Towards Emotion-Aware Storytelling Using Reinforcement Learning

Faeze Brahman\textsuperscript{1}  Snigdha Chaturvedi\textsuperscript{2}
\textsuperscript{1}Department of Computer Science and Engineering, University of California, Santa Cruz
\textsuperscript{2}Department of Computer Science, University of North Carolina at Chapel Hill
\texttt{fbrahman@ucsc.edu} \texttt{snigdha@cs.unc.edu}

Abstract

Emotions and their evolution play a central role in creating a captivating story. Here, 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. Our models include Emotion Supervision (EmoSup) and Emotion-Reinforced (EmoRL) models. The EmoRL model uses special reward 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 \cite{1}. Stories are made interesting through emotions that connect the characters, their motivations, goals, and achievements \cite{2}.

Cognitive scientists have pinpointed the central role of emotions in storytelling \cite{3,4}. However, despite the broad recognition of its importance, neural story generation methods have not explored the modeling of emotional trajectory. In this paper, we present the first study to take into account the emotional trajectory of the protagonist in neural story generation. At any point in a story, we represent the protagonist’s emotions using a set of basic emotions. Following recent theories \cite{5,6}, 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.

For modeling the evolving emotions of the protagonist, we define an emotion arc for a story. According to Prince’s change-of-state formalization \cite{7}, 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 two models based on GPT-2 \cite{8} that incorporate the protagonist’s emotion arc as a controllable attribute while preserving content quality: an Emotion Supervision (EmoSup), and Emotion-Reinforced (EmoRL) model based on reinforcement learning. To encourage the model to adhere to the given emotion arc, the EmoRL model uses an Emotion-Consistency reward, (EC-CLF), which is computed through an emotion classifier.
In the absence of a training corpus of stories labeled with the protagonist’s emotions, we automatically annotate a large-scale story corpus using Commonsense Transformers, COMET [9]. Our automatic and manual evaluations show that our models can not only express the desired emotion arcs but also produce fluent and coherent stories.

2 Emotion-aware Storytelling

In this work, we define the protagonist as the most frequently occurring character in a narrative [10]. We formulate the emotion-aware storytelling task as: given a story title as a sequence of tokens \( t = \{t_1, t_2, ..., t_m\} \), and an emotion arc for the protagonist as a sequence of basic emotions \( a = \{e_1, e_2, e_3\} \), generate a story \( y = \{y_1, y_2, ..., y_n\} \) that adheres to the title and emotion arc.

2.1 Transformer-based Storytelling Model

We choose GPT2 [8] as our base storytelling model because our initial experiments and recent works [11; 12] demonstrated that it outperforms other SOTA story generation models, in general. GPT2 uses multiple Transformer blocks of multi-head self-attention and fully connected layers (the left box in Fig. 2). We fine-tune GPT2 on a dataset of stories by minimizing the conditional log-likelihood:

\[
L_{ML} = -\sum_{i=m}^{m+n} \log p(y_i|y_{<i}, t)
\]

where \( m \) and \( n \) denote the number of tokens in the title and story respectively.

2.2 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 [13; 14]. Each title has the corresponding emotion arc prepended at the beginning, separated by a delimiter token. This way, emotion arcs receive special treatment [15], since they are propagated to all of the story and the model learns to maximize \( p(y_i|y_{<i}, t, a) \).

2.3 Emotion-Reinforced (EmoRL) Model

The emotion arc guides the generation in EmoSup as an initial input. However, we want to continually supervise the model. This motivates us to use a reinforcement learning framework where we propose an Emotion Consistency reward, EC-CLF, which optimizes adherence to the desired emotion arc.

**EC-CLF Reward** This reward infers the protagonist’s emotions in a given text 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. Training details for classifier is provided in Appendix A.

For computing the reward, we divide the generated story into segments: beginning, body, and ending.

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:

\[
r_{clf} = \frac{1}{k} \sum_{j=1}^{k} p_{clf}(e_j^*|x_j)
\]

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

**Policy Gradient** For training, we use the REINFORCE algorithm [16] to learn a generation policy $p_\theta$ of the storytelling model. The model generates a sample story $y^s$ from the model’s output distribution, and the goal is to minimize the negative expected reward, which is approximated by:

$$L_{RL} = -(r(y^s) - r(\hat{y})) \sum_{i=k+m}^{k+m+n} \log p_\theta(y^s_i | y^s_{<i})$$  \hspace{1cm} (3)

