Modeling Event Salience in Narratives via Barthes' Cardinal Functions

Takaki Otake[1] **Sho Yokoi**[1,2] **Naoya Inoue**[1,2] *
Ryo Takahashi[1,2] **Tatsuki Kuribayashi**[1,3] **Kentaro Inui**[1,2]
[1]Tohoku University [2]RIKEN [3]Langsmith Inc.
{takaki, yokoi, ryo.t, kuribayashi, inui}@ecei.tohoku.ac.jp
naoya.inoue.lab@gmail.com

Abstract

Events in a narrative differ in *salience*: some are more important to the story than others. Estimating event salience is useful for tasks such as story generation, and as a tool for text analysis in narratology and folkloristics. To compute event salience without any annotations, we adopt Barthes' definition of event salience and propose several unsupervised methods that require only a pre-trained language model. Evaluating the proposed methods on folktales with event salience annotation, we show that the proposed methods outperform baseline methods and find fine-tuning a language model on narrative texts is a key factor in improving the proposed methods.

1 Introduction

Narratives (e.g., folktales, literary short stories) are the representations of a series of events (Abbott, 2008). Events, the essential components of narratives, differ in *salience*: some are more important to the story than others. Taking *Cinderella* as an example, *The prince falls in love with Cinderella* is a salient event; however, *Cinderella draws water from a well* is not. Estimating event salience is a fundamental task in analyzing and processing narratives, ranging from narrative analysis to automatic story generation (Ouyang and McKeown, 2015; Choubey et al., 2018; Papalampidi et al., 2019).

This study aims to estimate event salience in an unsupervised manner. Manually annotating event salience is prohibitively costly because it requires annotators to deeply understand the notion of event salience in narratology (Finlayson, 2015). In fact, despite a long history of research, very few narrative corpora are annotated with event salience. Thus, it is crucial to develop a method for estimating event salience that does not rely on event salience-annotated corpora.

In order to estimate event salience without annotated data, we adopt the definition of *cardinal functions* (**CF**s) introduced by Barthes (1966; 1975), the successor of *Proppian function* [1] (Propp, 1928), as follows:

> *cardinal functions are logically essential to the narrative action and cannot be eliminated without destroying its causal-chronological coherence.* (Prince, 2003)

This definition suggests a simple test for identifying event salience: an event is highly salient if removing it greatly reduces the story's coherence. We adopt this idea for two reasons. First, CFs are commonly used in narrative analysis (Abbott, 2008). Second, the idea of CFs can be directly operationalized without any annotated data. Computing event salience based on the idea of CFs requires measuring narrative texts' coherence, but recent advances in discourse coherence models can provide a solution for this difficulty. To date, a wide variety of discourse coherence models have been proposed (Barzilay and Lapata, 2008; Li and Jurafsky, 2017). See et al. (2019) have reported that GPT-2 (Radford et al., 2019), a powerful left-to-right language model (LM), could accurately estimate narrative texts' coherence, importantly, without any annotated data. Note that, in folkloristics and narratology, another well-known concept of

* Present affiliation: Stony Brook University.

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 28th International Conference on Computational Linguistics, pages 1784–1794
Barcelona, Spain (Online), December 8-13, 2020

event salience, *motif* is "the smallest element in a tale having a power to persist in tradition" (Thompson, 1946), but CFs are more operationalizable given accurate discourse coherence models.

2 Related work

Numerous studies on the salience of text units (e.g., word, sentence) can be related to our work. Here, we review two particularly relevant topics. First, the deletion test (Carlson and Marcu, 2001) aims to identify salient discourse segments in rhetorical structure theory (Mann and Thompson, 1987). In the deletion test, annotators check how much discourse coherence is reduced by removing the discourse unit of interest. Notably, Carlson and Marcu (2001) and Barthes (1966) use essentially the same idea, to "remove the textual unit of interest, and see how the whole structure changes," although the task is quite different. Second, extractive summarization is a task of identifying salient sentences in documents, which is formally very similar to the task of our work. Despite various existing approaches for extractive summarization (Mani, 2001; Gambhir and Gupta, 2017), it is the open problem whether these methods can be directly applied to narrative texts. Extractive summarization conventionally focuses on domains with rigid structures, such as news articles or scientific papers, while narrative texts do not have such rigid structures (Kazantseva and Szpakowicz, 2010).

