Modeling Tension in Stories via Commonsense Reasoning and Emotional Word Embeddings

Anonymous ACL submission

Abstract

Dramatic tension is crucial for generating interesting stories. This paper aims to model dramatic tension from a story text using neural commonsense-reasoning language models and emotional word embeddings. We also propose a method of converting a categorical emotion word into a numerical value. The evaluation results using human-annotated stories demonstrate that our proposed method is promising in predicting tension development in a story.

1 Introduction

Tension is a feeling of tightness, anxiety, uncertainty, and stress, as opposed to the feeling of relaxation. Dramatic tension is a nervous feeling created while reading a book or watching a movie. In psychology, experiencing tension is associated with both negative and positive emotions. Tension, as a negative experience similar to stress, is not desirable. But it can also serve as “a major motivator to engage in certain activities” and thus be a positive experience (Lehne and Koelsch, 2015). Dramatic tension in a narrative keeps the reader engaged in the story by generating questions about the story’s outcome (Bal, 2017).

Tension aroused from a conflict is innately associated with a closure, since a conflict is resolved by achieving a closure (Abbott, 2008). The lack of closure in a narrative results in “suspense” - “an emotion or state of mind arising from a partial and anxious uncertainty about the progression or outcome of an action” (Prince, 2003), which is, along with surprise, central to interesting and successful narratives (Abbott, 2008; Hoeken and Vliet, 2000). Freytag (1863) proposed a pyramidal structure of a story that entertains the reader. A story forms a triangle, with tension escalating up to peak of the triangle, reaching a climax, then declining toward the story’s resolution.

Despite the importance of tension in a story, few studies have researched the computational aspects of tension. To the best of our knowledge, this paper is a first attempt to automatically calculate the dramatic tension elicited from reading a story text. While dramatic tension in general refers to the reader’s tension, it can also be aligned with the character’s tension when the two following conditions are met. First, the reader needs to empathize with the characters or to care about them. Second, there is no or little discrepancy between the knowledge of the reader and the characters. In this paper, we focus on the type of dramatic tension that is aligned with the character’s tension. Our contributions are as follows.

• We present a framework that measures tension using a commonsense language model and NLP techniques.
• We propose a method to convert a set of categorical emotion terms into a scalar representing tension using a word embedding model and an affective computing theory.
• We empirically evaluate our framework and demonstrate its utility in measuring dramatic tension.

2 Related Work

Dramatic tension has served as a guide for generating an intriguing plot in Interactive Storytelling (IS) fields such as Façade (Mateas and Stern, 2003) and IDTension (Szilas, 2003), based on the Aristotelian dramatic arc. Specifically, various attempts have been made in planning-based approaches to create dramatic tension. For instance, a story’s tension arcs are formalized by defining the reader’s knowledge about a clue in a “riddle” plot (Barros and Musse, 2008), and clichés-based dilemma is employed to add dramatic tension (Barber and Kudenko, 2007). In Ware et al. (2014), both external and internal conflicts deliver tension.
Different schemes are proposed to annotate story tension in diverse narrative media. For text stories, Li et al. (2018) presents 360 annotated short stories using ten categories of a dramatic arc. Yoon et al. (2019) suggests a machine-assisted annotation procedure to label tension development with TED talk videos. Specifically, Kybartas et al. (2021) explores tension space based on Ryan’s possible world theory (Ryan, 1991) in an emergent narrative game.

Dramatic tension arouses suspense, making the reader anticipate some notable events to unfold. Wilmot and Keller (2020) proposes a hierarchical language model for modeling narrative suspense by computing two types of unexpectedness - surprise in hindsight and uncertainty reduction looking forward. Several recent studies, in similar context, present unsupervised methods to learn and detect event salience in narratives, using pre-trained language models (Otake et al., 2020; Wilmot and Keller, 2021).

3 Our Method

Our framework consists of three stages: (1) protagonist detection; (2) commonsense reasoning; and (3) emotion quantification, as illustrated in Figure 1.

3.1 Protagonist Detection

In the protagonist detection stage, we identify all the entities mentioned in a story using a coreference resolution model. Following Brahman and Chaturvedi (2020)’s system, we regard the entity mentioned the most as the protagonist of the story. For each sentence, we examine whether it contains a protagonist or not, in order to determine the relation type (xReact or oReact) for commonsense reasoning.