We follow the self-critical training approach [17], and take the reward of the greedily decoded story $\hat{y}$ as the baseline reward $(r(\hat{y}))$. We optimize the following mixed loss [18]:

$$L_{\text{mixed}} = \gamma L_{RL} + (1 - \gamma) L_{ML}$$  \hspace{1cm} (4)

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

### 2.4 Dataset and Annotation Pipeline

We use the ROCStories corpus [19] for our experiments. We automatically annotated the stories with emotion arcs of the protagonists using a multi-step annotation pipeline described in more details in Appendix B (Fig. 3). In particular, we use a commonsense knowledge model, COMET [9], to reason about the emotional states of the protagonist at each sentence of the story which is in the form of phrases. We then map these phrases to one of the 5 basic emotions using NRC Affect Intensity Lexicon [20].

### 3 Experiments and Results

**Automatic Evaluation.** We use Perplexity, BLEU, Distinct-n, and Repetition-4 to evaluate content quality. To evaluate emotion faithfulness, we use (1) Seg-word, (2) Arc-word [21], (3) Seg-acc, (4) Arc-acc, and the reward function (5) EC-CLF. Details are provided in Appendix C.

**Manual Evaluation.** We also conduct a manual evaluation on AMT asking 3 judges to evaluate 100 pair of stories on a 0-3 scale with respect to emotion faithfulness and content quality and finally choose the better story or no preference. Details are provided in Appendix C.

**Baselines.** (1) GPT-2+FT, base GPT-2 model fine-tuned on the ROCStories corpus, for which emotion arcs are not provided as inputs; (2) Fusion+Emo [22] and (3) Plan&Write+Emo [23], which are two of the strongest storytelling baselines (we prepended emotion arcs to titles); and (4) PPLM [24].

**Results.** 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
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 | – | 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
EMoRL | 11.31 | 22.78 | 3.26 | 71.16 | 93.65 | 98.05 | 13.34

Table 1: Automatic evaluation of content quality (top) and emotion faithfulness (bottom). * indicates absence of emotion arc as input. For emotion faithfulness, EmoRL outperforms baselines (p < 0.05).

Specific Criteria | Overall Preference
---|---
Emotion Faithfulness | Content Quality | Better / Worse / Tie (%)
EmoRL 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
EmoRL 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
EmoRL vs. PPLM | +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 absolute scores for the two models, separated by a comma. RL-CLF is preferred over other methods (p < 0.05).

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 model). EmoRL has the highest BLEU scores, and EmoSup has the highest diversity and lowest repetition scores (p < 0.05).

The emotion faithfulness results (bottom) shows that, as expected, all models outperform GPT-2+FT, which is not provided the emotion arcs as inputs. In particular, EmoRL achieves the best performance on almost all measures.

Table 2 shows manual evaluation results, reporting the average improvements as well as absolute scores for emotion faithfulness and content quality (evaluated independently) and also the overall preference of the judges. Comparing with GPT-2+FT, we see that on average, EmoRL 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. All improvements are statistically significant (app. randomization, p < 0.05).

**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 can generate stories that follow different emotion 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 §B). For example, in Table 4 we show how EmoRL 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 E.1.
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 an ambulance. Luckily, the other person was treated for non-life threatening injuries.

sadness - sadness - 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.

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 and 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.

4 Conclusion

We proposed the emotion-aware storytelling task for modeling the emotion arc of the protagonist. Experiments demonstrated that our approach improved both content quality and emotion faithfulness. This paper is a step towards future research directions on planning emotional trajectory of various characters while generating stories. Our approach is general and provides a blueprint for similar works, e.g., for generating other emotional content or text with other attributes or properties.

References


A Emotion Classifier

Our EC-CLF reward captures the protagonist’s emotions using an emotion classifier. For this, we adapt the pre-trained BERT\textsubscript{large} for multi-label classification over 5 basic emotions: anger, fear, joy, sadness, and neutral. Following Devlin et al. [25], 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 [26], 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 [27] (Table 5). 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 §B. 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\%).

We first evaluate the classifier on the tweets corpus [26] by comparing it with several strong baselines [27]. 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 5. We can see that our emotion classifier, BERT\textsubscript{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 mentioned, we further fine-tuned this classifier on our automatically annotated story corpora (described in §B). 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 5. 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.

