Towards Emotion-Aware Storytelling Using Reinforcement Learning

Faeze Brahman[1] **Snigdha Chaturvedi**[2]
[1]Department of Computer Science and Engineering, University of California, Santa Cruz
[2]Department of Computer Science, University of North Carolina at Chapel Hill
fbrahman@ucsc.edu 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 [1]. Stories are made interesting through emotions that connect the characters, their motivations, goals, and achievements [2].

Cognitive scientists have pinpointed the central role of emotions in storytelling [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 [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 [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 [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.

Wordplay: When Language Meets Games Workshop @ NeurIPS 2020,
https://wordplay-workshop.github.io/

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 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:

$$\mathcal{L}_{ML} = -\sum_{i=m}^{m+n} \log p(y_i|y_{<i}, t) \tag{1}$$

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[1]. 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:

$$r_{clf} = \frac{1}{k}\sum_{j=1}^{k} p_{\text{clf}}(e_j^*|x_j) \tag{2}$$

[1]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.

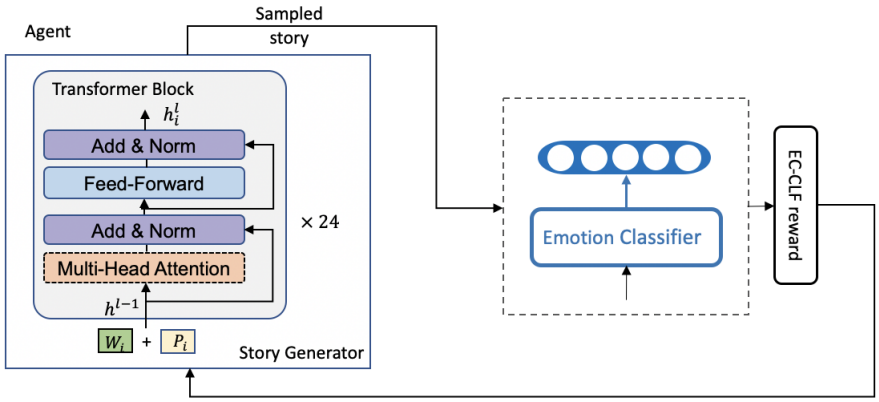

Figure 2: Transformer architecture (left) and emotion-reinforced storytelling framework (right)

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_θ 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:

$$\mathcal{L}_{RL} = -(r(y^s) - r(\hat{y})) \sum_{i=k+m}^{k+m+n} \log p_\theta(y_i^s | y_{<i}^s) \tag{3}$$

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]:

$$\mathcal{L}_{mixed} = \gamma \mathcal{L}_{RL} + (1 - \gamma)\mathcal{L}_{ML} \tag{4}$$

where γ 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**

Models	Arc-word	Seg-word	Arc-acc	Seg-acc	EC-CLF
Fusion + Emo	6.32	38.89	29.89	62.59	60.06
Plan&Write + Emo	5.61	32.98	26.38	60.99	58.13
PPLM	7.74	37.64	27.30	60.60	59.51
GPT-2 + FT*	4.46	33.28	17.32	48.69	47.93
EmoSup	7.33	40.86	31.25	64.26	62.88
EmoRL	**10.14**	**45.42**	**37.58**	**68.90**	**67.55**

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 ($2.24 \pm 0.80, 1.48 \pm 0.98$)	+0.25 ($2.25 \pm 0.82, 2.00 \pm 0.88$)	**60.00** / 22.00 / 18.00
EmoRL vs. EmoSup	+0.28 ($1.97 \pm 1.00, 1.69 \pm 1.05$)	+0.14 ($1.93 \pm 0.94, 1.79 \pm 0.97$)	**50.33** / 34.00 / 15.66
EmoRL vs. PPLM	+0.48 ($2.10 \pm 0.86, 1.62 \pm 0.94$)	+0.34 ($2.21 \pm 0.90, 1.87 \pm 0.96$)	**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 ($p < 0.05$).

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.

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.

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.

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

[1] R. McKee, "Storytelling that moves people: A conversation with screenwriting coach robert mckee," *Harvard business review*, vol. 81, pp. 51–5, 136, 07 2003.

[2] K. Vonnegut, "Palm sunday," *RosetTaBooks, LLC New York*, 1981.

[3] B. Parkinson and A. Manstead, "Making sense of emotion in stories and social life," *Cognition and Emotion*, vol. 7, pp. 295–323, 05 1993.

[4] P. Hogan, *What Literature Teaches Us about Emotion*. Studies in Emotion and Social Interaction, Cambridge University Press, 2011.

[5] R. E. Jack, O. G. Garrod, and P. G. Schyns, "Dynamic facial expressions of emotion transmit an evolving hierarchy of signals over time," *Current biology*, vol. 24, no. 2, pp. 187–192, 2014.

[6] S. Gu, W. Wang, F. Wang, and J. H. Huang, "Neuromodulator and emotion biomarker for stress induced mental disorders," *Neural plasticity*, 2016.

[7] G. Prince, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Prentice Hall, second ed., 2009.

[8] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, p. 8, 2019.

[9] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz, and Y. Choi, "COMET: Commonsense transformers for automatic knowledge graph construction," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4762–4779, 2019.

[10] D. Morrow, "Prominent characters and events organize narrative understanding," *Journal of Memory and Language*, vol. 24, no. 3, pp. 304–319, 1985.

[11] A. See, A. Pappu, R. Saxena, A. Yerukola, and C. D. Manning, "Do massively pretrained language models make better storytellers?," 2019. cite arxiv:1909.10705Comment: Accepted to CoNLL 2019.