In the context of narrative processing in NLP, several methods have been proposed to identify some kinds of salient events: suspenseful events in entertainment stories (Wilmot and Keller, 2020), turning points in a movie script (Papalampidi et al., 2019), and reportable events in personal narratives (Ouyang and McKeown, 2015). In contrast to studies focusing on a specific type of narrative (e.g., movie scripts, personal narratives), our method is potentially applicable to any type of narrative because Barthes' CFs is not a concept specific to those particular kinds of narratives and because our methods require only a pre-trained language model.

3 Estimating event salience

3.1 Task setup

We identify event salience in the simplified setting introduced by Ouyang and McKeown (2015). That is, we estimate a *sentence's salience* rather than an *event's salience*; we score each sentence in a narrative according to the degree to which it contains a salient event. This simplification enables us to avoid the difficult subtask of identifying phrases and clauses that express events, while addressing the task of identifying sentences that express salient events. Moreover, this sentence level identification can be easily applied to narrative processing and narrative analysis. Formally, given a narrative comprising n sentences $S_{\{1:n\}} := \{S_1, \ldots, S_n\}$ and the target sentence $S_k \in S_{\{1:n\}}$, our goal is to predict the salience score of S_k in $S_{\{1:n\}}$, denoted by $\sigma(S_k, S_{\{1:n\}}) \in \mathbb{R}$.

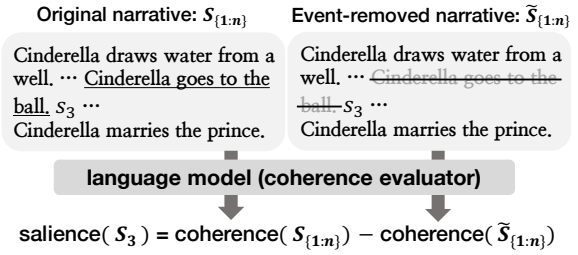

Figure 1: The basic idea of our method based on the definition of Barthes' cardinal function.

3.2 Proposed method

Overview In light of Barthes' definition (Section 1), we compute the salience score $\sigma(S_k, S_{\{1:n\}})$ as the amount of coherence loss when events in S_k are deleted from the original narrative $S_{\{1:n\}}$ (Figure 1). If a narrative's coherence is greatly reduced when events in a sentence are removed, the sentence is considered to contain a highly salient event.

To this end, let $\widetilde{S}_{\{1:n\}} := \{S_{\{1:k-1\}}, r(S_k), S_{\{k+1:n\}}\}$ be the modified narrative with all events in S_k removed from the given narrative $S_{\{1:n\}}$, where r is an *event removal function*, introduced in the following paragraph. Let $c(S)$ be the *coherence score* of a given narrative S. Then, the salience score of S_k can be

[1] *Proppian function* is defined as "an act of character, defined from the point of view of its significance for the course of the action" (Propp, 1928; Propp, 1968)

estimated as follows:

$$\sigma(S_k, S_{\{1:n\}}) := c(S_{\{1:n\}}) - c(\widetilde{S}_{\{1:n\}}) \tag{1}$$

In the following, we describe the details of (i) an event removal function r and (ii) coherence evaluator c.

Removing events in a sentence: r We employ the following three functions r.

