Modeling Protagonist Emotions for Emotion-Aware Storytelling

Faeze Brahman
Snigdha Chaturvedi
Motivation

What makes stories interesting and engaging to readers?

★ The emotion the storyteller puts into the narrative by connecting moments with characters, their motivations, goals, and achievements.

Modeling the emotional trajectory of the protagonist
Overview

**Task**: Emotion-Aware Storytelling

**Models**: Emotion-Reinforced (EmoRL) models

**Dataset** : Annotation Pipeline

**Evaluation**: Automatic / Human
Tell me a story about **Raw Burger:**

**Protagonist**

Tom went to a burger place with his friends.

**Emotion Arc**

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.
What if we want to generate stories with different emotion arcs?

**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.
Challenges

It is difficult to balance fluency and expressions of emotions in a natural and coherent way.

High-quality large-scale emotion-annotated data are hard to obtain.

Automatically annotate story corpus using COMET (Bosselut et al. 2019)
Overview

**Task:** Emotion-Aware Storytelling

**Models:** Emotion-Reinforced (EmoRL) models

**Dataset:** Annotation Pipeline

**Evaluation:** Automatic / Human
Overview

Task: Emotion-Aware Storytelling

Models: Emotion-Reinforced (EmoRL) models

Dataset: Annotation Pipeline

Evaluation: Automatic / Human
Approach

We build our model based on GPT2 (Radford et al. 2019):

We design two special rewards to *regularize the story generation* and *control emotion arc of the protagonist* through Reinforcement Learning.

We explore two different ways of tracking the emotions:

- Commonsense Transformer (COMET)
- Emotion Classifier
Tom yelled at the waiter.

Tracking Emotion — Classifier

Adapt a pre-trained BERT for multi-label classification over 5 basic emotions

Two-step fine-tuning:

- On human-annotated dataset (tweets)
- Further fine-tune on story training data — automatically annotated with the protagonist’s emotions
Storytelling Model Arch.

The story generator interact with environment during training to regularize the generation

- Emotion Matching (Ec-Em)
- Emotion Classifier (Ec-Clf)

Input:

- Title: \( t = \{t_1, t_2, \ldots, t_m\} \)
- Emotion arc: \( a^* = \{e_1, e_2, e_3\} \)

Output:

- Story: \( \gamma = \{y_1, y_2, \ldots, y_n\} \)
EmoRL — RL-EM

Ec-Em reward quantify the string alignment of the emotion arc of the generated story to desired emotion arc.

Generated story emotional reactions:

\[ a^g = \{g_1, g_2, \ldots, g_N\} \]

Desired Emotion arc:

\[ a^* = \{e_1, e_2, e_3\} \]

Reward:

\[ r_{em} = \text{levenshtein}(a^g, a^*) \]

**EmoRL — RL-CLF**

The equation for the reward is given by:

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

**Ec-CLf reward measures the probability of desired emotion arc using emotion classifier.**

---

*GPT2-M*

Policy Gradient

self-critical:
\[
\mathcal{L}_{RL} = -(r(y^s) - r(\hat{y})) \sum_{i=k+m}^{k+m+n} \log p_{\theta}(y^s_i | y^s_{<i})
\]

GPT2-M
Policy Gradient

**Self-critical:**

$$\mathcal{L}_{RL} = -(r(y^s) - r(\hat{y})) \sum_{i=k+m}^{k+m+n} \log p(y^s_i | y_\leq i)$$

**ML Loss:**

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

**Mixed Loss:**

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

---

**GPT2-M**
Overview

**Task:** Emotion-Aware Storytelling

**Models:** Emotion-Reinforced (EmoRL) models

**Dataset:** Annotation Pipeline

**Evaluation:** Automatic / Human
Overview

**Task**: Emotion-Aware Storytelling

**Models**: Emotion-Reinforced (EmoRL) models

**Dataset**: Annotation Pipeline

**Evaluation**: Automatic / Human
Dataset — Annotation Pipeline

ROCStories Corpus
(Mosatafazadeh et al.)

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) Coref Resolution
Character mentions
Dataset — Annotation Pipeline

**ROCStories Corpus**
(Mosatafazadeh et al.)

_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.

(2) Protag. Identification
Character with the most mentions: Iris
ROCStories
Corpus
(Mosatafazadeh et al.)

**Agent**

*Iris* hired a babysitter for *her* daughter.