[12] J. Guan, F. Huang, M. Huang, Z. Zhao, and X. Zhu, "A knowledge-enhanced pretraining model for commonsense story generation," *Transactions of the Association for Computational Linguistics*, pp. 93–108, 2020.

[13] A. Fan, D. Grangier, and M. Auli, "Controllable abstractive summarization," in *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 45–54, 2018.

[14] Y. Kikuchi, G. Neubig, R. Sasano, H. Takamura, and M. Okumura, "Controlling output length in neural encoder-decoders," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1328–1338, 2016.

[15] C. Kobus, J. Crego, and J. Senellart, "Domain control for neural machine translation," in *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pp. 372–378, 2017.

[16] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, p. 229–256, May 1992.

[17] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1179–1195, 2017.

[18] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," in *6th International Conference on Learning Representations*, 2018.

[19] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen, "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*, pp. 839–849, 2016.

[20] S. Mohammad, "Word affect intensities," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation* (N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, eds.), pp. 173–183, 2018.

[21] Z. Song, X. Zheng, L. Liu, M. Xu, and X. Huang, "Generating responses with a specific emotion in dialog," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3685–3695, 2019.

[22] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, 2018.

[23] L. Yao, N. Peng, R. M. Weischedel, K. Knight, D. Zhao, and R. Yan, "Plan-and-write: Towards better automatic storytelling," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pp. 7378–7385, 2019.

[24] S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu, "Plug and play language models: A simple approach to controlled text generation," in *8th International Conference on Learning Representations*, 2020.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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*, pp. 4171–4186, 2019.

[26] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, "SemEval-2018 task 1: Affect in tweets," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 1–17, 2018.

[27] N. Kant, R. Puri, N. Yakovenko, and B. Catanzaro, "Practical text classification with large pre-trained language models," *CoRR*, 2019.

[28] H. Meisheri and L. Dey, "TCS research at SemEval-2018 task 1: Learning robust representations using multi-attention architecture," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 291–299, 2018.

[29] C. Baziotis, A. Nikolaos, A. Chronopoulou, A. Kolovou, G. Paraskevopoulos, N. Ellinas, S. S. Narayanan, and A. Potamianos, "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* (M. Apidianaki, S. M. Mohammad, J. May, E. Shutova, S. Bethard, and M. Carpuat, eds.), pp. 245–255, 2018.

[30] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. F. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer, "AllenNLP: A deep semantic natural language processing platform," in *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pp. 1–6, 2018.

[31] M. Sap, R. L. Bras, E. Allaway, C. Bhagavatula, N. Lourie, H. Rashkin, B. Roof, N. A. Smith, and Y. Choi, "ATOMIC: an atlas of machine commonsense for if-then reasoning," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pp. 3027–3035, 2019.

[32] S. Chaturvedi, H. Peng, and D. Roth, "Story comprehension for predicting what happens next," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1603–1614, Sept. 2017.

[33] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

[34] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "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*, pp. 110–119, 2016.

[35] Z. Shao, M. Huang, J. Wen, W. Xu, and X. Zhu, "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*, pp. 3257–3268, 2019.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations* (Y. Bengio and Y. LeCun, eds.), 2015.

A 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. [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][2] 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$_{large}$, 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.

Models	Domain	Accuracy		
(Jaccard)	Mico F1	Macro F1		
Meisheri and Dey [28]	tweets	0.582	0.694	0.534
Baziotis et al. [29]	tweets	0.595	0.709	0.542
Kant et al. [27]	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 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

[2] `https://competitions.codalab.org/competitions/17751`

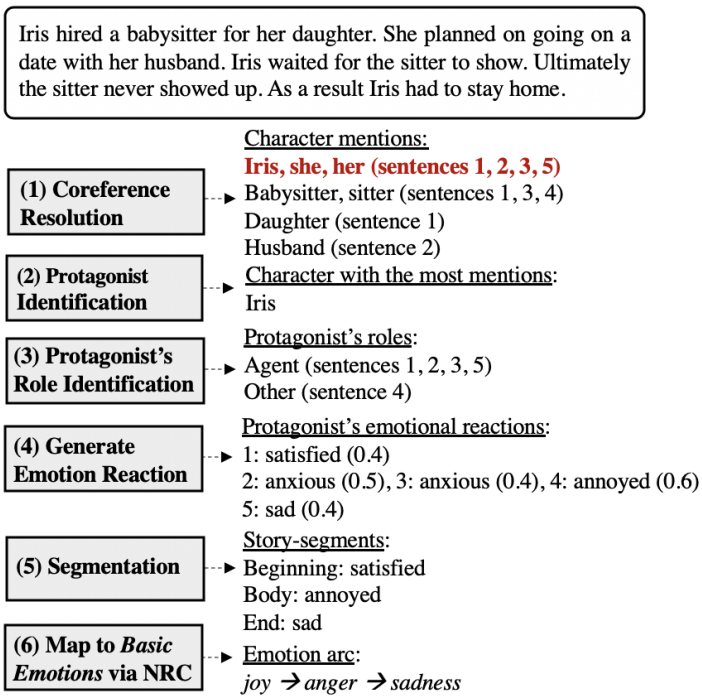

Figure 3: Annotation pipeline for *emotion arc*.

role as *Agent* or *Other* using its dependency parse[3]. 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 [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, 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 [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 [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 $score_e(\mathrm{e})$. 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 $score_g(\mathrm{e})$ 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*.

[3]We use AllenNLP [30] for coreference resolution and dependency parsing: `https://github.com/allenai/allennlp`

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[4]. (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-CLF** 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 [8] (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.

[4]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.

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