1. Sentence Deletion (**SD**): Removing the entire sentence
2. Verb Anonymization (**VA**): Replacing all verbs in the sentence with common verbs (e.g., "do", "does", "did") based on the POS tags of each verb [2]
3. Predicate and Arguments Anonymization (**PAA**): Replacing all verbs with common verbs (as in VA) and their main arguments with an indefinite pronoun (e.g., "someone", "something") [3]

We employ VA and PAA because predicates and their arguments are main components of commonly used event representations (Chambers and Jurafsky, 2009; Pichotta and Mooney, 2016; Martin et al., 2018; Niklaus et al., 2018).

Computing narratives' coherence: c Following See et al. (2019), we compute the generation probability of a narrative using a pre-trained language model and regard it as the narrative's coherence score. Importantly, pre-trained LMs allow us to evaluate narrative's coherence without any annotated data.

Here, the narrative's generation probability is the product of word probabilities, which is influenced by the number of words in the narrative. Thus, following Li and Jurafsky (2017), we estimate the coherence score by the *average* log-likelihood of all tokens. Moreover, we consider only sentences after the target sentence $S_{\{k+1:n\}}$ because sentences whose generation probabilities change with the removal of events in S_k are limited to $S_{\{k+1:n\}}$ when using left-to-right LM, such as GPT-2. In summary, we estimate the coherence score $c(S)$ as follows:

$$c(S_{\{1:n\}}) := \frac{1}{|S_{\{k+1:n\}}|} \log P(S_{\{k+1:n\}} \mid S_{\{1:k-1\}}, S_k), \tag{2}$$

$$c(\widetilde{S}_{\{1:n\}}) := \frac{1}{|S_{\{k+1:n\}}|} \log P(S_{\{k+1:n\}} \mid S_{\{1:k-1\}}, r(S_k)), \tag{3}$$

where $|S_{\{k+1:n\}}|$ denotes the number of tokens in $S_{\{k+1:n\}}$. In order to compute the salience score for each narrative text's last sentence, we add a special token that indicates the end of a text at the end of each narrative. The proposed methods can compute the salience score for the last sentence using the generation probability of this special token. See Appendix B for further details.

4 Experiments

In this section, we provide empirical evidence that the proposed methods can evaluate event salience in narratives. Concretely, we applied the proposed method with three event-removal methods on a manually annotated folktale dataset and confirmed their performance.

# of stories	15
# of sentences	1302
# of words	18862
# of functions (salient events)	170
average # of sentences / story	86.8
average # of words / story	1257.5
average # of functions / story	11.3

Table 1: Statistics of the ProppLearner corpus

4.1 Experimental setup

Dataset We used the ProppLearner corpus (Finlayson, 2015), which contains 15 Russian folktales.

[2] We replace verbs with "do", "does", "did", "done", or "doing". For example, we replace the verb whose POS tag is VBZ (verb, third person singular present) with "does".

[3] We replace ARG0 (i.e., agent) with "someone" and ARG1 (i.e., patient) with "something".

In the corpus, verbs corresponding to the Proppian function, i.e., salient event, which is the predecessor of CFs are annotated. Following the task setup, our goal is to detect sentences that contain such verbs, i.e., salient events. The ProppLearner corpus includes POS and semantic role annotations, which are used by VA and PAA. Table 1 shows the statistics of the ProppLearner corpus.

Language model (fine-tuning) We used GPT-2 as a pre-trained language model for computing coherence scores[4]. Note that See et al. (2019) reported that GPT-2 outperforms state-of-the-art story generation models in coherence evaluation. In Appendix A, we provide further evidence that GPT-2 can accurately evaluate the coherence of the narrative texts used in our experiments. Moreover, we consider three fine-tuning settings.

1. **No fine-tuning**
2. Fine-tuning GPT-2 on **BookCorpus** (Zhu et al., 2015) as domain adaptation.
3. Fine-tuning GPT-2 on **ProppLearner** as transductive domain adaptation (Vapnik, 1998; Ouchi et al., 2019).