*She* planned on going on a date with *her* husband.

*Agent*  
*Iris* waited for the sitter to show.

**Other**

Ultimately, the sitter never showed up.

*Agent*  
As a result, *Iris* had to stay home.

(3) **Protag. Role Identification**

Protagonist’s roles:  
Agent / Other
Dataset — Annotation Pipeline

**ROCStories Corpus**
(Mosatafazadeh et al.)

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.

(4) Generate Emotion Reaction

Protagonist’s Emotional reactions

- Satisfied
- Anxious
- Anxious
- Annoyed
- Sad
Dataset — Annotation Pipeline

**ROCStories Corpus**
(Mosatafazadeh et al.)

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.

(5) Segmentation

Satisfied

Annoyed

Sad
Dataset — Annotation Pipeline

**ROCStories Corpus**  
(Mosatafazadeh et al.)

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.

(6) Map to Basic Emo via NRC

joy

Satisfied

anger

Annoyed

sadness

Sad
Overview

**Task:** Emotion-Aware Storytelling

**Models:** Emotion-Reinforced (EmoRL) models

**Dataset:** Annotation Pipeline

**Evaluation:** Automatic / Human
Overview

Task: Emotion-Aware Storytelling

Models: Emotion-Reinforced (EmoRL) models

Dataset: Annotation Pipeline

Evaluation: Automatic / Human
Baselines

We use the following baselines:

- GPT2 + FT
- Fusion + Emo (Fan et al. 2018)
- Plan&Write + Emo (Yao et al. 2019)
- PPLM (Dathathri et al. 2020)
- EmoSup (GPT + FT + Emo)
Automatic Evaluation

Content quality:

- PPL, BLEU, Distinct-n, repetition-4

Emotion faithfulness:

- **Seg-word & Arc-word**: percent of segments/stories that contain corresponding emotion words.

- **Seg-acc & Arc-acc**: emotion accuracy of generated segments/stories as identified by emotion classifier.

- **EC-CLF & EC-EM (RL rewards)**
## Automatic Evaluation

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

Both Emo-reinforced models outperform all baselines for BLEU and diversity/repetition scores respectively.
Automatic Evaluation

Emotion Faithfulness

RL-CLf achieves the best performance followed by RL-Em

<table>
<thead>
<tr>
<th>Model</th>
<th>Arc-word</th>
<th>Seg-word</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fusion</td>
<td>12.5</td>
<td>37.5</td>
</tr>
<tr>
<td>P&amp;W</td>
<td>7.5</td>
<td>25.0</td>
</tr>
<tr>
<td>PPLM</td>
<td>10.0</td>
<td>27.5</td>
</tr>
<tr>
<td>GPT2+FT</td>
<td>8.0</td>
<td>30.0</td>
</tr>
<tr>
<td>EmoSup</td>
<td>15.0</td>
<td>32.5</td>
</tr>
<tr>
<td>RL-EM</td>
<td>17.5</td>
<td>35.0</td>
</tr>
<tr>
<td>RL-CLF</td>
<td>20.0</td>
<td>37.5</td>
</tr>
</tbody>
</table>
Automatic Evaluation

Emotion Faithfulness

All models are superior to GPT2+FT (as expected). RL-CLf is the best.
Human Evaluation

Test on 100 randomly selected title + emotion arc

Evaluation criteria:

- Emotion faithfulness (likert scale 0-3)
- Content quality (likert scale 0-3)
- Overall preference
RL-CLf improves over RL-Em on Emo. and Qual.

RL-CLf is not only better at adhering to emotion arc (+0.76) but also generates better content (+0.25)

Human Evaluation Results

Overall Preference of RL-CLF

RL-CLF is preferred over other models!

Win | Lose | Tie
---|---|---
vs. RL-EM: 9.66% | 38% | 52.33%
vs. GPT2+FT: 18% | 22% | 60%
vs. EmoSup: 15.66% | 34% | 50.33%
vs. PPLM: 13.33% | 25.66% | 61%
Summary

We present the first study on modeling the emotion arc of the protagonist in neural story generation.

We propose two emotion-consistency rewards designed to enforce the desired emotion arcs using RL.

We track the protagonist’s emotions using (1) commonsense knowledge models, and (2) emotion classifier trained using transfer learning.

We empirically demonstrate that our models can effectively generate stories that follow the desired emotion arc w/o sacrificing the story quality.
Thank you!
Questions?