<table>
<thead>
<tr>
<th>Models</th>
<th>Domain</th>
<th>Accuracy (Jaccard)</th>
<th>Micro F1</th>
<th>Macro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meisheri and Dey [28]</td>
<td>tweets</td>
<td>0.582</td>
<td>0.694</td>
<td>0.534</td>
</tr>
<tr>
<td>Baziotis et al. [29]</td>
<td>tweets</td>
<td>0.595</td>
<td>0.709</td>
<td>0.542</td>
</tr>
<tr>
<td>Kant et al. [27]</td>
<td>tweets</td>
<td>0.577</td>
<td>0.690</td>
<td>0.561</td>
</tr>
<tr>
<td>BERT\textsubscript{large} (ours)</td>
<td>tweets</td>
<td>0.595</td>
<td>0.708</td>
<td>0.522</td>
</tr>
<tr>
<td>BERT\textsubscript{large} (ours)</td>
<td>stories</td>
<td>0.617</td>
<td>0.650</td>
<td>0.357</td>
</tr>
</tbody>
</table>

Table 5: Emotion classification results on the tweets dataset (upper block), and the automatically annotated story corpus (lower block).

B Dataset and Annotation Pipeline

We use the ROCStories corpus [19] 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

[https://competitions.codalab.org/competitions/17751](https://competitions.codalab.org/competitions/17751)
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.

<table>
<thead>
<tr>
<th>(1) Coreference Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character mentions:</td>
</tr>
<tr>
<td>Iris, she, her (sentences 1, 2, 3, 5)</td>
</tr>
<tr>
<td>Babysitter, sitter (sentences 1, 3, 4)</td>
</tr>
<tr>
<td>Daughter (sentence 1)</td>
</tr>
<tr>
<td>Husband (sentence 2)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(2) Protagonist Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character with the most mentions:</td>
</tr>
<tr>
<td>Iris</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(3) Protagonist’s Role Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protagonist’s roles:</td>
</tr>
<tr>
<td>Agent (sentences 1, 2, 3, 5)</td>
</tr>
<tr>
<td>Other (sentence 4)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(4) Generate Emotion Reaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protagonist’s emotional reactions:</td>
</tr>
<tr>
<td>1: satisfied (0.4)</td>
</tr>
<tr>
<td>2: anxious (0.5), 3: anxious (0.4), 4: annoyed (0.6)</td>
</tr>
<tr>
<td>5: sad (0.4)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(5) Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story-segments:</td>
</tr>
<tr>
<td>Beginning: satisfied</td>
</tr>
<tr>
<td>Body: annoyed</td>
</tr>
<tr>
<td>End: sad</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(6) Map to Basic Emotions via NRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emotion arc:</td>
</tr>
<tr>
<td>joy → anger → sadness</td>
</tr>
</tbody>
</table>

Figure 3: Annotation pipeline for emotion arc.

role as Agent or Other using its dependency parse:\footnote{We use AllenNLP \cite{30} for coreference resolution and dependency parsing: \url{https://github.com/allenai/allennlp}}. 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, a knowledge base completion model trained on ATOMIC if-then knowledge triples \cite{31}. 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, $\phi_g$. 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 \cite{32}, 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 \cite{20}. 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 each of the basic emotions, $e$, as $score_e(g)$. 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 $score_e(g)$ 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.
C 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. (2) **BLEU**, which is based on n-gram overlaps [33]. Following Guan et al. [12], 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 [34]. 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 [35]. 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** [21]. 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-Cl.F** to score a generated story. For all these measures, we report averaged scores across all stories generated by a model.

To compute Arc-word and Seg-word measures, we use NRC Affect Intensity Lexicon [20]. 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** We also conduct a manual evaluation of generated stories using Amazon Mechanical Turk. Following Song et al. [21], 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.

D Training Hyper-parameters

Our proposed models follow the setting of medium-sized GPT-2 [3] (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 [36] 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 [22] 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.

E Supplementary Results

E.1 Qualitative Examples

We provide more qualitative examples in Fig 4. In the figure we show stories generated by our model for a given title and emotion arc for the protagonist.

---

4For 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.
<table>
<thead>
<tr>
<th>Title: A new outfit</th>
<th>Emotion arc: joy → joy → joy</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title: Wedding</th>
<th>Emotion arc: joy → anger → joy</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title: Study</th>
<th>Emotion arc: fear → fear → sadness</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title: The big storm</th>
<th>Emotion arc: joy → joy → sadness</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title: Delayed flight</th>
<th>Emotion arc: joy → anger → sadness</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title: The new pet</th>
<th>Emotion arc: neutral → joy → joy</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title: Larry practice yoga</th>
<th>Emotion arc: fear → joy → joy</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4: Qualitative examples of generated stories given a title and an emotion arc.