Baselines We compared the proposed methods with the following baseline methods:

- Random baseline: This method assigns a random score in the range $[0, 1)$ to each sentence.
- Sentence position baseline (ascending): This method assigns a score based on the position of each sentence. Here, we assumed that a sentence closer to the story's end has higher salience (Friedland and Allan, 2008).
- Sentence position baseline (descending): This method assigns a score in the opposite way to sentence position baseline (ascending).
- TF-IDF baseline: This method assigns the sum of the TF-IDF values[5] of the words in the sentence for each sentence.

Method	+ TF-IDF	Fine-tuning	MAP
Random	-	-	0.213
Sentence position (asc)	-	-	0.277†
Sentence position (desc)	-	-	0.185
TF-IDF	-	-	0.279†
Proposed method w/ **SD**	-	-	0.261†
		BookCorpus	0.265†
		ProppLearner	**0.280**†
	✓	-	0.294†
		BookCorpus	***0.301***†
		ProppLearner	0.295†
Proposed method w/ **VA**	-	-	0.245
		BookCorpus	0.258†
		ProppLearner	0.219
	✓	-	0.286†
		BookCorpus	0.287†
		ProppLearner	0.266†
Proposed method w/ **PAA**	-	-	0.254†
		BookCorpus	0.258†
		ProppLearner	0.266
	✓	-	0.285†
		BookCorpus	0.295†
		ProppLearner	***0.301***†

Table 2: MAP scores for the proposed methods and the baseline methods. We report the MAP score for random baseline method as the average over 10 seeds (standard deviation $= 0.015$). Values with a dagger mark are statistically significant improvements over the random baseline method, which was tested using the Wilcoxon signed-rank test (Wilcoxon, 1945) with $p < 0.05$. The bold score is the best performance in our proposed methods alone. The bold italic score is the best performance in combination methods of our proposed methods and the TF-IDF baseline method.

Evaluation metric We cast salience estimation as a ranking problem following Liu et al. (2018), where each method ranks a sentence based on its salience score. We used mean average precision (MAP) as an evaluation metric (Manning et al., 2008). We calculated the average precision for each story and reported their macro average score.

4.2 Experimental results

Table 2 shows the experimental results. The results show all proposed methods consistently outperform the random baseline method, and the proposed method (SD, ProppLearner) yields the best performance.

[4]We used transformers (Wolf et al., 2019) pre-trained model (12-layer, 768-hidden, 12-heads, 117M parameters).
[5]We used the score referred to as T3 in Nobata et al. (2003)

Event removal methods We found SD performed comparably to or relatively better than VA and PAA [6]. We employed VA and PAA, aiming to remove event information from the sentence more elaborately than SD. However, experimental results show that these methods do not improve the proposed method. We suspect unnatural sentences produced by the operations in VA and PAA might negatively affect inference of the language model, indicating some room for improvement in how to remove events from a sentence.

Effect of fine-tuning GPT-2 Fine-tuning GPT-2 on the BookCorpus slightly but consistently improved the proposed methods with SD, VA and PAA. We found that fine-tuning GPT-2 on the ProppLearner corpus (transductive setting) also improved the proposed methods with SD and PAA. In addition, we found that our methods' MAP scores and LM's perplexity on the ProppLearner corpus were strongly correlated. For each of SD, VA, and PAA, the Spearman's rank correlation coefficient between the MAP score in three LM settings and the LM's perplexity were -1.0, -0.5, and -1.0. This result shows that the better the LM fits the evaluation corpus, the better our methods perform.

Combining the proposed method and the baseline method We performed additional experiments with the same setting by combining each proposed method with the TF-IDF baseline method, which is the best baseline method. We normalized salience scores of each proposed method and the TF-IDF baseline method to [0, 1] within each story [7] and then added them to obtain the final salience score. Results are shown in Table 2 as **+TF-IDF**. For all cases, combination methods consistently improved MAP scores more than our proposed methods alone or the TF-IDF baseline method alone. The combination of the proposed method (SD, BookCorpus) and the TF-IDF baseline method and the combination of the proposed method (PAA, ProppLearner) and the TF-IDF baseline method achieved the best performance among all methods. The Wilcoxon signed-rank test on the best combination method (i.e., combination of the proposed method (SD, BookCorpus) and the TF-IDF baseline method) and the TF-IDF Baseline method resulted in a p-value of 0.21. This result suggests that TF-IDF-based salience cues are complementary to Barths' CFs-based cues, and they have been merged into a better measure of event salience.

