scores marked with † indicate models that have access to extra ground-truth information besides title (keywords and event tuples).

than all baselines for all measures. This demonstrates that it can be used as a good base storytelling model upon which our models are built.

B.4 Emotional-Aware Storytelling

Supplementary qualitative examples We provide more qualitative examples in Fig 8. In the figure we show stories generated by our model for a given title and emotion arc for the protagonist.

Title: A new outfit
Emotion arc: joy → joy → joy
Story: Sara went to the mall. She wanted to find something cute to wear. She noticed a beautiful red dress she liked. Sara bought the dress. She was happy that she found something nice.

Title: Wedding
Emotion arc: joy → anger → joy
Story: Ben is getting married. He is having fun. Suddenly , Ben gets into a huge fight with a guy. Ben gets out of the wedding early. Ben is so happy to have his wedding over.

Title: Study
Emotion arc: fear → fear → sadness
Story: Tina was so nervous for her big exam. She did not study at all. She was so afraid she would fail. She ended up failing. Tina felt very sad.

Title: The big storm
Emotion arc: joy → joy → sadness
Story: It was a beautiful day in the city. The sky was clear and the weather was mild. The school was hosting an evening celebration. A big storm rolled through and the students were worried. The storm destroyed all the decorations and ruined the party.

Title: Delayed flight
Emotion arc: joy → anger → sadness
Story: Tom was going on a vacation. He decided to go to Florida. Unfortunately his flight was delayed. He was so frustrated he called his airline. His airline cancelled his flight.

Title: The new pet
Emotion arc: neutral → joy → joy
Story: Sam was walking around the neighborhood. She saw a cute little dog. She decided to take him home. He got along well with everyone. Sam was glad to have a companion.

Title: Larry practice yoga
Emotion arc: fear → joy → joy
Story: Larry has always felt nervous about yoga. He has tried many times to practice but has never gotten the hang of it. He decides to take a yoga class at his local yoga studio. He is amazed by the benefits and feels confident about his yoga practice. Larry is happy he learned to enjoy yoga.

Figure 8: Qualitative examples of generated stories given a title and an emotion arc.