3.2 Commonsense Reasoning

In this study, we measure dramatic tension based on the protagonist’s emotional reaction. To infer the emotional reaction, we utilize commonsense reasoning transformer model, which was trained on external knowledge bases such as ATOMIC-2020 (Hwang et al., 2021). The reasoning model provides two relations, xReact and oReact; xReact infers the emotion of the sentence’s subject, and oReact infers the emotion of its object (or the other). We infer with the xReact relation when a sentence contains a protagonist; otherwise, we use the oReact relation for reasoning. We infer the protagonist’s emotional reaction via the following one-hop and two-hop of reasoning.

Single-hop: Reaction based on the current event

First, this method focuses on the current event in a story. We directly infer the emotion of the protagonist elicited from the current story sentence. We generate up to 3 emotions for each sentence, and exclude ‘none’ or empty results for further processing.

Multi-hop: Reaction based on the next event

We hypothesize that a reader’s tension may arise in anticipation of an action or event that will follow the current event. We use commonsense reasoning with the xEffect and oEffect relations to infer a follow-up sentence that is likely to occur as a result of the current sentence. Then, we use only the xReact relation to extract the emotion, as a
reaction to the next predicted event that occurs to the protagonist.

### 3.3 Emotion Quantification

The commonsense reasoning step produces the protagonist’s emotional reactions as categorical terms such as happy, sad, frustrated, and angry. In order to estimate the tension value, we need to interpret them as numerical values. Therefore, we propose a novel method that computes an affect score of a word when two base vectors are given using a pre-trained word embedding model.

![Figure 2: Converting an categorical emotion word into a scalar representation using a pre-trained word embedding model and projection.](image)

First, we set a pair of affective words \((b_0, b_1)\) to form the base vector. Given the emotional reaction of the protagonist \(e\) and the pair of base words \(b_0\) and \(b_1\), we embed them into the pre-trained word embedding space. Then, we compute the scalar projection of the embedding vector of \(e\) onto the line between the embedding vectors of \(b_0\) and \(b_1\). We compute the scalar value of \(e\) as the length of the projected vector onto the vector connecting the embedding vectors of \(b_0\) and \(b_1\). We can combine multiple base vectors by averaging them. Drawn from affective computing theories, we choose the pair of base words with opposite meanings to one another, referring to the Plutchik’s wheel of emotions and 2D emotion wheels models (Plutchik, 1980a).

### 4 Experiments

#### 4.1 Dataset

We use the ROC stories (Mostafazadeh et al., 2016) corpus for the experiment. We use the winter 2017 set which has 52,665 stories. Each story consists of five sentences containing everyday events that are causally and temporally related. We select 200 stories from the corpus to build a dataset for tension modeling. We recruited a total of 39 crowdsourcing workers via Amazon Mechanical Turk (AMT), assigning five annotators per story. Since numerical ratings among different people can be arbitrary, we asked them to annotate the changes in their level of tension (i.e., less tense, more tense, or no change) as the story unfolds. The gold label for each sentence of a story is determined by majority voting among the five annotators. More information for our annotation is described in Appendix A.

#### 4.2 Implementation Details

We use the end-to-end neural coreference model provided in the AllenNLP library\(^1\) for protagonist detection. We use the COMET-ATOMIC 2020 BART model (Hwang et al., 2021) for commonsense reasoning. For emotion projection, we utilize a pre-trained emotional NumberBatch (Seyedtabari et al., 2019) which is an emotional word embedding of NumberBatch, incorporating emotional constraint into the base word embedding. For modeling tension change, we determine ‘more tense or less tense’ if the difference between the tension inferred from reading the previous sentence and that inferred from the current sentence is greater than or equal to 0.2; otherwise, we determine that there is no change in tension between the two sentences.

#### 4.3 Results

In our evaluation, we utilize the following two simple baselines since no previous works on modeling tension are available.

- **Freytag’s Pyramid** - Freytag’s Pyramid theorizes that a good story forms a triangle, where dramatic tension slowly escalates from the story’s beginning up to the climax, followed by a steep downward slope up to its end. We refer to the pattern of tension escalating up to the fourth sentence and de-escalating at the fifth sentence as in Freytag’s Pyramid.

- **Gaussian Distribution** - Adopting the view in (Lin and Riedl, 2021) regarding a story arc as Gaussian Distribution when tension rises up to the story’s middle and declines towards the end, we consider the pattern of two consecutive escalating tension followed by two consecutive declining tension as in the Gaussian Distribution.