Appendix C shows examples of salience evaluation results in toy *Cinderella* story and qualitative analysis of the behavior of our proposed method.

5 Discussion and future work

One promising direction for improving our proposed methods is to improve the narrative coherence evaluator. For more accurate coherence evaluation, the coherence evaluator needs to have world knowledge and common sense reasoning skills. Imagine the story of *Cinderella*. To be able to identify that the absence of event *The prince falls in love with Cinderella* leads to coherence reduction, an ideal coherence evaluator needs to recognize that this event has a strong causal relation (in this case, precondition) with the next event *Cinderella marries the prince*. Recently, several techniques have been proposed to provide language models with more world knowledge (Guan et al., 2020) and to enhance the common sense reasoning skills of language models (Mao et al., 2019). Evaluating the coherence of a narrative using these LMs can potentially improve our proposed methods.

6 Conclusions

Inspired by the Barthes' definition of cardinal functions in narratology, we have proposed methods to estimate event salience in a narrative in an unsupervised manner using an LM. In our proposed methods, we have removed events from a narrative text and have estimated event salience by comparing the coherence score of the original narrative text with that of the event-removed narrative text. Experiments on a folktales dataset have demonstrated that the proposed methods outperformed baseline methods and fine-tuning the LM on a narrative text is an effective way to improve the proposed methods.

[6] We also tried other event removal methods, such as replacing verbs with common verbs and simultaneously replacing arguments with random input vectors; however, these modifications did not affect the results significantly.

[7] We used *scikit-learn* (Buitinck et al., 2013) implementation of MinMaxScaler

References

H Porter Abbott. 2008. *The Cambridge Introduction to Narrative*. Cambridge Introductions to Literature. Cambridge University Press, 2 edition.

Roland Barthes and Lionel Duisit. 1975. An Introduction to the Structural Analysis of Narrative. *New Literary History*, 6(2):237.

Roland Barthes. 1966. Introduction à l'analyse structurale des récits. *Communications, 8, 1966. Recherches sémiologiques : l'analyse structurale du récit.*

Regina Barzilay and Mirella Lapata. 2008. Modeling Local Coherence: An Entity-Based Approach. *Computational Linguistics*, 34(1):1–34.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Lynn Carlson and Daniel Marcu. 2001. Discourse tagging reference manual. *ISI Technical Report ISI-TR-545*, 54.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised Learning of Narrative Schemas and their Participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610, Suntec, Singapore, 8. Association for Computational Linguistics.

Prafulla Kumar Choubey, Kaushik Raju, and Ruihong Huang. 2018. Identifying the Most Dominant Event in a News Article by Mining Event Coreference Relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers) (NAACL-HLT)*, pages 340–345, New Orleans, Louisiana, 6. Association for Computational Linguistics.

Mark A Finlayson. 2015. ProppLearner: Deeply annotating a corpus of Russian folktales to enable the machine learning of a Russian formalist theory. *Digital Scholarship in the Humanities*, 32(2):284–300.

Lisa Friedland and James Allan. 2008. Joke retrieval: recognizing the same joke told differently. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM)*, pages 883–892, Napa Valley, California, USA. Association for Computing Machinery.

Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A Knowledge-Enhanced Pretraining Model for Commonsense Story Generation. *Transactions of the Association for Computational Linguistics (TACL)*, 8:93–108.

Anna Kazantseva and Stan Szpakowicz. 2010. Summarizing Short Stories. *Computational Linguistics*, 36(1):71–109.

Alice Lai and Joel Tetreault. 2018. Discourse Coherence in the Wild: A Dataset, Evaluation and Methods. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 214–223, Melbourne, Australia, 7. Association for Computational Linguistics.

Jiwei Li and Dan Jurafsky. 2017. Neural Net Models of Open-domain Discourse Coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 198–209, Copenhagen, Denmark, 9. Association for Computational Linguistics.

Zhengzhong Liu, Chenyan Xiong, Teruko Mitamura, and Eduard Hovy. 2018. Automatic Event Salience Identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1226–1236, Brussels, Belgium. Association for Computational Linguistics.

Inderjeet Mani. 2001. *Automatic summarization*, volume 3. John Benjamins Publishing.

William C Mann and Sandra A Thompson. 1987. Rhetorical structure theory: Description and construction of text structures. In *Natural language generation*, pages 85–95. Springer.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Evaluation in information retrieval. In *Introduction to Information Retrieval*, page 139–161. Cambridge University Press.

Huanru Henry Mao, Bodhisattwa Prasad Majumder, Julian McAuley, and Garrison Cottrell. 2019. Improving Neural Story Generation by Targeted Common Sense Grounding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5988–5993, Hong Kong, China, 11. Association for Computational Linguistics.

Lara Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark Riedl. 2018. Event Representations for Automated Story Generation with Deep Neural Nets. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 868–875, New Orleans, Louisiana, USA. AAAI Press.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 839–849, San Diego, California, 6. Association for Computational Linguistics.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A Survey on Open Information Extraction. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 3866–3878, Santa Fe, New Mexico, USA, 8. Association for Computational Linguistics.

Chikashi Nobata, Satoshi Sekine, and Hitoshi Isahara. 2003. Evaluation of Features for Sentence Extraction on Different Types of Corpora. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*, pages 29–36, Sapporo, Japan, 7. Association for Computational Linguistics.

Hiroki Ouchi, Jun Suzuki, and Kentaro Inui. 2019. Transductive Learning of Neural Language Models for Syntactic and Semantic Analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3665–3671, Hong Kong, China, 11. Association for Computational Linguistics.

Jessica Ouyang and Kathleen McKeown. 2015. Modeling Reportable Events as Turning Points in Narrative. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2149–2158, Lisbon, Portugal, 9. Association for Computational Linguistics.

Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. 2019. Movie Plot Analysis via Turning Point Identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1707–1717, Hong Kong, China, 11. Association for Computational Linguistics.

Karl Pichotta and Raymond Mooney. 2016. Learning Statistical Scripts with LSTM Recurrent Neural Networks. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 2800–2806, Phoenix, Arizona, USA. AAAI Press.

Gerald Prince. 2003. *A dictionary of narratology*. University of Nebraska Press.

V Ya Propp. 1928. *Morfologiya skazki [The morphology of a fairy tale]*. Leningrad: Academia.

Vladimir Propp. 1968. *Morphology of the Folktale*. University of Texas Press.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog*, 1(8).

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D Manning. 2019. Do Massively Pretrained Language Models Make Better Storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861, Hong Kong, China, 11. Association for Computational Linguistics.

S Thompson. 1946. *The folktale*. Dryden Press, New York.

Vladimir Vapnik. 1998. *Statistical learning theory*. Wiley.

Frank Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83.

David Wilmot and Frank Keller. 2020. Suspense in Short Stories is Predicted By Uncertainty Reduction over Neural Story Representation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1763–1788, Online, 7. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *The IEEE International Conference on Computer Vision (ICCV)*, 12.

A Preliminary experiments

In this section, we preliminary assess the ability of GPT-2 to evaluate the coherence of texts. Results support use of GPT-2 as a coherence evaluator in our methods.