\(^{1}\text{https://github.com/allenai/allennlp}\)
<table>
<thead>
<tr>
<th>Methods with base words</th>
<th>Accuracy</th>
<th>Macro-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freytag’s Pyramid</td>
<td>39.37</td>
<td>35.47</td>
</tr>
<tr>
<td>Gaussian Distribution</td>
<td>36.19</td>
<td>30.04</td>
</tr>
<tr>
<td>Single-hop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sadness ↔ joy</td>
<td>53.36</td>
<td>49.44</td>
</tr>
<tr>
<td>surprise ↔ anticipation</td>
<td>46.46</td>
<td>33.91</td>
</tr>
<tr>
<td>disgust ↔ trust</td>
<td>51.12</td>
<td>45.36</td>
</tr>
<tr>
<td>sadness, disgust ↔ joy, trust</td>
<td>53.92</td>
<td>50.23</td>
</tr>
<tr>
<td>tense ↔ calm</td>
<td>42.72</td>
<td>19.98</td>
</tr>
<tr>
<td>alarmed ↔ relaxed</td>
<td>48.32</td>
<td>38.15</td>
</tr>
<tr>
<td>afraid ↔ satisfied</td>
<td>52.43</td>
<td>47.83</td>
</tr>
<tr>
<td>Multi-hop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sadness ↔ joy</td>
<td>47.76</td>
<td>45.53</td>
</tr>
<tr>
<td>surprise ↔ anticipation</td>
<td>46.46</td>
<td>30.50</td>
</tr>
<tr>
<td>disgust ↔ trust</td>
<td>47.39</td>
<td>41.47</td>
</tr>
<tr>
<td>sadness, disgust ↔ joy, trust</td>
<td>47.95</td>
<td>46.23</td>
</tr>
</tbody>
</table>

Table 1: The overall results of evaluating the single-hop and multi-hop reasoning combined with different pairs of base words. For comparison with baselines, we employ Freytag’s Pyramid and Gaussian Distribution.

Table 1 shows the overall accuracy and macro-f1 scores using the ROC dataset annotated with tension development. The overall results indicate that the single-hop reasoning method outperforms the multi-hop method. We obtain the highest accuracy of 53.92 and F1 of 50.23 applying the single-hop reasoning with multiple emotions as the base words (i.e., ‘sadness’, ‘disgust’ ↔ ‘joy’, ‘trust’). This suggests that dramatic tension arises more from the current event than the event expected to occur. We obtain the second best performance with using the pair of ‘sadness’ and ‘joy’ as base words. We experimented combining multiple pairs of base words, but did not obtain a better result.

4.4 Case Study

Figure 3 shows successful examples of tension graphs generated using two stories in the ROC-Stories corpus. The upper graph shows that our method produces high tension values at the third and the fifth sentence where the protagonist is hit by an egg and leaves when the perpetrators ran away. In the latter case where all the workers reported ‘no change’ throughout the story, the generated tension does not alter significantly.

5 Conclusion

This paper presents a commonsense reasoning based approach for modeling dramatic tension, which is crucial to storytelling. To this end, we propose a computation method to convert emotion terms into a scalar using word embeddings and projection. We carried out evaluations with crowdsourced annotations using ROCStories as the source data. The evaluation results show that our proposed method outperforms simple static baselines in predicting tension development in a story. Our tension modeling can contribute to analyzing story structures and patterns (Reagan et al., 2016). Furthermore, this work can be used for applications such as authoring assistant tools to evaluate the quality of a story or to manage the user experience in interactive storytelling applications.

However, our current system is limited in that the reasoning model employed in this study does not consider previous context in reasoning, therefore, incorrect emotions can be inferred depending on the context of a story. We expect that our method will perform better when context-aware reasoning is provided.
Ethical and societal implications

Our paper describes a framework to model tension in short stories. This work can contribute to developing a simple contents filter by capturing high tension. The dataset used in our work is limited to the story domain, and does not have any privacy concerns. The annotation procedure also does not contain any privacy concerns, and the annotated data are not published.

References


S. G. Ware, R. M. Young, B. Harrison, and D. L. Roberts. 2014. A computational model of plan-based narrative conflict at the fabula level. IEEE Transactions on Computational Intelligence and AI in Games, 6(3):271–288.


A Tension Annotation

We conducted a tension annotation using Amazon Mechanical Turk. Figure 4 shows a screenshot of the annotation procedure. We allocated five minutes for each task and rewarded $0.6 for each HIT. We held a qualification test to check out understanding the narrative tension. After the qualification test, we received consent to use curated data for our research. All data are anonymized. A total of 39 workers participate in our annotation.