A.1 Preliminary experiment 1: Assessing GPT-2 as a coherence evaluator

In this preliminary experiment, we evaluated GPT-2 in a sentence ordering task, which is a common task for evaluating discourse coherence models (Barzilay and Lapata, 2008; Li and Jurafsky, 2017). See et al. (2019) reported that GPT-2 (Radford et al., 2019) better captures narrative text's coherence compared to the state-of-the-art model in story generation. However, See et al. (2019) evaluated GPT-2 in the document ranking task, a slightly different task from sentence ordering task. Thus, we examined GPT-2's ability as a coherence evaluator in the sentence ordering (i.e., the common task for evaluating discourse coherence models) and provided further evidence that GPT-2 could accurately evaluate the coherence of the narrative texts used in our experiments.

Given a pair from an original document and one of its permutations, the task is to assign a higher coherence score to original one. In evaluation, GPT-2 predicted that the text with a higher likelihood was more coherent. We report accuracy as the ratio of the model's correct predictions.

For each of 15 narratives in ProppLearner, we generated 80 random permutations. Then we obtained 1,200 pairs from original narrative texts and one of its permutations. Results showed that GPT-2 achieved 100% accuracy. Li and Jurafsky (2017) reported 87.3% accuracy on the same task [8]. Our result supported the validity of using GPT-2's likelihood to compute narratives' coherence.

A.2 Discussion: On evaluation of discourse coherence models

In a common task to evaluate a discourse coherence model (e.g., sentence ordering), the model is given an original text and an artificially created incoherent text; it is then required to score the former with a higher coherence score. In our preliminary experiments, we created incoherent texts by shuffling sentences as a common practice in sentence ordering task, and See et al. (2019) created an incoherent text by swapping adjacent sentences. These experiments demonstrated that GPT-2 can accurately perform these tasks.

However, as mentioned in Lai and Tetreault (2018), identifying a document's original sentence order is not the same as distinguishing low and high coherence. Just identifying sentences' correct order is not sufficient for evaluating coherence models, and we believe that more elaborate evaluation methods are needed. Lai and Tetreault (2018) provides a dataset that addresses this issue, but does not include the texts in the narrative domain.

A.3 Preliminary experiment 2: Sanity check via sentence deletion detection

In this preliminary experiment, we validated whether our method could detect event elimination as a step prior to identifying event salience. Given a narrative comprising n sentences $S_{\{1:n\}} := \{S_1, \ldots, S_n\}$, in which every sentence can be regarded as highly salient, and the target sentence $S_k \in S_{\{1:n\}}$, we evaluated whether GPT-2 can detect the sentence's deletion as a reduction in the subsequent story's likelihood: $\sigma(S_k, S_{\{1:n\}}) > 0$. If our method could not do so, our methods would be unlikely to work because they are required to reduce the subsequent story's likelihood when the target sentence (to be removed) contains salient event.

Dataset For this experiment, we need a dataset that allows us to assume that every sentence in a story is highly salient (i.e., removing any sentence would result in a significantly incoherent narrative). We used ROCStories (Mostafazadeh et al., 2016) because it is designed to meet the requirement that each story captures a rich set of causal and temporal common sense relations among daily events. Each story contains five sentences. As the event removing method, we examine SD in this preliminary experiment. We used the 2016 Spring Set (45,495 stories) and the 2017 Winter Set (52,664 stories).

[8] They used randomly selected paragraphs from Wikipedia as an original document. Therefore, we cannot directly compare our results with theirs.

Task Setting We calculated accuracy as the percentage of cases in which sentence deletion was correctly detected as $\sigma(S_k, S_{\{1:n\}}) > 0$. Random prediction would result in 50% accuracy.

Result SD with GPT-2 (No-fine-tuning) achieved 94% accuracy in both the 2016 Spring Set and the 2017 Winter Set. This result shows that SD can detect event deletion with SD as a reduction in the subsequent story's likelihood .

B Details of proposed approach

GPT-2, which is an LM we used for computing coherence score, has a limitation of input length and we can't always input the entire narrative text. Thus we practically compute coherence score $c(S_{\{1:n\}})$ as follows:

$$c(S) = \frac{1}{|S_{\{k+1:n-\ell'\}}|} \log P(S_{\{k+1:n-\ell'\}} \mid S_{\{1+\ell:k-1\}}, S_k), \tag{4}$$

where $|S_{\{k+1:n-\ell'\}}|$ denotes the number of tokens in $S_{\{k+1:n-\ell'\}}$, so as $c(\widetilde{S})$. $P(S_i)$ is computed as the product of the probability of words:

$$\begin{aligned} &\log P(S_i|\text{context}) = \log P(w_1^{(i)}|\text{context}) \\ &+ \sum_{j=2}^{|S_i|} \log P(w_j^{(i)}|\text{context}, w_1^{(i)}, \ldots, w_{j-1}^{(i)}). \end{aligned} \tag{5}$$

$|S_{\{i:j\}}|$ denotes the sum of the number of words in $(S_i, \ldots, S_j)$. ℓ' and ℓ are thresholds determined by input length limitation of the language model, L. We determine ℓ' and ℓ so that $|S_{\{k+1:n-\ell'\}}| + |S_{\{1+\ell:k\}}|$ are less than or equal to L and have a maximum value, respectively.

As mentioned at the end of Section3.2, we add a special token that indicates the end of a text at the end of each narrative text for computing the salience score for the last sentence in each narrative text [9]. The generation probability of this special token is used only when computing the salience score for the last sentence, otherwise it is ignored.

C Estimating event salience for toy example

Including salient event		Sentence	Salience score
-	S_1	Cinderella draws water from a well.	0.193
✓	S_2	A fairy godmother appears and provides Cinderella with clothes, a carriage, and a coachman.	0.309
✓	S_3	Cinderella goes to the ball.	0.214
-	S_4	Cinderella greets her stepsisters at the venue , but they do not notice.	−0.014
✓	S_5	The prince falls in love with Cinderella.	0.394
✓	S_6	Cinderella marries the prince.	−0.112

Table 3: The behavior of the proposed method (SD, No fine-tuning) in toy *Cinderella* story. Our method gives sentences a high salience score if the target sentence contains a salient event.

Table 3 shows the behavior of the proposed method (SD, No fine-tuning) in toy example, *Cinderella*. We found the last sentence tended to have a large variance in its salience score because only one special token in the succeeding story is used for estimating saliency.

[9] We used <|endoftext|> special token in transformers (Wolf et al., 2019) implementation.

salient event		sentence	likelihood diff when deleting					
			S_1	S_2	S_3	S_4	S_5	S_6
	S_1	Cinderella draws water from a well.	-	-	-	-	-	-
✓	S_2	A fairy godmother appears and provides Cinderella with clothes, a carriage, and a coachman.	0.562	-	-	-	-	-
✓	S_3	Cinderella goes to the ball.	0.807	0.876	-	-	-	-
	S_4	Cinderella greets her stepsisters at the venue , but they do not notice.	0.087	0.257	0.296	-	-	-
✓	S_5	The prince falls in love with Cinderella.	0.175	0.274	0.034	-0.111	-	-
✓	S_6	Cinderella marries the prince.	0.139	-0.113	0.220	0.082	0.394	-

Table 4: The more detailed behavior of our proposed method (SD, No fine-tuning) in toy *Cinderella* story. The value in row i, column j, represents the difference in S_i's generation probability (token-wise likelihoods are averaged within a sentence) before and after S_j is removed from the story. A Large value indicates that the removal of S_j greatly reduces the generation probability of S_i.

Table4 shows the more detailed behavior of the proposed method (SD, No fine-tuning) in *Cinderella*. For example, the last row shows that deleting salient sentences (e.g., S_3, S_5) resulted in a larger decrease in the likelihood of the ending sentences (S_6) than deleting less salient sentences (e.g., S_1, S_4). In addition, if we look at the likelihood difference of S_3, *Cinderella goes to the ball*, the likelihood dropped more when the S_2, related sentence to S_3, is removed than when S_1, which is unrelated to S_3 is removed.